



Modélisation bayésienne et robotique

Francis Colas

► **To cite this version:**

Francis Colas. Modélisation bayésienne et robotique. Robotique [cs.RO]. Université de Lorraine, 2017. tel-01647934

HAL Id: tel-01647934

<https://hal.inria.fr/tel-01647934>

Submitted on 24 Nov 2017

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

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

Modélisation bayésienne et robotique

soutenue le 17 mai 2017

pour l'obtention d'une

Habilitation à Diriger des Recherches de l'Université de Lorraine (Spécialité informatique)

par

Francis Colas

Composition du jury

- Président :* Oussama Khatib, professeur à Stanford University, Stanford, USA
- Rapporteurs :* Roland Chapuis, professeur à l'Université Clermont Auvergne, Clermont-Ferrand
Raja Chatila, professeur à l'Université Pierre et Marie Curie, Paris
Simon Lacroix, directeur de recherche CNRS au LAAS, Toulouse
- Examineurs :* Pierre Bessière, directeur de recherche CNRS à l'ISIR, Paris
François Charpillat, directeur de recherche Inria au Loria, Nancy
Ye-Qiong Song, professeur à l'Université de Lorraine, Nancy

Table des matières

Introduction	1
1 Modélisation bayésienne de la perception humaine	7
1.1 Introduction du problème	7
1.2 Modèle	8
1.2.1 Modèle générique	9
1.2.2 Spécialisation au plan	10
1.3 Expériences	11
1.3.1 Influence de la taille du champ de vision	12
1.3.2 Ambiguïté entre la vitesse et l'inclinaison	14
1.3.3 Influence des mouvements de la tête	14
1.3.4 Influence des mouvements des yeux	16
1.3.5 Translation en profondeur	16
1.3.6 Effet du cisaillement (<i>shear</i>)	18
1.4 Discussion et conclusion	19
2 Modélisation bayésienne de la prise de décision humaine	23
2.1 Introduction du problème	23
2.2 Modèle	25
2.2.1 Modèle de représentation	25
2.2.2 Modèles de décision	28
2.3 Expérience	30
2.3.1 Méthode de comparaison	30
2.3.2 Résultats globaux	31
2.3.3 Situations typiques	32
2.4 Discussion et conclusion	33
3 Alignement de nuages de points en robotique	37
3.1 Introduction du problème	37
3.2 Formalisation	38
3.2.1 Algorithme générique	38
3.2.2 Modules	40
3.2.3 Méthodologie	45
3.3 Expériences	48
3.3.1 Comparaison sur les jeux de données	48
3.3.2 Application réelle	49
3.4 Discussion et conclusion	55
4 Estimation d'état en robotique	57
4.1 Introduction du problème	57

Table des matières

4.2	Modèles	58
4.2.1	Modèles pour bateau	58
4.2.2	Modèles pour robot au sol	61
4.3	Expériences	63
4.3.1	Bateau autonome	63
4.3.2	Robot au sol	66
4.4	Discussion et conclusion	68
5	Navigation en robotique	71
5.1	Introduction du problème	71
5.2	Système	72
5.2.1	Perception	72
5.2.2	Planification	75
5.2.3	Exécution	78
5.3	Expériences	79
5.3.1	Terrain plat	79
5.3.2	Escalier tournant	81
5.4	Discussion et conclusion	81
6	Projet de recherche	85
6.1	Introduction	85
6.1.1	Motivation	85
6.1.2	Objectifs	86
6.1.3	Approche	86
6.2	Robotique située	87
6.2.1	Estimation	87
6.2.2	Étalonnage	88
6.2.3	Planification et observation	89
6.2.4	Maintenance	90
6.3	Apprentissage automatique	91
6.3.1	Apprentissage par démonstration de comportements	91
6.3.2	Interactions avec les utilisateurs	92
6.4	Conclusion	93
	Bibliographie	95
	Curriculum Vitæ	109
1	Formation	109
2	Expérience professionnelle	109
3	Enseignement et encadrement	109
3.1	Cours	109
3.2	Encadrement individuel	110
4	Intégration à la communauté de recherche	111
4.1	Activités de review et d'organisation	111
4.2	Responsabilités collectives	111
4.3	Autres	112

5	Contrats et projets	112
5.1	Investigateur principal	112
5.2	Participant	112
6	Prix	112
7	Liste des publications	113
7.1	Chapitre de livre	113
7.2	Journaux	113
7.3	Conférences internationales	114
7.4	Autres	116

Introduction

Ce document présente mes travaux et mon projet de recherche. Mais l'habilitation à diriger les recherches est aussi l'occasion de prendre un peu de recul sur mon parcours.

Ingénieur en informatique de formation, j'ai longtemps été fasciné par l'intelligence artificielle, à la fois comme outil de compréhension du vivant et comme technologie pour construire des objets et systèmes plus utiles et puissants. J'ai eu la grande chance de pouvoir faire un DEA (équivalent du master 2 recherche) en parallèle de ma dernière année d'école d'ingénieur avec en particulier un projet de master portant sur la modélisation bayésienne des cellules de direction de la tête, ces neurones chez le rat dont l'activité reflète l'orientation de la tête de l'animal (son cap) et ce, indépendamment de sa position dans un environnement donné.

Ce projet a été une étape importante pour moi avec la découverte des probabilités non plus comme partie abstraite et parfois rébarbative des mathématiques parlant de tribus et de mesure mais comme un outil, finalement assez intuitif, de représentation de connaissances. J'ai fait aussi mes premiers pas dans la science à la frontière entre la biologie, la modélisation mathématique et, tout de même un peu, l'informatique. Cette interdisciplinarité est, pour moi, une chance d'ouverture non seulement à des savoirs différents mais aussi à des méthodes diverses. La science est aussi une construction sociale avec des gens qui n'ont que rarement l'occasion d'échanger en profondeur avec des collègues issus d'autres disciplines. Rien qu'à lire des articles on peut deviner des manières de présenter, des habitudes de travail ou des critères de validations différents. Au-delà d'un constat d'hétérogénéité, c'est aussi une source de réflexion sur son propre domaine de recherche.

Ce stage a confirmé mon intérêt pour la recherche et a donc été le début d'une carrière dans laquelle je me suis successivement intéressé à des systèmes biologiques et robotiques. Ces deux champs sont reliés pour moi par l'interprétation subjectiviste des probabilités selon laquelle les probabilités peuvent mesurer les degrés de croyance en certaines hypothèses depuis le point de vue d'un agent, c'est-à-dire à partir de ce qu'il sait, suppose ou observe. En effet, un animal ou un robot sont tous deux dans une situation similaire, caractérisée par une connaissance incomplète du monde et de son fonctionnement qui les empêche de savoir avec certitude ce qui les entoure et ce qu'ils devraient faire. On peut dire aussi que le monde est trop complexe pour ce que nos sens ou les capteurs des robots peuvent en observer ou mesurer et que de nombreuses choses peuvent survenir sans qu'il soit possible de tout prévoir ou prévenir. Pour gérer cette incomplétude, on remarque qu'elle se traduit vis-à-vis d'un agent par de l'incertitude qui peut être modélisée à l'aide de probabilités.

Modélisation et programmation bayésiennes

Il peut être utile ici de détailler un peu les approches bayésiennes de modélisation et de programmation. La modélisation, en toute généralité, consiste à construire une repré-

sentation d'un système que l'on puisse manipuler ou observer de manière à mieux le comprendre. Cette représentation, appelée modèle, n'est pas confondue avec le système et n'a pas nécessairement vocation à en représenter tous les aspects. Dans les sciences le plus souvent, les modèles sont, pour reprendre le mot de Galilée, écrits en langage mathématique, pour son caractère formel. Il existe cependant plusieurs formalismes au sein des mathématiques qui peuvent servir à la construction d'un modèle. Ainsi, en physique classique, les règles sont souvent écrites sous la forme d'équations reliant entre elles des variables représentant des quantités réelles. On peut aussi écrire des modèles sous forme de propositions logiques dont on peut interroger la valeur de vérité en faisant de l'inférence logique.

La modélisation bayésienne utilise les concepts de la théorie des probabilités pour spécifier et manipuler ses représentations. On introduit ainsi des variables aléatoires, dont la valeur n'est pas connue a priori, et l'on écrit des relations permettant de calculer des valeurs de probabilité lorsque les variables prennent chacune une valeur donnée. En cela, on peut voir la modélisation bayésienne comme une extension de la logique, laquelle calcule une valeur booléenne, vrai ou faux, pour une affectation des variables, alors qu'ici il s'agit d'une valeur continue. En logique on ne peut dire que « ceci est vrai » ou « ceci est faux » sans pouvoir comparer entre deux choses vraies. En modélisation bayésienne, on ne peut pas dire « ceci est vrai », mais on peut dire « ceci est plus probable que cela ». À la limite on peut aussi dire « ceci est impossible » si sa probabilité est nulle, ce qui peut être le pendant à « ceci est faux ».

La construction d'un modèle bayésien comporte donc un certain nombre d'étapes. Il faut en premier spécifier ce dont on va parler, c'est-à-dire les variables aléatoires et ce qu'elles représentent dans le système réel. Il faut ensuite définir la fonction, la distribution de probabilité conjointe, qui associe à chacune des affectations de toutes les variables la valeur de probabilité. Cette fonction est souvent complexe à définir directement et l'on va avoir recours à des hypothèses d'indépendance conditionnelles pour la définir comme le produit de fonctions qui ne font intervenir qu'un sous-ensemble des variables et qui sont, de ce fait, plus simples. Lorsqu'enfin la distribution de probabilité conjointe est spécifiée, il est possible d'observer le modèle en s'intéressant à des relations précises entre certaines variables, qui sont calculables grâce à l'inférence probabiliste, c'est-à-dire le calcul d'une distribution de probabilité à partir de la distribution conjointe. Cette inférence est déterministe et peut toujours être faite, même si elle est parfois coûteuse.

La programmation bayésienne est une formalisation particulière de la modélisation bayésienne qui permet de spécifier complètement un modèle et une question probabiliste spécifique. Comme son nom l'indique, la programmation bayésienne ne s'arrête pas à la modélisation mais essaie de se rapprocher de la programmation informatique en ce que le programme bayésien ainsi défini doit permettre de décrire des fonctions exécutables par des ordinateurs. Il peut s'agir alors de commander un robot et l'on peut, par exemple, définir un programme bayésien de suivi de mur dont la question probabiliste serait la distribution de probabilité sur la direction à suivre par le robot sachant les valeurs relevées par les capteurs jusqu'à présent. On voit ici que le programme bayésien s'arrête à calculer une distribution de probabilité et qu'il est souvent nécessaire de choisir une valeur particulière de commande à partir de la distribution. Cette étape est celle de la décision, qui peut être vue comme une fonction de choix dont il est parfois préférable qu'elle soit non déterministe.

Mon approche de la modélisation doit beaucoup à la programmation bayésienne, notamment en ce que j'essaie toujours d'explicitier mon modèle complètement en suivant, sinon le formalisme précis, du moins les étapes importantes d'un programme, à savoir la définition des variables, des hypothèses d'indépendance conditionnelle (décomposition de la distribution de probabilité conjointe), de chacune des distributions de probabilité conditionnelles avec leurs paramètres et enfin de la question. Ce cadre est pour moi à la fois une aide à la structuration de la communication d'un modèle, mais aussi une discipline intellectuelle à même de prévenir certaines erreurs. Néanmoins, à mesure de la progression du présent document, on peut observer une distanciation progressive à l'égard de ce formalisme qui doit être vue notamment comme une plus grande liberté vis-à-vis de la lettre, en restant fidèle à l'esprit. Mais c'est peut-être aussi une limite ou plutôt un écart entre un formalisme qui suppose souvent des calculs complexes et des applications qui ont des contraintes importantes sur les calculs. Ainsi, certains de ces calculs sont soit des simplifications ou des approximations connues de calculs d'inférence, comme c'est le cas par exemple du filtre de Kalman étendu, soit des processus étrangers à cette inférence et qui ne sont pas susceptibles, du moins facilement, de formalisation bayésienne.

En cela, mes travaux n'ont pas amené d'avancées en algorithmique bayésienne, la discipline encore embryonnaire qui compléterait la programmation bayésienne par une réflexion sur la construction des modèles.

Organisation du document

Au-delà du fond de modélisation bayésienne, un autre aspect de mon travail est de m'intéresser autant à l'action qu'à la perception. Le problème de la perception peut être globalement décrit comme la construction d'une représentation interne de l'environnement à partir de connaissances incomplètes et d'observations partielles. Le problème de l'action est plutôt de décider de ce qu'on doit faire pour atteindre un but donné à partir, encore une fois, de connaissances incomplètes et d'observations partielles. Ces deux problèmes sont évidemment distincts en ce que la perception peut être vue comme un problème d'inférence, c'est-à-dire de calcul d'une distribution de probabilité, et l'action comme un problème de décision, c'est-à-dire de choix d'une valeur particulière en fonction d'un critère donné. Mais je suis particulièrement intéressé par leur articulation : la perception crée une représentation qui n'a finalement de sens que par son utilisation qui est au final la prise de décision. A contrario, le critère de décision peut être le résultat d'une inférence probabiliste similaire à celui de la perception.

Du côté de la biologie, je me suis ainsi intéressé d'une part à la perception humaine de plans par le flux optique et d'autre part à la prise de décision dans une tâche de suivi multi-cibles. En robotique, j'ai travaillé notamment sur l'alignement de nuages de points, l'estimation d'état et la navigation en trois dimensions (3D).

Le premier travail que j'aborde dans ce document est donc l'étude de la perception de formes à l'aide du flux optique ([chapitre 1](#)). L'homme perçoit les formes 3D de plusieurs manières, l'une d'entre elle étant le mouvement. En effet, le déplacement de l'image de l'objet sur la rétine, appelé flux optique, dépend à la fois du mouvement relatif entre la rétine et l'objet, mais aussi de la forme de ce dernier. En inversant ce processus, on peut donc recouvrer la forme d'un objet à l'aide du flux optique. Néanmoins, il s'agit d'un

problème mal posé en ce que nombreuses combinaisons de forme et mouvement relatif peuvent donner le même flux optique. Pour résoudre ce problème, on fait donc appel à des hypothèses restreignant ces solutions et la première est l'hypothèse de rigidité, qui préfère que la forme de l'objet reste constante. Des travaux de psychophysique mettant en œuvre des observateurs en mouvement ont cependant montré que cette hypothèse était insuffisante. Je propose donc, au [chapitre 1](#), un modèle bayésien qui couple l'hypothèse de rigidité et celle de stationnarité et qui permet de rendre compte d'une grande variété de résultats expérimentaux. Ce travail a été pour moi l'occasion d'appliquer la programmation bayésienne pour reformuler des hypothèses qualitatives en un modèle quantitatif.

Je passe ensuite à l'étude de l'action avec l'exemple de la sélection du mouvement des yeux. De nombreux éléments, tels que la texture, la couleur ou la forme, jouent un rôle dans cette sélection et je me suis intéressé plus particulièrement à l'incertitude. Plus précisément, la question est de savoir si, dans des situations de suivi d'objets, la réduction d'incertitude est mise en œuvre dans la sélection de la prochaine position du regard. Pour évaluer cette hypothèse, nous avons confronté les prédictions de différents modèles à des résultats d'une expérience de psychophysique. La tâche de cette expérience est une variante de la tâche classique de suivi multi-cibles (*multiple object tracking*) mais avec le regard libre, ce qui fait que le mouvement des yeux permet de compenser la faible résolution du champ visuel périphérique. Les modèles que nous proposons sont de complexité croissante et sont inspirés, dans leur structure en couches, de la neurophysiologie des aires cérébrales impliquées dans le processus de sélection. Le [chapitre 2](#) présente ces modèles et montre, en les comparant aux résultats expérimentaux, l'avantage de prendre en compte l'incertitude pour la prédiction du mouvement des yeux. Dans ce travail, j'ai particulièrement prêté attention à la comparaison entre les modèles pour déterminer celui d'entre eux qui semble le plus plausible au regard des données.

Après ces deux contributions dans le domaine de la modélisation pour les sciences du vivant, j'ai eu la chance de pouvoir poursuivre mes recherches en robotique. Là encore, la première contribution que je présente ressort plutôt de la perception puisqu'il s'agit d'algorithmes d'alignements de nuages de points. Ces algorithmes associent des jeux de données pour les mettre dans des coordonnées communes. Ils sont utilisés très largement dans la reconstruction d'objet, dans l'inspection, pour des applications médicales ou pour la localisation en robotique. Nous nous sommes intéressés en particulier à la robotique mobile où l'alignement se fait sur des nuages de points. Alors que le principe de ces algorithmes est simple, de nombreuses variations en ont été proposées pour des applications différentes. Dans le [chapitre 3](#), nous prenons une perspective historique sur ce problème d'alignement et montrons que la plupart des solutions proposées dans la littérature peuvent être organisées en une grille de lecture formelle. À l'aide de cette formalisation, nous proposons une bibliothèque logicielle modulaire, des jeux de données et une méthode de comparaison de ces variantes. Je montre enfin un exemple d'application sur un robot d'exploration. Ce travail a été l'occasion d'une réflexion sur la démarche scientifique dans un domaine fortement empreint d'ingénierie. Il a été aussi pour moi la première expérience d'encadrement d'une équipe et en particulier d'un étudiant en thèse.

Un autre aspect de la perception est la proprioception, c'est-à-dire la perception de la position de son corps avec ses différentes parties. En robotique, ce problème se pose souvent sous la forme de l'estimation de l'état du système. Je présente ainsi, au [chapitre 4](#),

des modèles d'estimation d'état sur deux robots différents. Le premier robot est un cataract autonome pour le suivi environnemental et le deuxième est le robot d'exploration introduit au chapitre précédent. Le problème principal de ces deux robots est l'estimation précise de la position et de l'orientation à l'aide de différents capteurs avec des caractéristiques variées en termes de fréquence, de précision ou de quantité mesurée. Dans les deux cas, nous proposons plusieurs variantes, soit du modèle de transition, soit du modèle d'observation afin d'en comparer les performances suivant différents critères. Nous montrons en particulier que même si le modèle générique est le même, il est nécessaire d'adapter les modèles de transition et d'observation à chaque système robotique. Ces contributions ont été le fruit de plusieurs collaborations entre différents projets desquels nous avons pu extraire des problématiques et des solutions communes.

Enfin, le dernier travail que je présente ici porte sur la navigation en robotique mobile. Un problème important dans ce domaine est la capacité, pour un robot, de se déplacer de manière autonome et, plus précisément, de définir un plan et de calculer les commandes motrices pour l'appliquer afin d'atteindre un but prédéfini. Pour les robots d'intérieur ou les voitures qui se déplacent sur un terrain plat, le problème est bien maîtrisé et des logiciels libres existent qui peuvent être rapidement déployés sur de tels robots. Par contre, pour de nombreuses applications telles que la recherche et le sauvetage, les robots au sol doivent s'adapter à un terrain fortement accidenté. Le **chapitre 5** présente un système capable de planifier et suivre un chemin dans un environnement complexe à partir de données capteurs bruitées pour le robot d'exploration. Pour pouvoir s'adapter à la complexité d'un espace de configuration de grande dimension, nous séparons la planification des positions de celle de la configuration. Nous démontrons les capacités du système en faisant franchir au robot d'exploration des chapitres précédents une cage d'escalier difficile. Ce dernier travail conclut la présentation non exhaustive de mes travaux de recherche par une instance de la problématique de l'action en robotique.

Il faut noter que, autant les problématiques de perception et d'action sont similaires pour un agent plongé dans un environnement, qu'il soit biologique ou artificiel, autant les questions scientifiques posées en biologie et en robotique sont bien différentes. En biologie, la question est de savoir comment ces processus, existants et observables dans une certaine mesure, fonctionnent, c'est-à-dire comprendre ce qu'ils font et de quelle manière. En robotique, il s'agit au contraire de concevoir et réaliser des systèmes à mêmes de réaliser des tâches, ce qui implique souvent d'inclure des processus similaires à la perception et à l'action. La robotique peut donc s'appuyer sur les connaissances acquises en biologie dans la limite où les substrats sont très différents et les connaissances incomplètes. La robotique, comme l'informatique, peut s'inspirer parfois du monde du vivant mais aussi souvent trouver des solutions très différentes et plus adaptées à ses capacités.

Enfin, je conclus la partie scientifique du document par mon projet de recherche pour les prochaines années. La recherche en robotique s'est longtemps attachée à améliorer l'autonomie des robots seuls dans leur environnement. Les robots, seuls ou en groupes, sont ainsi mis dans un environnement donné pour réaliser une tâche donnée et doivent donc se construire une représentation de l'environnement à partir simplement des informations de leurs capteurs, dont le point de vue est partagé, puisqu'ils se trouvent sur le robot. Or ce point de vue restreint est une contrainte forte qu'il devient maintenant possible de contourner à l'aide de la multiplicité croissante des capteurs disséminés dans nos environnements. Autant sur la planète Mars un robot mobile est relativement isolé

Introduction

(même s'il peut avoir un appui de satellites), autant dans des environnements urbains ou intérieurs, un robot a énormément à gagner à utiliser des sources d'information complémentaires. Mon objectif est ainsi plus précisément d'améliorer la mobilité des robots pour des applications d'assistance à la personne, en ne considérant plus le robot comme isolé, mais au contraire comme un élément d'un triptyque robot–environnement–humain. À plus long terme, je vise aussi une plus grande utilité par la facilitation de l'interaction et de la programmation des robots, à l'aide notamment de nouvelles approches d'apprentissage par démonstration et surtout d'une collaboration avec des utilisateurs finaux.

1 Modélisation bayésienne de la perception humaine

1.1 Introduction du problème

Le mouvement relatif entre un observateur et les objets d'une scène visuelle induit une déformation de l'image projetée sur la rétine, que l'on appelle le flux optique. Le flux optique est généré par le mouvement en 3D des formes et contient donc de l'information sur ces formes. La parallaxe de mouvement et l'effet de profondeur sont deux illustrations relevées par HELMHOLTZ (1867) et quantifiées expérimentalement par WALLACH et O'CONNELL (1953).

Bien qu'il soit simple de calculer le flux optique connaissant la géométrie 3D de la scène et le mouvement, le problème de perception est l'inverse : reconstruire la géométrie 3D et le mouvement à partir du flux optique. Or il existe une infinité de combinaisons de forme et mouvement qui donnent le même flux optique. Il s'agit donc d'un problème mal posé.¹

Une manière de résoudre ce problème est de supposer que les objets observés sont rigides. Dans ce cas, les degrés de liberté associés au mouvement sont grandement réduits et l'on peut, avec très peu de flux optique, recouvrer la forme et le mouvement (ULLMAN, 1979). Plusieurs algorithmes ont été développés, par exemple, pour l'observation de plans rigides (LONGUET-HIGGINS et PRAZDNY, 1980). Des résultats en psychophysique montrent que la performance humaine sur certaines tâches de *structure-from-motion* (SFM) est globalement cohérente avec des prédictions basées sur l'hypothèse de rigidité (WALLACH et O'CONNELL, 1953 ; KOENDERINK, 1986). Des modèles, dit affines, se basent sur l'information locale de vitesse plutôt que sur l'ensemble du flux optique (TODD et BRESSAN, 1990 ; TODD et NORMAN, 1991) mais d'autres travaux les remettent en doute (DOMINI et BRAUNSTEIN, 1998). Enfin il a été montré que la perception humaine ne choisissait pas toujours l'interprétation rigide (WEXLER, PANERAI et al., 2001).

La plupart des expériences de SFM mettent en jeu un sujet statique observant le flux optique généré par un objet en mouvement. Cependant le phénomène de SFM a été noté aussi lorsqu'un sujet en mouvement observe une scène statique (B. J. ROGERS et GRAHAM, 1979). Jusqu'à récemment il était entendu que des formes 3D perçues à l'aide du mouvement du sujet ou de l'objet étaient les mêmes tant que le stimulus du flux optique restait le même (WALLACH, STANTON et BECKER, 1974 ; B. J. ROGERS et GRAHAM, 1979). Dans certains cas, il s'avère que c'est faux : même lorsque le flux optique est contrôlé pour être le même, le mouvement du sujet influence la forme perçue (S. ROGERS et B. J. ROGERS, 1992 ; T. M. H. DIJKSTRA et al., 1995 ; WEXLER, LAMOURET et DROULEZ, 2001).

L'influence du mouvement de l'observateur sur la forme perçue a mené à la formulation d'une seconde hypothèse dans l'interprétation du flux optique, celle de la *Stationnarité* : le

1. On dit d'un problème mathématique qu'il est mal posé lorsqu'il n'admet pas une unique solution.

système visuel préfère la solution dont le mouvement, dans un référentiel allocentré, est minimal (WEXLER, PANERAI et al., 2001 ; WEXLER, 2003). Cette hypothèse se tient quand on remarque que la plupart des objets d'un champ visuel sont statiques. Alors que l'hypothèse de rigidité peut être vue comme la minimisation du mouvement *relatif* entre différents points d'un objet, l'hypothèse de stationnarité est la minimisation du mouvement *absolu* dans un référentiel indépendant de l'observateur. Prises séparément, ni la stationnarité ni la rigidité ne peuvent expliquer les résultats de SFM chez l'homme. Cependant aucun modèle cohérent n'avait intégré les deux hypothèses.

Dans ce chapitre, qui résume mon travail de thèse (COLAS, 2006 ; COLAS, DROULEZ et al., 2007), je présente un modèle bayésien générique basé sur ces deux hypothèses pour la perception de surfaces par le flux optique. L'idée est de construire un modèle de l'observateur soumis à un stimulus incertain. Dans un second temps, j'instancie ce modèle pour la perception de plans 3D. Ce choix est motivé à la fois par l'existence de données expérimentales mais aussi par la complexité des analyses. Ce modèle ne rend pas seulement compte des résultats expérimentaux ayant mené à la formulation de l'hypothèse de stationnarité mais aussi d'autres résultats non attendus qui ont été reportés.

Nous nous intéressons particulièrement aux expériences mettant en œuvre un plan en rotation avec une texture neutre ou non informative. Dans ces expériences, seul le mouvement relatif renseigne sur l'orientation du plan. Mais nous faisons varier les conditions expérimentales en particulier le mouvement de la tête ou des yeux, du plan, les deux mouvements, et la taille du champ de vision. Même si le plan est un cas particulier, c'est aussi un cas important en ce que le champ visuel est essentiellement composé de surface qui, si elles sont assez régulières, peuvent être approchées localement par des plans.

L'inférence bayésienne a commencé à être utilisée comme cadre théorique pour comprendre la perception ou les interactions entre plusieurs modalités sensorielles (Y. WEISS, SIMONCELLI et ADELSON, 2002 ; ERNST et BANKS, 2002 ; KERSTEN, MAMASSIAN et YUILLE, 2004). Dans la plupart de ces travaux, le raisonnement probabiliste est réduit à des formes simplifiées autour du théorème de Bayes. Une des simplifications classiques est l'utilisation de l'hypothèse linéaire Gaussienne qui n'est pas valide dans le cas du flux optique. Une autre simplification habituelle est la restriction des modèles bayésiens à une combinaison d'un *a priori* et d'un ensemble d'observations. Or, pour pouvoir combiner plusieurs hypothèses, telles que la rigidité et la stationnarité, de manière mathématiquement satisfaisante, il nous a semblé nécessaire de mettre ce problème de perception dans un cadre plus général incluant non seulement les observations (données sensorielles) et la perception (estimation d'un état) mais aussi des variables intermédiaires cachées. Notre modèle nous permet alors d'exprimer et tester plusieurs hypothèses de manière naturelle et efficace.

1.2 Modèle

Nous présentons le modèle en deux temps : d'abord un modèle unifié de perception de la structure d'un objet par le flux optique ; puis la spécialisation de ce modèle à la perception des plans qui sera analysée dans la [section 1.3](#).

1.2.1 Modèle générique

Fondements du modèle Le modèle générique est l'expression sous forme bayésienne des hypothèses évoquées plus haut. Plus précisément, on considère quatre hypothèses :

- l'hypothèse de rigidité (H1) ;
- l'hypothèse de stationnarité (H2) ;
- l'indépendance entre la structure de l'objet et son mouvement (H3) ;
- l'indépendance entre les conditions d'observation, le mouvement de l'objet et le mouvement de l'observateur (H4).

L'hypothèse de rigidité est une relation particulière entre le flux optique observé, le mouvement relatif d'un objet et sa structure. Plus précisément, le flux optique observé est supposé correspondre à celui théoriquement généré par l'objet rigide avec son mouvement relatif.

L'hypothèse de stationnarité indique que le mouvement de l'objet dans un référentiel extérieur à l'observateur est nul, ce qui peut aussi s'exprimer comme une relation entre le mouvement relatif de l'objet et le mouvement de l'observateur.

Les hypothèses (H3) et (H4) sont plus mineures en ce qu'elles n'expriment pas une relation particulière entre des variables mais simplement une absence de relation.

Formalisation bayésienne Le modèle générique met en œuvre plusieurs variables :

- le flux optique observé (Φ) ;
- la pose ou structure 3D de l'objet (Θ) ;
- le mouvement relatif de l'objet par rapport à l'observateur (X) ;
- le mouvement absolu de l'observateur (M) ;
- les conditions générales d'observation (Λ).

Notons que le flux optique, champ de mouvement de l'image sur la rétine, peut être représenté de plusieurs manières. Pour ce modèle, il sera plus pratique de le représenter sous une forme paramétrique compacte plutôt que, par exemple, comme un tableau des vitesses en chaque point de la rétine. Une telle représentation paramétrique, dont l'exemple du plan est donnée à la [section 1.2.2](#), nécessiterait donc, pour être intégré comme observation, une phase de prétraitement d'une séquence d'images. Cependant, dans le cas d'expériences pour lesquelles on décide a priori du stimulus sous la forme de flux optique qui est ensuite rendu sous la forme d'une séquence d'images pour être présenté au sujet, le stimulus peut être fourni au modèle directement sous sa forme paramétrée, ce qui élimine les étapes de rendu et de prétraitement.

Les hypothèses exprimées au-dessus, ainsi que la sémantique du modèle nous permettent alors d'écrire la décomposition suivante pour la distribution de probabilité conjointe sur les variables :

$$P(\Theta M X \Lambda \Phi) = P(\Theta)P(\Lambda)P(M)P(X|M)P(\Phi|\Theta X \Lambda)$$

On retrouve dans cette structure les hypothèses (H3) et (H4) :

$$\begin{aligned} P(\Theta M X \Lambda) &= P(\Theta)P(M X \Lambda) && \text{(H3),} \\ &= P(\Theta)P(\Lambda)P(M X) && \text{(H4),} \\ &= P(\Theta)P(\Lambda)P(M)P(X|M) && \text{règle de Bayes permettant d'exprimer (H1).} \end{aligned}$$

De même, il est spécifié que le flux optique observé est indépendant du mouvement de l'observateur si l'on connaît notamment le mouvement relatif de l'objet :

$$P(\Phi|\Theta M X \Lambda) = P(\Phi|\Theta X \Lambda).$$

La question que l'on souhaite poser est de recouvrer la structure lorsque l'on observe le flux optique, ainsi que le mouvement propre et les conditions d'observations $P(\Theta|\phi m \lambda)$. L'inférence bayésienne permet d'écrire :

$$P(\Theta|\phi m \lambda) \propto P(\Theta) \sum_{x \in X} P(x|m) P(\phi|\Theta x \lambda) \quad (1.1)$$

Dans l'équation 1.1, les différentes distributions sont les suivantes :

- $P(\Theta)$: distribution *a priori* sur la structure de l'objet, souvent uniforme sur le domaine de spécification ;
- $P(X|M)$: hypothèse de stationnarité, c'est-à-dire que X est plus probablement tel qu'il corresponde à un mouvement absolu plus petit ;
- $P(\Phi|\Theta X \Lambda)$: hypothèse de rigidité, c'est-à-dire que le flux optique est plus probablement tel qu'il corresponde à celui généré par l'objet rigide.

Le choix précis des domaines des variables et des distributions dépend naturellement de la spécialisation du modèle à la famille d'objets considérée. Néanmoins, la structure du modèle étant la même, l'équation 1.1 reste valide.

1.2.2 Spécialisation au plan

Dans le reste de ce chapitre, nous étudierons exclusivement le cas de l'observation d'un plan. Ce choix permet une réduction significative de la complexité du modèle et de l'inférence tout en présentant des propriétés intéressantes d'ambiguïté de perception.

Variables Dans le cas d'un plan, la structure ou pose de l'objet se réduit à sa distance à l'observateur et son orientation. Par convention et sans perte de généralité, on considère la distance du point de fixation à l'observateur comme l'unité de distance. La structure de l'objet se réduit alors à son orientation. Pour des raisons de commodité, nous avons choisi d'utiliser les gradients de profondeur selon les axes transversal et vertical : $\Theta = (\chi, \nu) = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y} \right)$.

Les mouvements, absolu de l'observateur, M , et relatif de l'objet, X , sont représentés par leurs composantes de translation et de rotation (torseur cinématique). En particulier dans la suite, on notera respectivement T et Ω les composantes de translation et rotation du mouvement relatif.

Dans le cas d'un plan, on peut montrer que le flux optique est entièrement caractérisé par huit composantes (voir COLAS, DROULEZ et al., 2007, annexe A) : deux pour la vitesse à l'origine, Φ^0 ; quatre pour les dérivées du champ de vitesse à l'origine, Φ^1 ; et deux degrés de liberté pour ses dérivées secondes, Φ^2 (LONGUET-HIGGINS, 1984).

Enfin, les conditions d'observations sont réduites à deux valeurs : petit ou grand champ de vision.

Variable	Symb.	Minimum	Maximum	Dim.	Pas/dim.
Gradient de profondeur	Θ	-4,125	4,125	2	33
Vitesse linéaire	T	-1,375 m s ⁻¹	-1,375 m s ⁻¹	3	11
Vitesse angulaire	Ω	-1,375 rad s ⁻¹	-1,375 rad s ⁻¹	3	11

TAB. 1.1: Domaine des variables discrétisées. Pour chaque variable, la table présente la valeur minimale, maximale, le nombre de dimensions et le nombre de pas de discrétisation par dimension.

Distributions La distribution *a priori* sur la structure de l'objet a été choisie de manière à être la moins informative possible pour ne pas rajouter de biais au résultat de l'inférence. Concrètement, nous avons choisi une distribution invariante par rotation de l'objet. D'autres distributions *a priori* peuvent être choisies, notamment sur des critères écologiques (préférence pour les surfaces horizontales, par exemple).

La distribution sur le mouvement relatif $P(X|M)$, expression de l'hypothèse de stationnarité, est choisie comme une distribution gaussienne centrée sur l'opposé du mouvement propre, ce qui correspond à l'absence de mouvement de l'objet dans le référentiel allocentré. La distribution gaussienne est la distribution de maximum d'entropie (la moins informative) connaissant la moyenne et l'incertitude (covariance) de la distribution. Elle correspond aussi dans ce cas à la distribution de Gibbs basée sur l'énergie cinétique.

De même, la distribution sur le flux optique observé $P(\Phi|\Theta X \Lambda)$, expression de l'hypothèse de rigidité, est choisie comme une distribution gaussienne centrée sur le flux optique théorique. Pour modéliser le fait que les composantes de deuxième ordre du flux optique sont plus difficiles à observer en petit champ de vision, les conditions d'observation Λ changent la valeur de la covariance sur ces composantes.

Implémentation et paramètres Les équations du flux optique étant non linéaires, l'inférence n'admet pas de solution en forme close. Nous avons donc discrétisé les espaces des variables pour calculer la distribution à l'aide de l'équation 1.1. Les détails des domaines des variables discrétisées sont montrés dans la table 1.1.

De plus, les distributions gaussiennes nécessitent des paramètres pour leurs matrices de covariance (voir la table 1.2). Pour toutes les distributions, nous avons simplifié le nombre de paramètres en considérant des matrices de covariance diagonales (indépendance conditionnelle des composantes) et proportionnelles à la matrice identité (symétrie entre les différentes composantes). Ces paramètres ont été déterminés par optimisation locale à partir de valeurs raisonnables. L'inférence a été effectuée à l'aide du moteur d'inférence bayésienne ProBT (LEBELTEL et al., 2004 ; BESSIÈRE, MAZER et al., 2013).

1.3 Expériences

Le flux optique a été utilisé pour étudier de nombreux cas d'ambiguïté de perception. L'exemple générique est l'inversion de profondeur due à la symétrie du flux optique jusqu'à l'ordre un. La figure 1.1 montre une représentation d'un flux optique et différentes configurations de position du plan, mouvement du plan et mouvement de l'observateur

Paramètre	Matrice de covariance
$\sigma_T = 0,3 \text{ m s}^{-1}$	$\Sigma_T = \sigma_T^2 Id_{3 \times 3}$
$\sigma_\Omega = 1,2 \text{ rad s}^{-1}$	$\Sigma_\Omega = \sigma_\Omega^2 Id_{3 \times 3}$
$\sigma_{\Phi^0} = 1,0 \text{ m s}^{-1}$	$\Sigma_{\Phi^0} = \sigma_{\Phi^0}^2 Id_{2 \times 2}$
$\sigma_{\Phi^1} = 0,0025 \text{ s}^{-1}$	$\Sigma_{\Phi^1} = \sigma_{\Phi^1}^2 Id_{4 \times 4}$
$\sigma_{\Phi^2 \lambda=P} = 5,0 \text{ m s}^{-1}$	$\Sigma_{\Phi^2 \lambda=P} = \sigma_{\Phi^2 \lambda=P}^2 Id_{2 \times 2}$
$\sigma_{\Phi^2 \lambda=G} = 0,2 \text{ m s}^{-1}$	$\Sigma_{\Phi^2 \lambda=G} = \sigma_{\Phi^2 \lambda=G}^2 Id_{2 \times 2}$

TAB. 1.2: Paramètres et matrices de covariance pour les distributions.

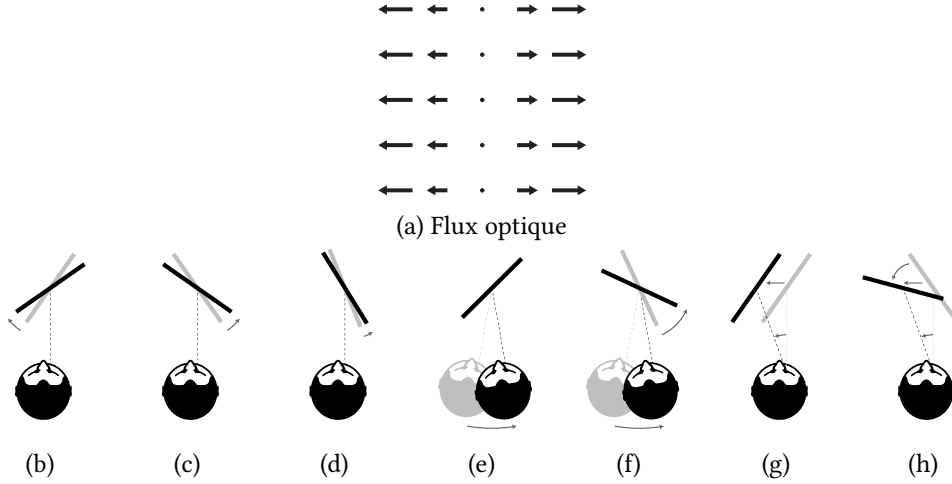


FIG. 1.1: Différentes configurations (b–h) générant le même flux optique de premier ordre (a). (Adapté de (COLAS, DROULEZ et al., 2007).)

qui peuvent le générer. Les différences éventuelles de flux optique entre certaines de ces différentes configurations ne peuvent se mesurer qu’au niveau du flux optique de second ordre. Ces différentes ambiguïtés sont détaillées par des expériences de psychophysique auxquelles nous comparons notre modèle.

1.3.1 Influence de la taille du champ de vision

La première ambiguïté que l’on peut étudier est celle de l’inversion de profondeur. Si l’on compare la [figure 1.1b](#) et la [figure 1.1c](#), on remarque que ces deux configurations sont symétriques à la fois par rapport au plan sagittal² et au plan fronto-parallèle passant par le point de fixation.³ Sur la base du seul flux optique de premier ordre, il est impossible de distinguer entre un plan incliné d’un côté et tournant dans un certain sens du plan incliné de l’autre côté et tournant dans l’autre sens.

C’est une ambiguïté connue et il a été montré qu’elle disparaissait en grand champ de vision (T. M. H. DIJKSTRA et al., 1995). Nous reprenons l’expérience décrite par CORNILLEAU-

2. Le plan sagittal est le plan vertical séparant la gauche de la droite.

3. Un plan fronto-parallèle est un plan parallèle au plan frontal, le plan vertical séparant l’avant de l’arrière.

	Expérience	Modèle
Petit champ de vision	48,8 %	44,6 %
Grand champ de vision	3,1 %	3,3 %

TAB. 1.3: Influence de la taille du champ de vision sur le taux d'inversion. Les données expérimentales sont reprises de CORNILLEAU-PÉRÈS et al. (2002).

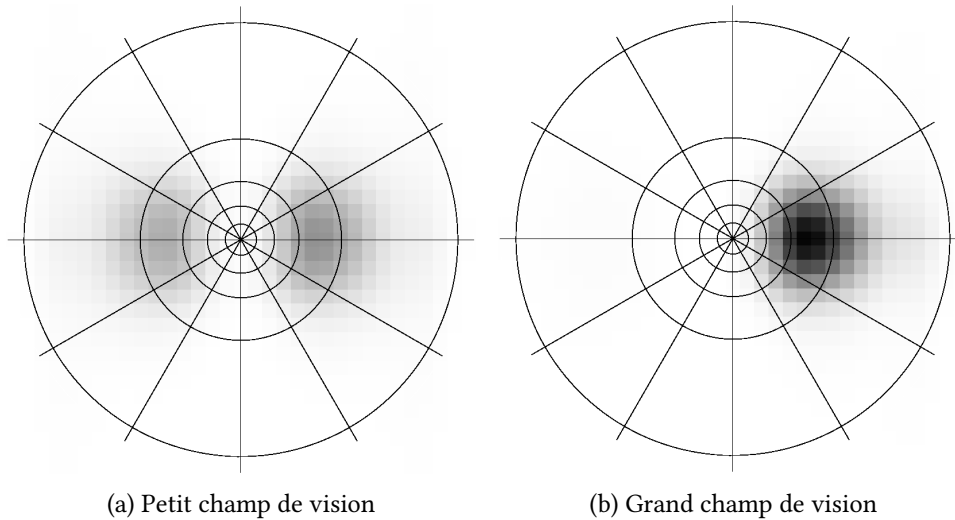


FIG. 1.2: Exemples de distribution de probabilité sur la configuration du plan. Une densité de probabilité plus élevée est indiquée par un gris plus sombre. La configuration du plan est représentée en coordonnées polaires : l'angle polaire est l'orientation du plan et le rayon est la tangente de l'angle d'inclinaison. (Adapté de (COLAS, DROULEZ et al., 2007).)

PÉRÈS et al. (2002) dans laquelle un sujet immobile devait observer un plan en rotation et reporter son orientation. Deux conditions ont été testées avec un champ de vision respectif de 8° et 60° .

La mesure rapportée est le taux d'inversion, correspondant à la proportion de réponse moins proche du plan d'origine que de sa symétrie. En cas de confusion totale, ce taux serait donc de 50 %.

La [table 1.3](#) montre les taux rapportés dans le papier original et ceux rapportés par le modèle. On remarque qu'en petit champ de vision, le taux d'inversion est proche de 50 % alors qu'il est très faible en grand champ de vision. Cela indique que l'influence du flux optique de second ordre est très limitée en petit champ de vision.

On peut comparer les distributions de probabilité sur la configuration du plan, $P(\Theta|\phi m \lambda)$, entre grand et petit champ de vision ([figure 1.2](#)). En accord avec la [table 1.3](#), on peut observer la présence de deux modes en des positions symétriques dans la distribution en petit champ de vision ([figure 1.2a](#)), mais un seul mode correspondant au stimulus en grand champ de vision ([figure 1.2b](#)), lorsque le flux optique d'ordre deux permet de lever l'ambiguïté.

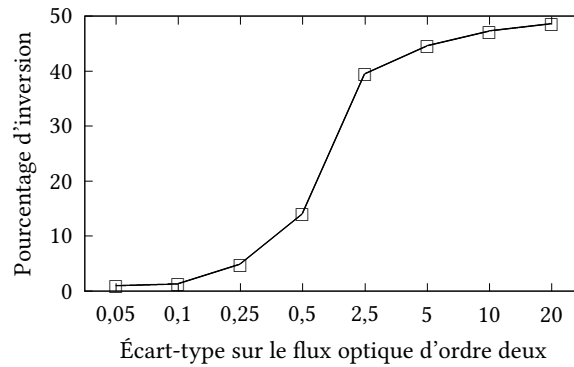


FIG. 1.3: Influence de l'incertitude sur le flux optique d'ordre deux sur le taux d'inversion prédit par le modèle. (Adapté de (COLAS, DROULEZ et al., 2007).)

Concrètement dans le modèle cet effet est gouverné par l'écart type sur les composantes du flux optique d'ordre deux, $\sigma_{\Phi 2}$. Plus cet écart type est grand, plus l'incertitude sur le flux optique d'ordre deux est grande et plus le taux d'inversion élevé (figure 1.3).

1.3.2 Ambiguïté entre la vitesse et l'inclinaison

Une ambiguïté supplémentaire du flux optique est le lien entre l'inclinaison du plan (son angle par rapport à l'axe transversal) et la vitesse de rotation. Plus précisément, le flux optique (figure 1.1a) d'un plan incliné en rotation (figure 1.1c) peut correspondre aussi, à l'ordre un, à celui d'un plan plus incliné mais en moindre rotation (figure 1.1d).

DOMINI et CAUDEK (1999) proposent ainsi une expérience où le plan est soit peu incliné (environ 56° soit une tangente de 1,5) soit très incliné (environ 72° soit une tangente de 3) et tourne avec une vitesse de $0,25 \text{ rad s}^{-1}$ ou $0,5 \text{ rad s}^{-1}$. De cette expérience, ils rapportent la moyenne des tangentes des inclinaisons estimées par les sujets (table 1.4). On observe tout d'abord une sous-estimation de l'inclinaison par les sujets. De plus, on retrouve la confirmation de l'ambiguïté entre la vitesse et l'inclinaison par le fait que les inclinaisons rapportées sont similaires pour un plan faiblement incliné en rotation rapide et un plan plus incliné en rotation plus lente.

Cet effet est correctement reproduit par notre modèle par un compromis entre la distribution *a priori* sur la configuration du plan, $P(\Theta)$, et celle sur le mouvement, $P(X|M)$. À noter que DOMINI et CAUDEK (2003) proposent un modèle de maximum de vraisemblance pour expliquer cette ambiguïté qui est un cas particulier de notre modèle (voir COLAS, DROULEZ et al., 2007, section 3.3).

1.3.3 Influence des mouvements de la tête

Il a aussi été montré que le mouvement de l'observateur modifiait la perception de la profondeur à l'aide du flux optique. Une manière de mettre en lumière cet effet est de présenter strictement le même flux optique à un observateur immobile ou en mouvement.

Un premier effet du mouvement propre est la diminution de l'ambiguïté menant à l'inversion (S. ROGERS et B. J. ROGERS, 1992 ; T. M. H. DIJKSTRA et al., 1995 ; WEXLER, LA-MOURET et DROULEZ, 2001 ; WEXLER, PANERAI et al., 2001). Comme pour un observateur

Vitesse angulaire (rad s^{-1})	Expérience		Modèle	
	0,25	0,5	0,25	0,5
Peu incliné (1,5)	1,13	1,29	0,66	1,00
Très incliné (3)	1,28	1,71	1,00	1,64

TAB. 1.4: Moyenne de la tangente de l'inclinaison. Les résultats expérimentaux sont re-produits de (DOMINI et CAUDEK, 1999).

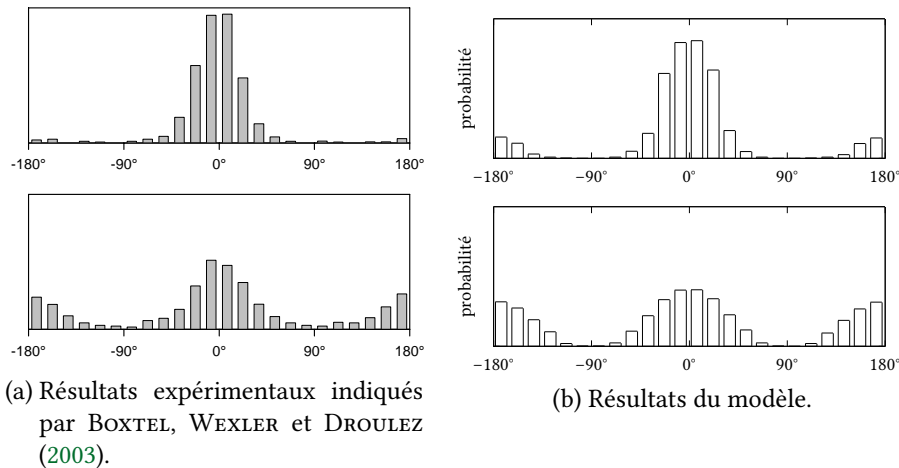


FIG. 1.4: Distributions d'erreur sur l'angle d'orientation. En haut : en condition active (observateur en mouvement) ; en bas : en condition immobile. (Adapté de (COLAS, DROULEZ et al., 2007).)

immobile, le flux optique de la [figure 1.1a](#) peut correspondre, pour un même mouvement de l'observateur à plusieurs configurations ([figure 1.1e](#) et [figure 1.1f](#)). La symétrie fronto-parallèle est conservée, mais elle correspond dans ce cas à des mouvements absolus différents (puisque le plan de symétrie bouge avec l'observateur).

BOXTEL, WEXLER et DROULEZ (2003) compare la perception d'un même flux optique en petit champ de vision lorsque le sujet est actif, ou immobile. Les histogrammes des erreurs des sujets dans chacune de ces deux conditions sont reproduits à la [figure 1.4a](#). On observe qu'alors qu'en condition immobile on retrouve une distribution présentant deux modes (0° et 180°) correspondant à l'ambiguïté de profondeur, en condition active, l'ambiguïté disparaît et il ne reste plus que le mode centré autour de 0° d'erreur.

La [figure 1.4b](#) montre que le modèle reproduit correctement cet effet grâce à la distribution sur le mouvement relatif sachant le mouvement propre, $P(X|M)$. En effet, en condition immobile, la probabilité des mouvements des deux percepts ([figure 1.1b](#) et [figure 1.1c](#)) est la même alors qu'en condition active, le mouvement de l'un ([figure 1.1f](#)) est moins probable que celui de l'autre ([figure 1.1e](#)).

1.3.4 Influence des mouvements des yeux

L'expérience précédente s'intéresse à des mouvements d'un sujet contraint de garder le regard en un point de fixation. Un autre type de mouvement propre est celui des yeux suivant un objet. La même ambiguïté de profondeur peut alors intervenir (figure 1.1g et figure 1.1h).

Ce cas a été testé dans une expérience avec un profil sinusoïdal où la phase du profil (le haut est loin ou proche de l'observateur) est similaire à l'orientation du plan (NAJI et FREEMAN, 2004). Plus précisément, les auteurs proposent trois conditions :

- (A) l'objet est en translation et l'observateur fixe un point de fixation (pas de suivi) ;
- (B) l'objet est en rotation et l'observateur fixe un point de fixation (pas de suivi) ;
- (C) l'objet est en translation comme la condition (A) et l'observateur le suit du regard, ce qui donne le même flux optique que la condition (B).

Pour chacune des conditions le champ de vision est petit et la difficulté de la tâche peut être variée en renforçant l'amplitude du profil sinusoïdal, ce qui correspond à incliner le plan plus ou moins fortement. Dans le cas où la phase du profil est soit 0° soit 180° , on peut affecter un signe à l'amplitude pour indiquer la phase. NAJI et FREEMAN (2004) rapportent les résultats en termes de pourcentage de perception « *top-far* » ce qui correspond à une phase positive dans leur cas.

La condition (B) est similaire à l'ambiguïté d'inversion de profondeur et on s'attend à ce que les sujets se trompent avec une forte proportion puisque l'on est en petit champ de vision et que les deux percepts ont la même quantité de mouvement. Dans les conditions (A) et (C), le mouvement de l'œil de l'observateur induit une dissymétrie entre les deux percepts comme dans l'expérience précédente.

La figure 1.5 montre d'une part les résultats rapportés par les observateurs (en haut) et ceux prédits par le modèle. On constate que les conditions (A) et (C) permettent bien la résolution de l'ambiguïté contrairement à la condition (B). Cependant, la sigmoïde de la condition (C) est plus abrupte que celle de la condition (A), ce qui montre une meilleure distinction du percept simulé et de son symétrique. C'est dû à un mouvement plus important du profil inverse en condition (A) avec une rotation plus importante. On remarque en condition (B) une préférence marquée des sujets pour un percept « *top-near* » ce qui n'est pas le cas du modèle (*a priori* non informatif).

1.3.5 Translation en profondeur

Un autre type d'ambiguïté lié au mouvement propre se produit en cas de translation suivant l'axe transversal. Dans certains cas précis, la combinaison d'une telle translation et d'un mouvement de rotation suivant un axe (par exemple horizontal) peut produire un flux optique très proche d'une simple rotation suivant un autre axe (vertical dans ce cas). La figure 1.6 illustre cet exemple en montrant la configuration et les flux optiques générés par les mouvements.

Nous reprenons les résultats de WEXLER, LAMOURET et DROULEZ (2001) et WEXLER (2003) dont l'expérience consiste à demander à des sujets d'estimer l'orientation d'un plan (angle de sa pente par rapport à l'axe vertical) en étant successivement en mouvement (figure 1.6e et figure 1.6f) ou immobile (figure 1.6c et figure 1.6d). Dans le cas où le sujet

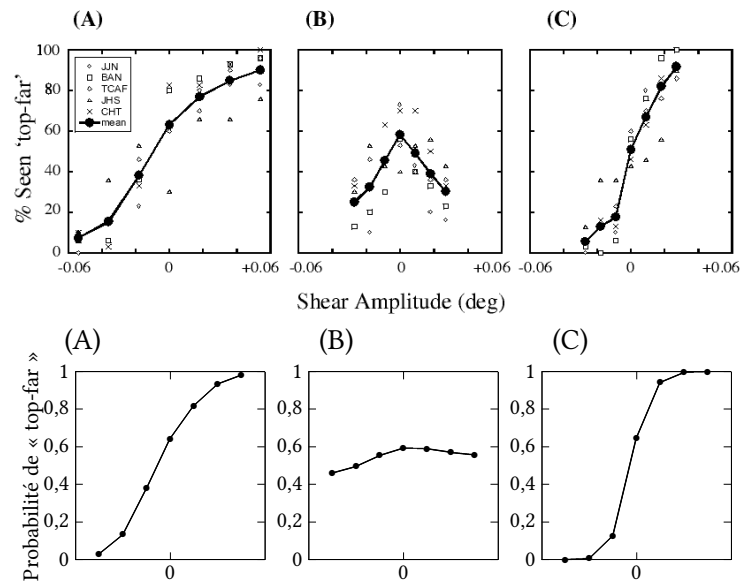


FIG. 1.5: Pourcentage de réponses « *top-far* » en fonction de la force du stimulus. En haut : résultats expérimentaux (NAJI et FREEMAN, 2004) ; en bas : résultats du modèle (COLAS, DROULEZ et al., 2007).

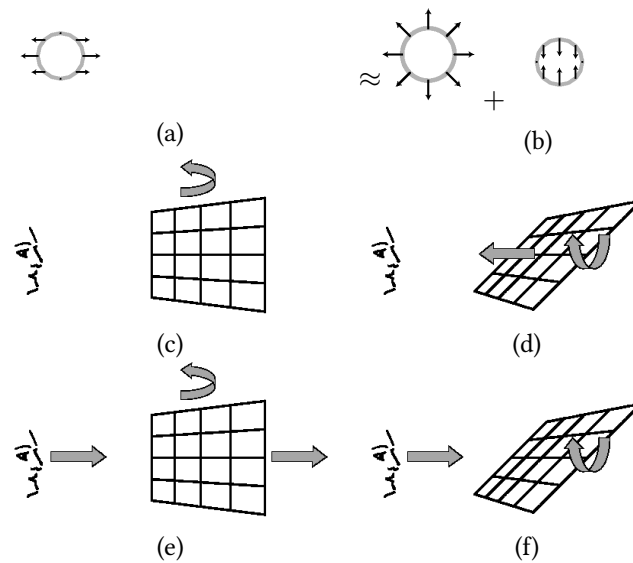


FIG. 1.6: Ambiguïté liée à la translation. (a) flux optique lié à une rotation verticale. (b) séparation du même flux optique en composante de translation et de rotation. (c) observateur immobile, pas de translation relative. (d) observateur immobile, translation relative. (e) observateur actif, pas de translation relative. (f) observateur actif, translation relative. (Adapté de (COLAS, DROULEZ et al., 2007).)

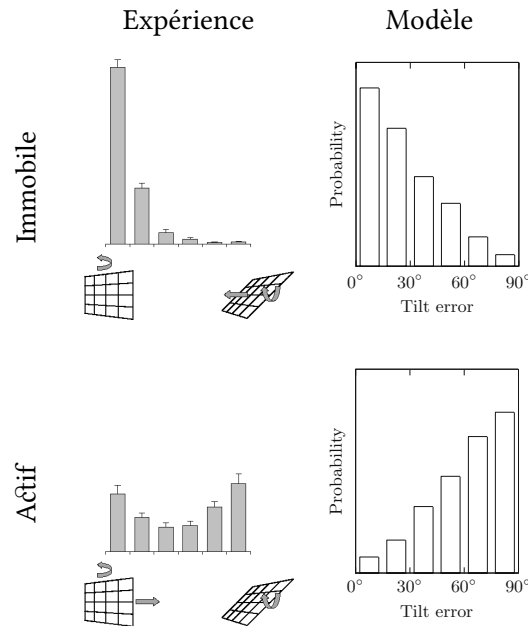


FIG. 1.7: Distribution sur l'orientation du plan. En haut : en condition immobile ; en bas : en condition active. Colonne de gauche : résultats expérimentaux (WEXLER, 2003) ; colonne de droite : résultats du modèle (COLAS, DROULEZ et al., 2007).

est immobile, on lui présente le même flux optique que celui qu'il avait perçu en condition active.

L'hypothèse de rigidité prévoit ainsi qu'en condition active la configuration de la [figure 1.6f](#) soit préférée alors qu'en condition active ce serait celle de la [figure 1.6c](#) ; c'est-à-dire une différence d'angle de 90° pour exactement le même flux optique observé.

La [figure 1.7](#) montre que les résultats expérimentaux sont plus nuancés. En effet, en condition immobile le percept majoritaire est bien celui le plus statique mais, en condition active, la distribution de réponses des sujets est bimodale, avec une légère préférence pour le percept statique, mais une proportion significative de réponses non statiques. Les résultats du modèle sont plus cohérents avec l'hypothèse de stationnarité.

1.3.6 Effet du cisaillement (*shear*)

Un dernier aspect que nous avons testé avec le modèle est l'effet de la composante de cisaillement (*shear*) dont il a été montré qu'elle détériorait les performances des sujets (CORNILLEAU-PÉRÈS et al., 2002). Cette composante du flux optique apparaît en fonction de l'angle entre l'orientation du plan et l'axe de rotation (nous appellerons cet angle, angle de cisaillement). La [figure 1.8](#) illustre les cas où il n'y a pas de cisaillement et où il est maximal.

Nous reprenons les résultats expérimentaux par BOXTEL, WEXLER et DROULEZ (2003) qui montrent que l'erreur sur l'estimation de l'orientation du plan est plus importante pour un grand cisaillement plutôt que pour un plus petit ([figure 1.9a](#)). Cet effet est d'autant plus marqué lorsque l'observateur est immobile où les réponses sont alors proches du

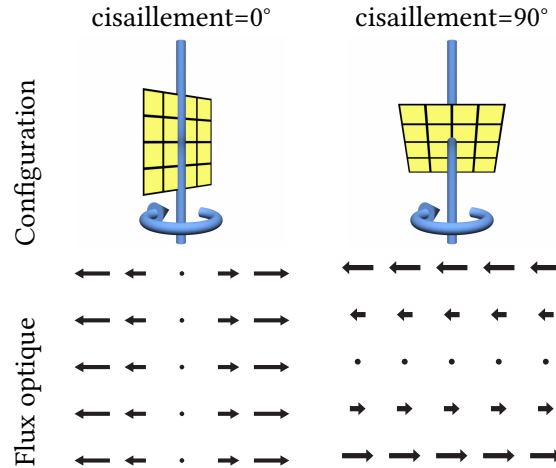


FIG. 1.8: Illustration du cisaillement dans le flux optique. En haut : configuration du plan et de son axe de rotation ; en bas : flux optique généré pour un observateur immobile. À gauche : absence de cisaillement ; à droite : cisaillement maximal. (Adapté de (COLAS, DROULEZ et al., 2007).)

hasard.

La [figure 1.9b](#) montre que notre modèle est capable de reproduire cet effet. Notre modèle est capable de reproduire cet effet en se reposant sur la distribution sur le mouvement relatif qui exprime la stationnarité. Plus précisément, les mouvements correspondant à une erreur sont différents pour un grand et un petit cisaillement. Pour un petit cisaillement il s'agit d'une rotation et d'une translation alors qu'il s'agit de deux rotations pour un grand cisaillement. Ainsi, les valeurs respectives des variances sur les composantes de rotation et de translation de la distribution $P(X|M)$ sont responsables de la reproduction de cet effet. Dans notre modèle, il serait même possible de renverser cet effet en changeant ces valeurs.

1.4 Discussion et conclusion

De nombreux choix interviennent dans la conception d'un modèle. Nous avons séparé ici l'expression conceptuelle des hypothèses de stationnarité et de rigidité dans le modèle générique d'un certain nombre de choix nécessaires à la spécification complète d'un modèle calculable. Dans notre spécialisation du modèle générique au cas particulier d'un plan, nous avons été amenés en particulier à définir les représentations des variables.

Concrètement, nous avons choisi de limiter la représentation du flux optique à une représentation paramétrique minimale pour un plan en mouvement relatif. De fait, ce choix exclut du domaine de notre modèle toute étude de l'influence d'un flux optique qui ne serait pas simplement issu d'un plan. Il serait alors intéressant d'étudier l'effet de composantes additionnelles du flux optique sur la perception de plans.

Un deuxième choix important est celui de ne s'intéresser qu'à un modèle instantané de perception. Il serait possible de modifier le modèle pour prendre en compte une succession d'observation. De la sorte, on pourrait s'intéresser à des questions sur l'évolution en

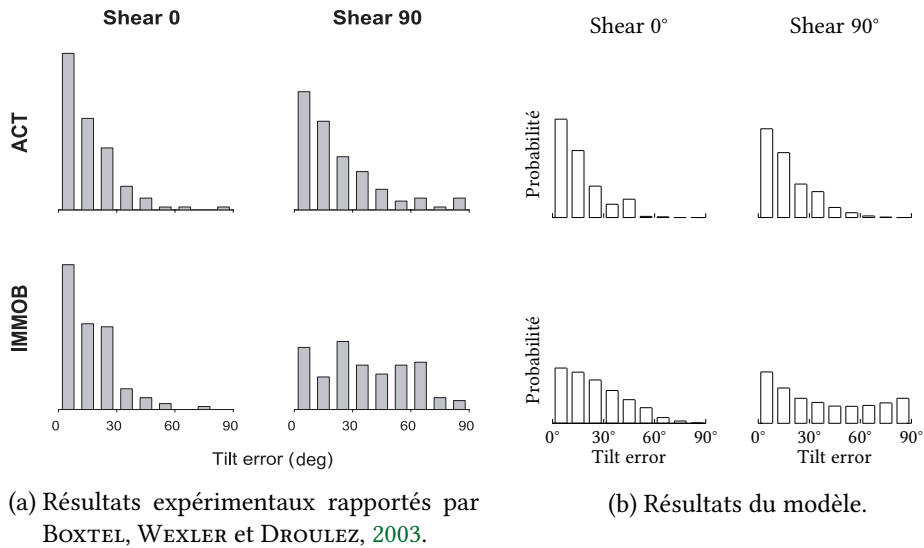


FIG. 1.9: Distributions d'erreur sur l'orientation du plan en fonction de la condition (observateur immobile ou actif) et de l'angle de cisaillement (reprises de (COLAS, DROULEZ et al., 2007)).

cours d'observation du percept mais aussi sur l'effet d'un changement de flux optique (accélération, par exemple). La plupart des expériences ne s'intéressent pour l'instant qu'à un mouvement sans variation au cours d'une courte observation.

Un dernier choix important pour les résultats d'un modèle est celui des paramètres. Notre modèle en a six, comme indiqué dans la [section 1.2.2](#) et ils ont été fixés de manière à reproduire les résultats expérimentaux. Il faut noter que notre modèle n'est pas un approximateur universel⁴ et que le nombre de paramètre est assez faible pour la diversité de résultats mis en œuvre. De plus certaines des expériences de la [section 1.3](#) se placent dans des situations similaires du point de vue de notre modèle mais obtiennent des résultats numériquement différents. Cela est dû à la fois à de petites variations de protocole expérimental et à la variabilité entre les sujets. Ainsi nous ne pensons pas être dans un cas de surapprentissage, malgré l'absence de séparation entre données d'apprentissage et données d'évaluation.

Enfin, l'intérêt d'un modèle est de proposer une formulation permettant d'expliquer des phénomènes, mais aussi de proposer de nouvelles expériences et d'en prédire des résultats. Dans le cas de notre modèle, on a vu par exemple que pour rendre compte de l'effet du cisaillement, il fallait que la variance sur la rotation soit plus grande que celle sur la translation dans l'expression de la stationnarité. Une expérience suggérée par cette observation est donc d'essayer de vérifier cette relation indépendamment du cisaillement.

En conclusion, nous avons présenté dans ce chapitre un travail sur la modélisation de la perception humaine. Plus précisément, nous avons proposé un modèle générique pour faire une synthèse des hypothèses de rigidité et de stationnarité de la littérature. Nous avons montré comment spécialiser ce modèle à l'observation de plans et confronté ses

4. Un approximateur universel est un système permettant de reproduire n'importe quelle fonction avec un choix judicieux de ses paramètres.

résultats à ceux, expérimentaux, issus de la littérature. L'approche bayésienne semble donc adéquate pour modéliser des phénomènes de perception humaine.

2 Modélisation bayésienne de la prise de décision humaine

2.1 Introduction du problème

Une saccade oculaire est un mouvement rapide de l'œil que l'on fait régulièrement, quelques fois par seconde. Ce genre de mouvement permet d'aller fixer son regard quelque part mais est aussi utilisé pour explorer la scène visuelle. De nombreux facteurs interviennent dans le choix d'une saccade, comme la saillance, mais nous nous intéressons ici au cas du suivi d'objets identiques et bien repérables. Une caractéristique importante de la vision est la différence entre la zone centrale, correspondant à la fovéa, avec une grande acuité visuelle, et la vision périphérique beaucoup moins précise. De cette différence dérive une plus grande incertitude sur la scène visuelle périphérique et donc éventuellement le besoin de faire des saccades oculaires pour réduire cette incertitude.

Dans ce chapitre, repris de (COLAS, FLACHER et al., 2009) nous nous intéressons au rôle possible de l'incertitude dans le processus de sélection de mouvements en cas de perception active. Nous proposons un modèle bayésien inspiré par la neurophysiologie des zones cérébrales impliquées dans cette sélection pour pouvoir évaluer différents principes de sélection pendant une tâche de suivi de plusieurs objets (*multiple object tracking* (MOT) avec regard libre (PYLYSHYN et STORM, 1988 ; TANNER, CANTO-PEREIRA et BÜLTHOFF, 2007)).

Nous n'avons pas l'ambition de développer un modèle complet de la neurophysiologie de la sélection des mouvements des yeux mais nous nous inspirons de leur anatomie et de leur électrophysiologie. Les saccades et la poursuite¹ partagent une grande partie de leur architecture fonctionnelle (KRAUZLIS, 2004). Parmi ces régions du cerveau (figure 2.1), le colliculus supérieur (*superior colliculus* (SC)), les champs oculaires frontaux (*frontal eye fields* (FEF)) et la partie latérale du sillon intrapariétal (*lateral intraparietal cortex* (LIP)) ont un certain nombre de points communs. Elles reçoivent toutes des informations concernant la position de points d'intérêt dans le champ visuel (activité visuelle), mémorisent ces positions (activité retardée) et sont impliquées dans la sélection des cibles des saccades parmi ces points (activité pré-saccadique) (MOSCHOVAKIS, SCUDDER et HIGHSTEIN, 1996 ; WURTZ et al., 2001 ; SCUDDER, KANEKO et FUCHS, 2002). Ces positions sont encodées par des cellules avec des champs récepteurs ou moteurs définis dans un référentiel rétinotopique.

Dans le SC, ces cellules sont clairement organisées en carte topographique chez plusieurs espèces animales (ROBINSON, 1972 ; McILWAIN, 1983 ; SIMINOFF, SCHWASSMANN et KRUGER, 1966 ; HERRERO et al., 1998). Chez les primates, ces cartes ont une géométrie logarithmique complexe (figure 2.2) (ROBINSON, 1972 ; OTTES, VAN GISBERGEN et EGGERMONT, 1986) similaire à celle du cortex strié (SCHWARTZ, 1980). Pour le FEF, les études

1. La poursuite est un type de mouvements des yeux dans lequel le regard se déplace lentement et s'asservit à un objet de la scène visuelle.

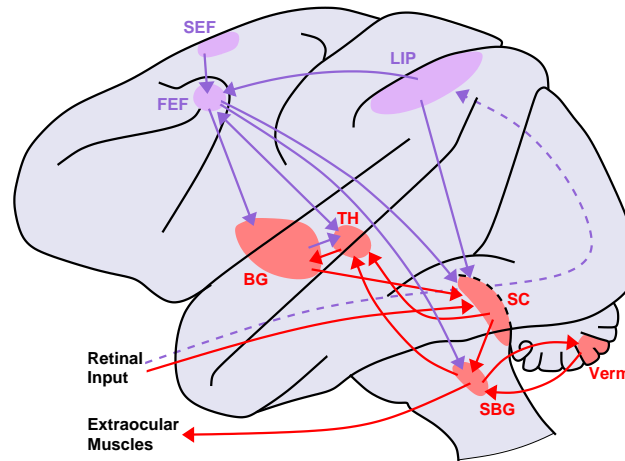


FIG. 2.1: Aires du cerveau partagées par les circuits de saccade et de poursuite (macaque). BG (Basal Ganglia) : ganglions de la base, BON (Brainstem Oculomotor Nuclei) : noyaux oculomoteurs du tronc cérébral, FEF (Frontal Eye Field) : champs visuels frontaux, LIP (Lateral bank of the IntraParietal sulcus) : partie latérale du sillon intrapariétal, SC (Superior Colliculus) : colliculus supérieur, SEF (Supplementary Eye Fields) : champs visuels supplémentaires, TH : thalamus, Verm : vermis du cervelet. Adapté de (KRAUZLIS, 2004) et repris de (COLAS, FLACHER et al., 2009).

de la géométrie montrent clairement un encodage logarithmique de l'excentricité de la position (SOMMER et WURTZ, 2000) mais des études complémentaires sont nécessaires pour comprendre l'encodage de l'orientation. Enfin, la structure des cartes du LIP reste à décoder, même si une organisation topographique continue semble exister avec une sur-représentation du champ visuel central (BEN HAMED et al., 2001).

Les neurones en relation avec la mémoire de travail spatiale dans SC (MAYS et SPARKS, 1980) FEF (GOLDBERG et BRUCE, 1990) et LIP (GNADT et ANDERSEN, 1988 ; BARASH et al., 1991b ; BARASH et al., 1991a) (appelée aussi cellules quasi-visuelles ou QV) sont capables de ré-allocation dynamique (dynamic remapping). C'est-à-dire que ces cellules peuvent être activées par la mémoire de la position d'une cible, même si la cible n'était pas dans le champ récepteur de la cellule au moment de l'observation de cette cible. Elles se comportent comme si elles faisaient partie d'une carte mémorielle rétinotopique, intégrant un mécanisme de ré-allocation permettant le déplacement suite à un mouvement des yeux d'une activation mémorisée. Des modèles neuronaux de ce genre de cartes ont déjà été proposés, soit pour SC, soit pour FEF (DROULEZ et BERTHOZ, 1991 ; BOZIS et MOSCHOVAKIS, 1998 ; MITCHELL et ZIPSER, 2003).

Sans chercher à copier la neurophysiologie, nous avons repris certaines caractéristiques dans notre modèle. Nous avons plusieurs couches organisées en cartes rétinotopiques avec une géométrie complexe logarithmique et la capacité de ré-allocation dynamique.

Pour étudier la sélection de mouvement oculaire de manière contrôlée, nous utilisons des données issues d'une version avec regard libre de la tâche classique de suivi multi-cibles (MOT voir la figure 2.3). Dans cette expérience, on présente aux participants un ensemble de cibles parmi des distracteurs. Tous ces objets sont des disques de 1° indistinctibles les uns des autres à part dans la phase de présentation où les cibles clignotent.

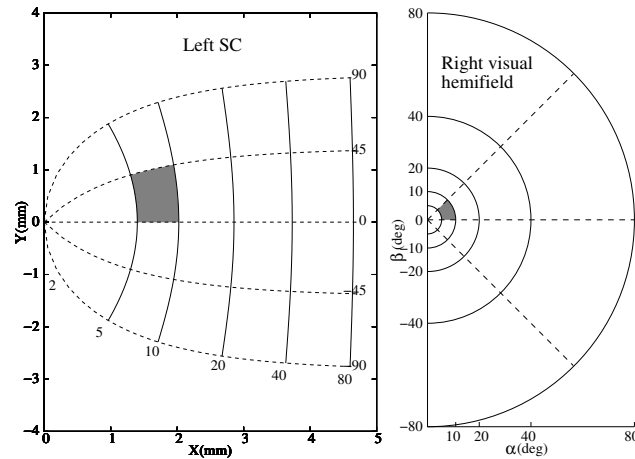


FIG. 2.2: Géométrie du colliculus supérieur chez le macaque. La position des objets dans le champ visuel (à droite) est projetée sur la surface du sc (à gauche) à l'aide d'une fonction logarithmique dans l'espace complexe. Les zones grisées à droite et à gauche représentent la même partie du champ visuel. (Repris de (COLAS, FLACHER et al., 2009).)

Durant l'expérience, les objets bougent de manière aléatoire et l'objectif est de se rappeler quels objets sont les cibles. Dans cette expérience, la scène visuelle est simplifiée pour permettre l'étude de la sélection de l'action en éliminant le plus possible les perturbations.

Nous continuons ce chapitre en décrivant le modèle que nous proposons avant d'en décrire les résultats (section 2.3) et de conclure (section 2.4).

2.2 Modèle

Le modèle que nous proposons est composé de deux parties. La première partie s'occupe de la perception et mémoire de la scène visuelle (modèle de représentation) tandis que la seconde s'intéresse à la sélection de l'endroit où regarder ensuite (modèles de décision). Nous proposons plusieurs modèles de décision que nous comparons à l'aide de données expérimentales afin de pouvoir tirer des conséquences.

2.2.1 Modèle de représentation

La partie de représentation, commune à tous les modèles de décision est donc une carte rétinotopique de la scène visuelle. Cette représentation est elle-même structurée en deux couches. La première, appelée *grille d'occupation*, est simplement l'entrée visuelle sans distinction entre les cibles et les distracteurs. La seconde couche est une mémoire de la position des seules cibles. C'est la représentation de la connaissance par l'observateur des positions des cibles, basée sur la grille d'occupation.

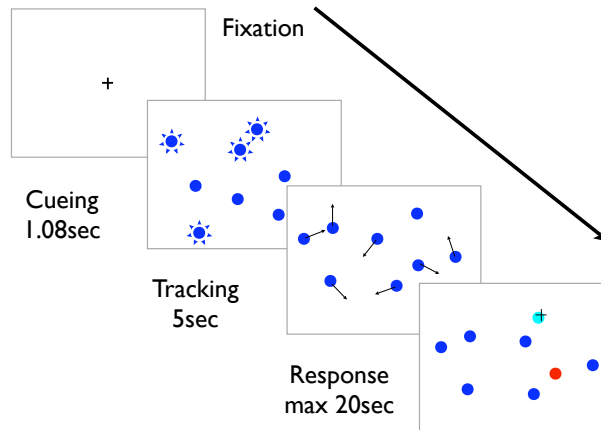


FIG. 2.3: Expérience de suivi multi-cibles. Un ensemble d'objets est présenté pendant 1,08 s : les cibles sont identifiées par leur clignotement ; les distraiteurs ne clignotent pas. Ensuite le clignotement cesse et tous les objets bougent de manière aléatoire pendant 5 s. Une fois les objets arrêtés, le participant doit identifier les cibles d'origine. Adapté de (PYLYSHYN et ANNAN, 2006) et repris de (COLAS, FLACHER et al., 2009).

Grille d'occupation

Les grilles d'occupation sont une des méthodes de représentation d'un environnement. Elles ont été introduites à l'origine pour représenter des obstacles dans des applications robotiques (ELFES, 1989). L'idée générale est de discrétiser l'environnement en une grille dont les cellules sont une variable indiquant la présence ou l'absence d'un obstacle. On peut donc voir une grille d'occupation comme une collection de distributions de probabilité sur la présence d'un obstacle dans chaque cellule de la grille.

Nous appliquons ce modèle à la présence d'objets dans la scène visuelle. Plus précisément, nous considérons un ensemble O de variables binaires $O_{(x,y)}^t$, pour chaque instant temporel $t \in \llbracket 0, t_{max} \rrbracket$ et chaque position $(x, y) \in \mathcal{G}$ avec \mathcal{G} une grille régulière en coordonnées logarithmique complexe dans le référentiel rétinocentré. Nous supposons que nous avons les entrées visuelles dans ce même espace de coordonnées, représentées par un ensemble V de variables binaires indiquant si un objet (cible ou distraiteur) est perçu dans la cellule correspondante. Finalement, nous considérons les mouvements des yeux précédemment effectués M^t de manière à permettre la ré-allocation telle qu'observée par les mémoires rétinocentrées corticales et sous-corticales.

Nous écrivons la distribution de probabilité conjointe sur toutes les variables en supposant l'indépendance des cellules d'occupation entre elles, conditionnellement à l'état précédent de la grille et au mouvement de la grille. On suppose de plus que chaque observation est indépendante des autres variables conditionnellement à l'état d'occupation

de la cellule correspondante. Ainsi, nous écrivons :

$$\begin{aligned} P(OVM) &= P(O^0) \prod_{t=1}^{t_{max}} P(O^t V^t M^t | O^{t-1}) \\ &= \prod_{(x,y) \in \mathcal{G}} P(O_{(x,y)}^0) \prod_{t=1}^{t_{max}} \left\{ P(M^t) \prod_{(x,y) \in \mathcal{G}} \left[P(O_{(x,y)}^t | M^t O^{t-1}) P(V_{(x,y)}^t | O_{(x,y)}^t) \right] \right\}. \end{aligned}$$

La mise à jour de la connaissance du modèle correspond à la question suivante :

$$\begin{aligned} P(O^t | V^{1:t} M^{1:t}) &\propto \prod_{(x,y) \in \mathcal{G}} P(V_{(x,y)}^t | O_{(x,y)}^t) \\ &\times \sum_{O^{t-1}} \left\{ \left[\prod_{(x,y) \in \mathcal{G}} P(O_{(x,y)}^t | M^t O^{t-1}) \right] P(O^{t-1} | V^{1:t-1} M^{1:t-1}) \right\}. \end{aligned} \quad (2.1)$$

Cette expression fait intervenir une somme sur tous les états possibles de la grille, ce qui est très coûteux en calculs. En remarquant que $P(O_{(x,y)}^t | M^t O^{t-1})$ est l'expression de la ré-allocation et qu'une cellule $O_{(x,y)}^t$ ne dépend réellement que du sous-ensemble de ses antécédents par le mouvement des yeux, on peut simplifier l'inférence sur toute la grille par un ensemble d'inférences plus simples :

$$\begin{aligned} P(O_{(x,y)}^t | V^{0:t} M^{1:t}) &\propto P(V_{(x,y)}^t | O_{(x,y)}^t) \\ &\times \sum_{\substack{O_{\mathcal{A}(x,y,M_t)}^{t-1} \\ \mathcal{A}(x,y,M_t)}} \left\{ \frac{P(O_{(x,y)}^t | M^t O_{\mathcal{A}(x,y,M_t)}^{t-1})}{\prod_{(x',y') \in \mathcal{A}(x,y,M_t)} P(O_{(x',y')}^{t-1} | V^{1:t-1} M^{1:t-1})} \right\}, \end{aligned}$$

où $\mathcal{A}(x, y, M^t)$ est l'ensemble des cellules (x', y') que le déplacement des yeux M^t amène en (x, y) .

Ainsi, l'inférence du modèle nécessite la spécification de trois distributions conditionnelles. $P(V_{(x,y)}^t | O_{(x,y)}^t)$ est une simple matrice indiquant qu'il y a une forte probabilité d'observer un objet quand il y en a un et réciproquement une faible probabilité lorsqu'il n'y en a pas. $P(O_{(x,y)}^t | M^t O_{\mathcal{A}(x,y,M_t)}^{t-1})$ est la ré-allocation de l'occupation en fonction des surfaces géométriques occupées dans les cellules antécédentes avec de l'incertitude rajoutée à cause du mouvement propre des objets. Enfin, $P(O_{(x',y')}^{t-1} | V^{1:t-1} M^{1:t-1})$ est simplement la distribution calculée à l'instant précédent ou une distribution *a priori* pour l'initialisation ($t = 0$).

Position des cibles

Le modèle précédent maintient une description de la scène visuelle sans distinction entre les cibles et les distracteurs. Pour introduire cette distinction, nous ajoutons un ensemble de variables T_i^t représentant la position de chaque cible $i \in \llbracket 1, N \rrbracket$ pour chaque instant $t \in \llbracket 0, t_{max} \rrbracket$ en coordonnées logarithmique complexe dans le référentiel rétinocentré. Ce type de variable est le moyen standard de représentation la position d'objets d'intérêt, par opposition à la grille d'occupation qui représentation la présence d'objets

en des lieux d'intérêts (les cellules). Dans ce dernier cas, il peut y avoir un nombre d'objets quelconque entre zéro et le nombre de cellules alors que nous souhaitons suivre un nombre restreint mais fixe de cibles.

Nous complétons le modèle précédent de grille d'occupation avec ces variables :

$$P(OVMT) = \prod_{(x,y) \in \mathcal{G}} P(O_{(x,y)}^0) \prod_{i=1}^N P(T_i^0) \times \prod_{t=1}^{t_{max}} \left[\begin{array}{l} P(M^t) \\ \times \prod_{(x,y) \in \mathcal{G}} \left[\begin{array}{l} P(O_{(x,y)}^t | M^t O^{t-1}) \\ \times P(V_{(x,y)}^t | O_{(x,y)}^t) \end{array} \right] \\ \times \prod_{i=1}^N P(T_i^t | M^t O^t T_i^{t-1}) \end{array} \right].$$

Les distributions $P(T_i^0)$ sont des distributions *a priori* fixées à partir des positions de départ des cibles telles qu'indiquées durant la phase d'identification des cibles. Le modèle dynamique des cibles $P(T_i^t | M^t O^t T_i^{t-1})$ est similaire au modèle dynamique des objets mais avec la grille d'occupation pour observation (voir COLAS, FLACHER et al., 2009, B.2).

À chaque instant, la position de chacune des cibles est calculée par l'inférence suivante :

$$P(T_i^t | V^{1:t} M^{1:t}) \propto \sum_{T_i^{t-1}} \left\{ P(T_i^{t-1} | V^{1:t-1} M^{1:t-1}) \times \sum_{O^t} \left[\begin{array}{l} P(T_i^t | M^t O^t T_i^{t-1}) \\ \times P(O^t | V^{1:t} M^{1:t}) \end{array} \right] \right\}, \quad (2.2)$$

avec $P(T_i^{t-1} | V^{1:t-1} M^{1:t-1})$ le résultat de cette même inférence à l'instant précédent et $P(O^t | V^{1:t} M^{1:t})$ le résultat de la question (2.1) à l'instant courant. On retrouve la même somme sur toutes les configurations de la grille qui peut être simplifiée de la même manière qu'au-dessus en ne considérant que les cellules antécédentes et en séparant les cellules.

Les deux questions (2.1) et (2.2) sont la connaissance actuelle sur la scène visuelle que l'on peut déduire des observations visuelles et des mouvements des yeux.

2.2.2 Modèles de décision

À l'aide de cette connaissance, on peut alors construire le modèle de décision du prochain endroit où regarder. Pour tester plusieurs hypothèses, nous proposons plusieurs modèles que nous comparons dans la [section 2.3](#).

En premier lieu, nous souhaitons vérifier que la représentation ci-dessus est utile à la prédiction des mouvements des yeux. Pour cela, nous allons comparer le meilleur modèle n'utilisant pas cette représentation avec un modèle qui l'utilise.

Ensuite nous souhaitons évaluer si l'incertitude, prise en compte explicitement, peut aider à la prédiction des mouvements des yeux. Nous allons donc comparer un modèle qui utilise cette incertitude avec un modèle qui ne l'utilise pas.

Ainsi nous avons besoin de définir trois modèles : un premier qui n'utilise pas la représentation (*constant*, π_A), un qui utilise la représentation mais pas explicitement l'incertitude (*position des cibles*, π_B) et un dernier qui utilise l'incertitude (*incertitude*, π_C). Chaque modèle π_k va calculer la distribution de probabilité sur la prochaine position des yeux $C^t \in \mathcal{G}$ pour chaque instant $t \in \llbracket 1, t_{max} \rrbracket$:

$$P(C^t | V^{1:t} M^{1:t} \pi_k)$$

Modèle constant

Le modèle constant sert juste de comparaison pour les autres modèles se basant sur la représentation. Nous le définissons comme indépendant à la fois de l'instant temporel, de la représentation, des entrées visuelles et des mouvements des yeux passés :

$$\forall t \in \llbracket 1, t_{max} \rrbracket, P(C^t | V^{1:t} M^{1:t} \pi_k) = P(C^t | \pi_A) = P(C^1 | \pi_A)$$

On peut montrer que le meilleur modèle $P(C^1 | \pi_A)$ pour prédire les mouvements des yeux est celui tel que la probabilité de bouger correspond à la fréquence observée de mouvement des yeux. Nous avons appris cette distribution à partir d'un sous-ensemble d'apprentissage des données expérimentales.

Modèle des positions des cibles

Le second modèle que nous proposons fait l'hypothèse que l'on tend à regarder près des cibles. Il inclut une distribution *a priori* identique au modèle constant et une vraisemblance similaire à une fusion de données :

$$P(CVMT | \pi_B) = \prod_{t=1}^{t_{max}} P(V^t M^t | \pi_B) \prod_{i=1}^N P(T_i^t | V^{1:t} M^{1:t} \pi_B) P(C^t | T^t \pi_B).$$

L'inférence peut s'écrire sous la forme :

$$P(C^t | V^{1:t} M^{1:t} \pi_B) \propto P(C^t | \pi_A) \times \prod_{i=1}^N \sum_{T_i^t} P(T_i^t | V^{1:t} M^{1:t}) \times P(T_i^t | C^t),$$

où $P(T_i^t | V^{1:t} M^{1:t})$ est le résultat de la question (2.2) et $P(T_i^t | C^t)$ une distribution gaussienne centrée sur C^t . Ce modèle n'est pas restreint à regarder les cibles mais peut regarder à un barycentre des cibles.

Modèle avec incertitude

Plus précisément, le modèle précédent sera amené à regarder plus probablement à un endroit le moins éloigné possible des masses de probabilité des cibles. Si une des cibles présente beaucoup d'incertitude, la décision sera probablement moins influencée par cette cible que par les autres. L'incertitude a donc un certain rôle dans ce modèle en ce que le regard la fuit.

Dans le troisième modèle, nous proposons d'inclure explicitement l'incertitude comme une variable sur laquelle raisonner. L'idée est qu'il est plus efficace de collecter de l'information là où elle manque plutôt que là où il y a moins d'incertitude.

Ainsi, nous introduisons un nouvel ensemble I de variables $I_{(x,y)}^t \in [0, 1]$ représentant un indice d'incertitude en la cellule $(x, y) \in \mathcal{G}$ à l'instant $t \in \llbracket 1, t_{max} \rrbracket$. N'importe quel indice convient à partir du moment où il est corrélé à l'incertitude. Dans notre cas, comme l'occupation d'une cellule est une variable binaire, nous avons choisi la probabilité de cette variable comme indice d'incertitude. Plus cet indice est proche de $\frac{1}{2}$ plus l'incertitude est grande. Un autre choix aurait pu être de prendre l'entropie de la distribution.

Comme pour le modèle précédent, nous spécifions le modèle avec comme distribution *a priori* celle issue du modèle précédent pour rajouter la prise en compte de l'incertitude comme vraisemblance :

$$P(CVMI|\pi_C) = \prod_{t=1}^{t_{max}} \left\{ P(V^t M^t | \pi_C) P(C^t | V^{1:t} M^{1:t} \pi_B) \times \prod_{(x,y) \in \mathcal{G}} P(I_{(x,y)}^t | C^t \pi_C) \right\}$$

L'inférence peut s'écrire sous la forme :

$$P(C^t | V^{1:t} M^{1:t} I^{1:t} \pi_C) \propto P(C^t | V^{1:t} M^{1:t} \pi_B) P(I_{C^t}^t | C^t \pi_C),$$

dans laquelle $I_{(x,y)}^t = P(O_{(x,y)}^t | V^{1:t} M^{1:t})$ de la question (2.1) et $P(I_{(x,y)}^t | C^t) = (x, y) \pi_C$ est une distribution bêta indiquant que I est plus probablement proche de $\frac{1}{2}$.

Ainsi ces trois modèles de décision sont structurés de manière à ce que chacun est le raffinement du modèle précédent.

2.3 Expérience

Nous souhaitons comparer ces modèles de manière quantitative et commençons donc pas définir la mesure qui convient. Ensuite nous présentons les résultats globaux des trois algorithmes avant de montrer plus en détails des situations typiques permettant d'illustrer les différences.

2.3.1 Méthode de comparaison

Chacun de nos modèle calcule une distribution de probabilité sur le prochain mouvement des yeux (voir la figure 2.4). On peut ainsi calculer, pour chacun des modèles, la probabilité affectée au mouvement enregistré dans une situation donnée et aussi la probabilité affectée à l'ensemble des mouvements enregistrés.

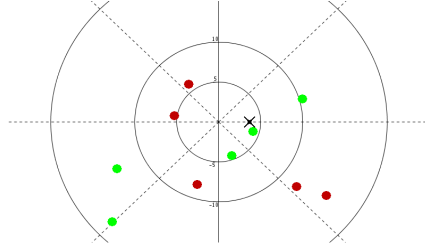
Mais les valeurs de probabilité en elle-même n'ont pas une sémantique absolue. Par contre leur comparaison entre des modèles (qui portent sur les mêmes variables) indique quel modèle est le meilleur prédicteur des mouvements des yeux. C'est la méthode de *maximum de vraisemblance*.

Cependant en pratique, à mesure que le nombre de données augmente, la vraisemblance d'un modèle décroît généralement de manière exponentielle vers zéro, et le rapport des vraisemblances entre deux modèles diverge ou converge vers zéro de la même manière. Ainsi nous comparons nos modèles en utilisant la moyenne géométrique du rapport de vraisemblance des mouvements des yeux observés. La moyenne géométrique converge vers une valeur non nulle à mesure que le nombre d'essais augmente.

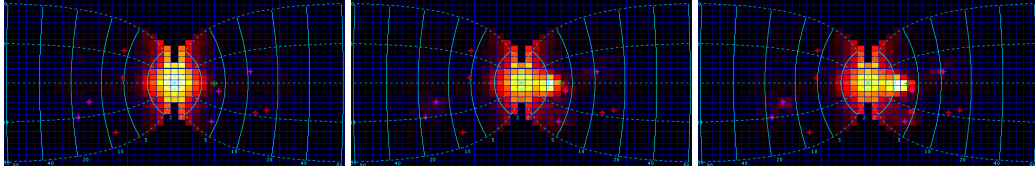
Plus formellement, la vraisemblance d'un modèle π donné est :

$$\prod_{n=1}^N \prod_{t=1}^{t_{max}} P([C^t = c_n^{t+1}] | v_n^{1:t} c_n^{1:t} \pi),$$

avec N le nombre d'essais et c_n^t le mouvement des yeux à l'instant t enregistré pendant l'essai n .



(a) Exemple de champ visuel. La scène est présentée dans le référentiel de la rétine. Les cibles sont en rouge sombre et les distraiteurs en vert clair. Le mouvement du sujet est représenté par la croix noire.



(b) Distribution pour π_A . (c) Exemple de distribution pour π_B . (d) Exemple de distribution pour π_C .

FIG. 2.4: Exemples de distribution sur le mouvement des yeux par les différents modèles proposés. Les distributions sont représentées en coordonnées logarithmiques complexes dans le référentiel de la rétine. Les deux moitiés du champ visuel sont représentées côte-à-côte. En cyan, continu : les lignes d'iso-excentricité ; en cyan, pointillé : les lignes d'iso-direction. La couleur des cellules indique la probabilité croissante depuis les couleurs foncées aux plus claires. (Repris de (COLAS, FLACHER et al., 2009).)

Notre mesure de comparaison entre deux modèles π_1 et π_2 est donc :

$$\mu(\pi_1, \pi_2) = \sqrt[N]{\frac{\prod_{n=1}^N \prod_{t=1}^{t_{max}} P([C^t = c_n^{t+1}] | v_n^{1:t} c_n^{1:t} \pi_1)}{\prod_{n=1}^N \prod_{t=1}^{t_{max}} P([C^t = c_n^{t+1}] | v_n^{1:t} c_n^{1:t} \pi_2)}}.$$

2.3.2 Résultats globaux

Le jeu de données regroupe 11 participants avec 110 essais chacun pour un total de 1210 essais (TANNER, CANTO-PEREIRA et BÜLTHOFF, 2007). Chaque essai a été discrétisé en $t_{max} = 24$ étapes pour un total de 29 040 points de donnée. Une partie du jeu de données (124 essais choisis aléatoirement) a été utilisée pour déterminer les paramètres des divers modèles et les résultats ont été calculés sur le reste des essais ($N = 1089$).

La [table 2.1](#) montre les comparaisons entre nos modèles. En premier lieu, on remarque que le modèle constant est bien moins probable que les deux autres (d'un facteur respectif de 280 et 320). Cela confirme que la représentation de la [section 2.2.1](#) a une utilité dans la décision du prochain mouvement des yeux.

La [table 2.1](#) montre aussi que le modèle qui prend en compte l'incertitude est meilleur que le modèle des cibles d'en moyenne 14%. Cela va dans le sens de notre hypothèse que l'incertitude est utile pour prévoir les mouvements des yeux.

Comme indiqué au-dessus, le choix de la moyenne géométrique permet d'avoir des

	Constant	Cibles	Incertitude
Constant	1	280	320
Cibles	$3,5 \times 10^{-3}$	1	1,14
Incertitude	$3,1 \times 10^{-3}$	0,87	1

TAB. 2.1: Comparaison entre les modèles. Un nombre supérieur à 1 signifie que le modèle de la colonne est meilleur que le modèle de la ligne.

rapports stables indépendamment du nombre d'essais. Le rapport de vraisemblance entre le modèle avec et sans incertitude est de $4,9 \times 10^{63}$. Avec la moitié des essais, ce rapport est la racine carrée : $7,0 \times 10^{31}$. Cela illustre bien que le rapport de vraisemblance n'est pas une mesure stable et justifie notre choix.

2.3.3 Situations typiques

Les résultats montrent donc que le modèle prenant en compte explicitement l'incertitude explique globalement mieux les mouvements des yeux que les deux autres. Cependant, si on regarde les mouvements individuels, ce n'est pas toujours le cas. Dans cette section, sont analysées certaines situations pour donner une meilleure compréhension des modèles.

Situations pour lesquelles $\pi_C > \pi_B$

En regardant les instants où le rapport de probabilité est le plus grand entre le modèle avec incertitude et celui des cibles on peut observer deux types de configurations où l'incertitude est un meilleur prédicteur. La première catégorie regroupe les situations dans lesquelles une cible et un distracteur sont relativement proches (figure 2.5a). Dans le cas, le modèle des cibles n'est attiré que par la cible alors que le modèle avec l'incertitude est attiré par l'incertitude autour des deux objets.

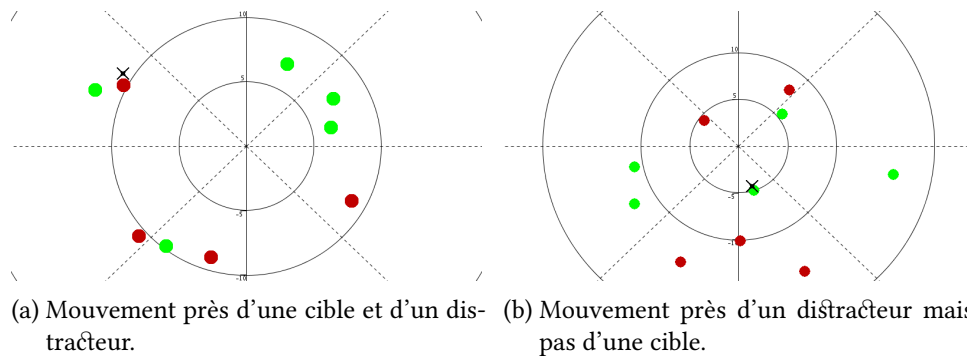


FIG. 2.5: Mouvements mieux prédits par le modèle avec incertitude qu'avec seulement les positions des cibles. La scène est présentée dans le référentiel de la rétine. En rouge sombre : les cibles ; en vert clair : les distracteurs ; croix noire : mouvement du sujet. (Adapté de (COLAS, FLACHER et al., 2009).)

La seconde catégorie regroupe les situations pour lesquelles le mouvement des yeux s'est fait près d'un distracteur (figure 2.5b). Dans ce cas, le modèle des cibles n'a aucune raison de regarder à cet endroit alors que le modèle d'incertitude peut réduire l'incertitude liée au distracteur.

Situations pour lesquelles $\pi_B > \pi_C$

À l'opposé, il existe des cas où le modèle ne prenant pas en compte l'incertitude mais simplement la position des cibles prédit mieux le mouvement des yeux. Le cas principal est lorsque le mouvement des yeux se fait entre plusieurs cibles (figure 2.6). Dans ce cas, à l'endroit du mouvement des yeux il y a peu d'incertitude qu'il n'y ait rien. En conséquence, le modèle d'incertitude abaisse la probabilité relativement à d'autres endroits où il y a plus d'incertitude (au voisinage des objets).

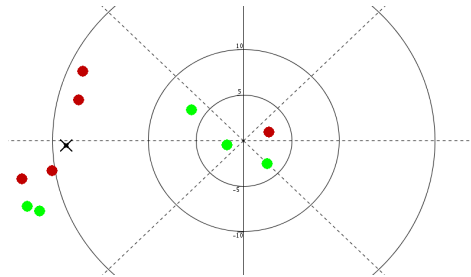


FIG. 2.6: Mouvement des yeux mieux prédit par le modèle prenant en compte la position des cibles qu'avec en plus l'incertitude. La scène est présentée dans le référentiel de la rétine. En rouge sombre : les cibles ; en vert clair : les distracteurs ; croix noire : mouvement du sujet. (Adapté de (COLAS, FLACHER et al., 2009).)

Situations pour lesquelles $\pi_A > \pi_B, \pi_C$

Enfin, le meilleur des modèles est parfois le modèle constant. Cela arrive en général lors de fixations (pas de mouvements) en dehors des objets. La plupart des saccades se font à proximité des objets du champ visuel, mais elles sont suivies de mouvements de suivi (continus et de faible amplitude) ou de fixation. Comme les objets bougent, en cas de fixation, les positions suivantes du regard seront de plus en plus loin des objets. Ainsi ces mouvements de faible amplitude sont souvent mieux capturés par la distribution *a priori* (revoir la figure 2.4b) que par les modèles qui cherchent à regarder à proximité des cibles ou de l'incertitude (qui est faible lorsque l'on sait qu'il n'y a rien). La figure 2.7 montre deux exemples de cette situation. À noter que dans la situation représentée dans la figure 2.7b, une saccade a eu lieu vers le groupe des trois cibles de droite à l'étape de temps suivante.

2.4 Discussion et conclusion

En conclusion, nous proposons un modèle en deux parties : une représentation de la scène visuelle et un modèle de décision à partir de cette représentation. La représentation suit à la fois l'occupation de la scène visuelle (s'il y a une cible ou un distracteur à un

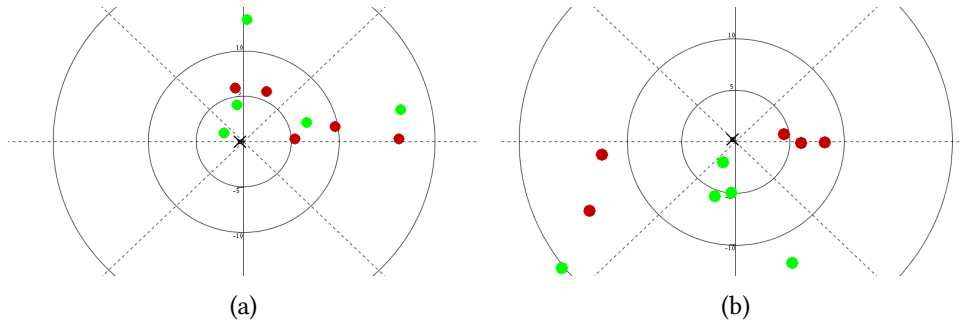


FIG. 2.7: Mouvements mieux prédits par le modèle constant. La scène est présentée dans le référentiel de la rétine. En rouge sombre : les cibles ; en vert clair : les distracteurs ; croix noire : mouvement du sujet. (Adapté de (COLAS, FLACHER et al., 2009).)

endroit donné) et la position de chacune des cibles. À l'aide de cette représentation nous avons pu tester plusieurs modèles de décision et montrer que le modèle qui prend en compte explicitement l'incertitude est le plus vraisemblable vis-à-vis des mouvements des yeux enregistrés sur des sujets d'une expérience de psychophysique.

De plus, la distribution de fréquence des mouvements montre que, la plupart du temps, les mouvements des yeux sont de faible amplitude, c'est-à-dire soit des fixations ou des mouvements de poursuite des objets. Dans ces cas, la vraisemblance du modèle constant est comparable ou même meilleure à celle des autres. Ainsi, la différence est due aux événements saccadiques, quand le modèle avec les positions des cibles ou celui avec l'incertitude ont une bien meilleure vraisemblance que ce modèle constant dont la probabilité décroît avec l'excentricité. La différence entre les modèles avec ou sans l'incertitude est due au filtrage par le modèle promouvant les endroits avec plus d'incertitude. La différence est moins importante qu'avec le modèle constant car l'incertitude des différentes cibles est souvent similaire. Il serait intéressant de changer le stimulus de manière à pouvoir manipuler l'incertitude avec plus de finesse.

Le stimulus est adapté de la tâche classique de suivi de plusieurs cibles qui a d'abord été développée pour étudier les phénomènes d'attention. Notre modèle utilise un ensemble de variables pour suivre les positions des cibles. Cet ensemble de variable est fixe et fini, ce qui veut dire que notre modèle ne peut pas suivre plus de cibles qu'il n'a de variable. Le nombre choisi a été suggéré par la performance des participants humains pour cette expérience particulière. Ce nombre n'est pas fixe et semble dépendre de facteurs divers tels que la vitesse et la distance entre les objets (ALVAREZ et FRANCONERI, 2007). De plus, chacune de nos variables de position couvre l'ensemble du champ visuel même si certains travaux semblent indiquer que la capacité de représentation est séparée dans chaque héli-champ (ALVAREZ et CAVANAGH, 2005). Il serait intéressant de tester cela dans notre modèle avec un ensemble de variables de position de cible pour la partie droite et un autre pour la partie gauche. Cependant, à cause des mouvements des yeux et des cibles elles-mêmes, celles-ci changent parfois de côté, ce qui nécessite un mécanisme supplémentaire de communication entre ces ensembles de variables.

Enfin, une des caractéristiques principales de notre modèle est de mener tous les calculs et les représentations en coordonnées logarithmiques tel qu'observé dans les structures cérébrales concernées. De manière inattendue, nous avons constaté dans les données psy-

chophysiques que la distribution des objets est assez uniforme dans cet espace. Cela n'est pas le cas avec des mouvements des yeux quelconques ou en absence de mouvement des yeux et implique donc une stratégie particulière pour les mouvements des yeux. Une interprétation pourrait être que les mouvements des yeux sont choisis de manière à optimiser l'utilisation de l'espace de représentation : c'est-à-dire distribuer les objets de manière uniforme dans ces coordonnées logarithmiques. Cela semble être une confirmation indirecte que les mouvements des yeux sont gouvernés par des structures utilisant ces coordonnées.

3 Alignement de nuages de points en robotique

3.1 Introduction du problème

En robotique, la problématique de la perception est légèrement différente en ce qu'il s'agit de définir à la fois une représentation et un processus permettant de construire cette représentation à partir des données des capteurs. Je me suis intéressé plus précisément à la construction de représentations en 3D sous la forme de nuages de points, à l'aide de données issues d'un scanner laser. Lorsque l'on connaît précisément la position du capteur, il est possible d'accumuler les points mesurés à chaque instant en un gros nuage de points représentant la géométrie de l'environnement. Le problème est que cette position n'est souvent pas connue de manière précise et absolue ce qui donne alors des représentations déformées.

Une manière classique pour obtenir une meilleure représentation est d'aligner les nuages de points de manière à ce que les points correspondant aux mêmes endroits de l'environnement soient superposés. La famille d'algorithmes la plus utilisée est l'*iterative closest point* (ICP), qui consiste en l'alternance de phases d'association par la méthode du plus proche voisin et de minimisation de l'erreur d'alignement.

Si l'on connaissait l'association correcte, c'est-à-dire que les deux points d'une paire correspondent bien au même endroit physique, alors on pourrait calculer directement la transformation minimisant l'erreur d'alignement. Cependant, cette association est inconnue en premier lieu et il faudrait optimiser conjointement l'association et la transformation.

L'ICP contourne cette difficulté en faisant cette optimisation de manière itérative en approchant à chaque étape l'association inconnue par les plus proches voisins. Ainsi le nuage de points va se déplacer à mesure de l'optimisation et à chaque fois changer l'association. Ce processus converge¹ et amène vers un optimum local.

Ce processus générique a été découvert et utilisé dans plusieurs champs de recherche tels que la vision par ordinateur, la robotique ou l'imagerie médicale. Si on se restreint à l'alignement de nuages de points 3D, on trouve déjà de nombreuses applications : la reconstruction d'objets (CHEN et MEDIONI, 1991), l'inspection (BESL et MCKAY, 1992), la médecine (CHAMPLEBOUX et al., 1992) ou la navigation par des véhicules autonomes (ZHANG, 1994).

En robotique, l'utilisation des algorithmes de *simultaneous localization and mapping* (SLAM) (THRUN, BURGARD et FOX, 1998 ; NÜCHTER et al., 2004) et l'introduction, à la fin des années 90, des scanners laser (LU et MILIOS, 1997 ; PFISTER et al., 2002) ont amené

1. Il n'y a pas nécessairement convergence en présence de bruit mais en pratique les corrections deviennent relativement petites.

de très nombreuses publications proposant des variantes sur cet algorithme. RUSINKIEWICZ et LEVOY (2001) a proposé un premier travail sur la comparaison des algorithmes d'ICP basé sur des scans simulés pour mettre en avant les contraintes spatiales et le bruit capteur. Mais avec l'arrivée de nouveaux capteurs moins chers tels que les scanners laser Sokuiki d'Hokuyo (YOSHITAKA et al., 2006) ou les caméras de profondeur de type Kinect (POMERLEAU, MAGNENAT et al., 2011 ; HENRY et al., 2012), on observe plusieurs dizaines de publications par an proposant des variations sur l'ICP.

Cette profusion de contributions dans la littérature rend difficile la formation d'une vision globale et d'une comparaison des différentes variations, d'autant que la méthodologie scientifique n'est pas toujours bien suivie (AMIGONI, REGGIANI et SCHIAFFONATI, 2009). Ainsi, nous proposons une méthodologie basée sur la comparaison des algorithmes sur des jeux de données communs (POMERLEAU, COLAS, SIEGWART et MAGNENAT, 2013 ; POMERLEAU, LIU et al., 2012). Nous avons aussi consolidé le formalisme et effectué une grande revue de la littérature et des applications possibles de l'ICP (POMERLEAU, COLAS et SIEGWART, 2015). Cela a constitué le travail de thèse de POMERLEAU (2013).

Dans ce chapitre, nous commençons par exposer la formalisation de l'algorithme et de la méthodologie de comparaison avant de passer en revue les applications et résultats expérimentaux.

3.2 Formalisation

L'ICP est un algorithme itératif qui peut présenter trois types de problèmes. Il peut être lent s'il fait de nombreuses itérations sur un grand nombre de points. Il peut converger avec une petite erreur due aux caractéristiques locales des nuages de points telles que le bruit sur les mesures ou leur répartition (variations de densité). Enfin, il peut converger vers un minimum local différent de la solution escomptée.

Les nombreuses variations de l'ICP proposées s'attachent essentiellement à circonvier à ces problèmes observés en pratique en effectuant une ou souvent plusieurs modifications de l'algorithme le plus simple. Une première étape nécessaire à la comparaison de ces propositions est donc de les réinterpréter à l'aide d'une grille de lecture modulaire permettant de mettre en avant les différences (section 3.2.1). Nous présenterons ensuite une méthode pour faire la comparaison elle-même (section 3.2.3).

3.2.1 Algorithme générique

Nous avons déjà énoncé l'objectif général de l'algorithme comme étant la minimisation globale de l'erreur d'alignement entre les nuages de points. Cette minimisation globale est approchée par la minimisation itérative de cette erreur d'alignement en supposant l'appariement des points par la méthode des plus proches voisins.

Plus précisément, les nuages de points peuvent contenir des informations géométriques (position du point, vecteur normal, etc.) qui sont affectées par les transformations et des informations non géométriques (couleur, identificateur, température, etc.) appelées parfois descripteurs.

Des filtres sont souvent appliqués sur les nuages de points, soit pour supprimer des points n'apportant pas plus d'information pour réduire le temps de calcul, soit pour ajouter de l'information au point, telles que des propriétés locales des points : densité, cour-

bure, vecteur normal, etc. Ces dernières informations ne sont généralement pas présentes dans les données brutes du capteur mais peuvent permettre une meilleure association des points ou un meilleur calcul de l'erreur d'alignement.

Plus formellement, soit ${}^{\mathbb{A}}\mathcal{P}$ le nuage de points d'entrée dans le référentiel \mathbb{A} à associer au nuage de point de référence ${}^{\mathbb{B}}\mathcal{Q}$ dans le référentiel \mathbb{B} . Le but de l'alignement se ramène à estimer la transformation ${}^{\mathbb{B}}\mathcal{T}$ en minimisant l'erreur $\text{error}(\mathcal{P}', \mathcal{Q})$:

$${}^{\mathbb{B}}\hat{\mathcal{T}} = \arg \min_{\mathcal{T}} \left(\text{error} \left(\mathcal{T} \left({}^{\mathbb{A}}\mathcal{P} \right), {}^{\mathbb{B}}\mathcal{Q} \right) \right), \quad (3.1)$$

dans laquelle $\mathcal{T}(\mathcal{P})$ est l'application de la transformation géométrique \mathcal{T} au nuage d'entrée \mathcal{P} .

Une spécificité du problème d'alignement est que la fonction d'erreur est en fait calculée sur des paires de points issues de l'association entre les deux nuages de points. L'association la plus simple est de prendre le point de \mathcal{Q} le plus proche de chacun des points de \mathcal{P} . Elle peut être basée sur la seule position des points, mais aussi sur les valeurs des descripteurs dont l'invariance par rapport à la transformation est une propriété utile pour cette étape.

Si l'on note $\mathcal{M} = \text{match}(\mathcal{P}, \mathcal{Q}) = \{(\mathbf{p}, \mathbf{q}) \mid \mathbf{p} \in \mathcal{P}, \mathbf{q} \in \mathcal{Q}\}$ l'ensemble des paires de points associés entre \mathcal{P} et \mathcal{Q} , l'erreur d'alignement est de la forme :

$$\text{error}(\mathcal{P}, \mathcal{Q}) = \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{M}} d(\mathbf{p}, \mathbf{q}),$$

avec $d(\mathbf{p}, \mathbf{q})$ une mesure de distance entre les points \mathbf{p} et \mathbf{q} .

Pour rendre cette fonction plus robuste, des associations aberrantes peuvent être identifiées et supprimées des points associés. De plus, on introduit parfois des poids destinés à faire varier l'influence de certaines paires de points dans la fonction d'erreur :

$$\text{error}(\mathcal{P}, \mathcal{Q}) = \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{M}} w(\mathbf{p}, \mathbf{q}) d(\mathbf{p}, \mathbf{q}).$$

Il est clair que si l'association est correcte, minimiser cette erreur donne la meilleure estimation de ${}^{\mathbb{B}}\mathcal{T}$ (équation 3.1). Cependant, sauf si les descripteurs associés aux points sont suffisamment discriminants (comme c'est le cas avec les descripteurs visuels), l'association n'est généralement pas parfaite. L'idée de l'ICP est que même si l'association n'est pas la bonne, minimiser l'erreur d'alignement rapproche tout de même les nuages de points ce qui amène, à l'étape suivante, à une meilleure association.

Concrètement, on construit donc une séquence de transformations ${}_{i-1}^i\mathcal{T}$ successivement appliquées à \mathcal{P} . À une étape donnée, on définit un ensemble de paires \mathcal{M}_i de points associés à partir de la position courante des points de \mathcal{P} . On calcule alors une nouvelle transformation ${}_{i-1}^i\mathcal{T}$ basée sur cette association :

$${}_{i-1}^i\mathcal{T} \leftarrow \arg \min_{\mathcal{T}} \left(\text{error} \left(\mathcal{T} \left({}_i\mathcal{P}' \right), \mathcal{Q}' \right) \right).$$

Enfin, l'estimation de la transformation entre les deux nuages de points de départ s'obtient donc à l'aide de la composition de toutes les transformations intermédiaires :

$${}^{\mathbb{B}}\hat{\mathcal{T}} = \left(\bigcirc_i {}_{i-1}^i\mathcal{T} \right) \circ \mathcal{T}_{init},$$

3 Aligement de nuages de points en robotique

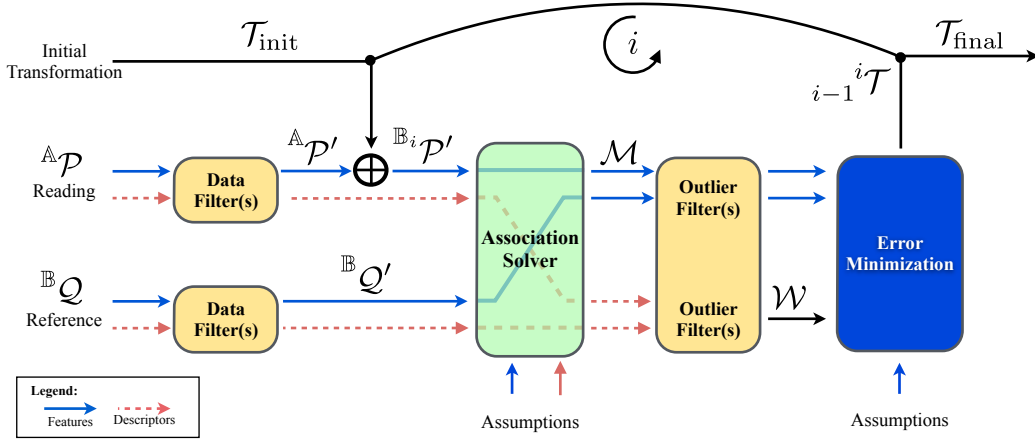


FIG. 3.1: Flux de données dans l'algorithme d'ICP (repris de (POMERLEAU, COLAS et SIEGWART, 2015)).

avec $\bigcirc_{i-1}^i T = \dots \circ {}_2^3 T \circ {}_1^2 T$ la composition itérative des transformations et T_{init} une transformation initiale.

En résumé, l'algorithme générique peut être résumé par l'**algorithme 1** et représenté graphiquement par la **figure 3.1**.

ALG. 1 Algorithme générique d'ICP.

Require : ${}^A P$
Require : ${}^B Q$ ▷ référence
Require : T_{init} ▷ transformation initiale
 ${}^A P' \leftarrow \text{datafilter}({}^A P)$ ▷ filtrage d'entrée
 ${}^B Q' \leftarrow \text{datafilter}({}^B Q)$ ▷ filtrage d'entrée
 ${}_{i-1}^i T \leftarrow T_{init}$
repeat
 ${}^i P' \leftarrow {}_{i-1}^i T({}^{i-1} P')$ ▷ déplacement du nuage
 $M_i \leftarrow \text{match}({}^i P', Q')$ ▷ association des points
 $W_i \leftarrow \text{outlier}(M_i)$ ▷ filtrage des associations
 ${}_{i+1}^i T \leftarrow \arg \min_T (\text{error}(T({}^i P'), Q'))$
until convergence
Ensure : ${}^B \hat{T} = \left(\bigcirc_{i-1}^i T \right) \circ T_{init}$

3.2.2 Modules

La **figure 3.1** montre l'architecture générique de l'ICP; chacun des éléments ou des modules au sein de ce schéma a donné lieu à des variations dans la littérature. Une étude plus en détails est disponible (POMERLEAU, COLAS et SIEGWART, 2015) et je ne fais ici que donner les grandes lignes.

Sources de données

Les sources des données pour les nuages de points sont fortement dépendantes de l'application. Précédemment les applications en robotique utilisaient principalement un scanner laser en deux dimensions (2D) pour faire de la localisation en intérieur ou en extérieur (YOSHITAKA et al., 2006 ; BOSSE et ZLOT, 2008).

Plus récemment, les capteurs et la puissance de calcul ont permis de travailler en 3D avec des scanners laser 2D mis en rotation, des scanners laser 3D (tels que les Velodyne HDL-64 ou HDL-32) ou des caméras de profondeur (telles que les Microsoft Kinect ou Asus Xtion) (SEGAL, HAEHNEL et THRUN, 2009 ; ARMESTO, MINGUEZ et MONTESANO, 2010 ; POMERLEAU, MAGNENAT et al., 2011 ; MOOSMANN et STILLER, 2011).

Transformations

Espace des transformations Dans de nombreuses applications en robotique, les transformations considérées sont les isométries directes, c'est-à-dire les transformations rigides de l'espace duquel est issu les points. Lorsqu'intervient un problème d'échelle, typiquement avec des caméras, on peut élargir aux similitudes directes. Enfin, on trouve aussi des transformations non-rigides, par exemple pour des applications médicales (MAURER et al., 1996 ; FELDMAR et AYACHE, 1996) ; on parle alors d'alignement flexible.

Transformation initiale Le choix de la transformation initiale est un élément déterminant de la performance de l'algorithme, surtout en l'absence de descripteurs non géométriques. Si elle est trop éloignée de la transformation réelle, l'algorithme peut se retrouver piégé dans un minimum local.

La transformation initiale peut provenir d'une source externe. C'est le cas, par exemple, lorsqu'elle est fournie par des opérateurs (GODIN, RIOUX et BARIBEAU, 1994 ; PULLI, 1999) mais aussi dans des solutions robotiques plus intégrées où l'on utilise les informations des autres capteurs du robot (odométrie, centrale inertielle, GPS, etc.) dont la fusion peut donner une approximation acceptable. Dans certains algorithmes de SLAM à base de filtre à particules, chaque particule correspond à une pose du robot qui correspond à une transformation initiale (GRISSETTI, STACHNISS et BURGARD, 2005 ; GRISSETTI, STACHNISS et BURGARD, 2007).

La transformation initiale peut aussi provenir d'un alignement des nuages de points effectué avec des paramètres différents. L'exemple classique est une stratégie de raffinement où les nuages de points commencent par être décimés pour laisser peu de points qui permettent un alignement grossier, mais rapide, puis on remet des points à chaque étape pour obtenir une transformation plus précise (ZHANG, 1994 ; JOST et HÜGLI, 2002 ; MAGNUSSON et al., 2009 ; BOSSE et ZLOT, 2009a). Les critères de fin d'itérations sont aussi adaptés pour réduire le temps de calcul en sachant que le résultat sera affiné ensuite.

Enfin, la transformation initiale peut provenir d'un alignement différent. Il s'agit typiquement d'un alignement basé sur des descripteurs qui sera alors indépendant de la transformation initiale. Ensuite une ou plusieurs étapes d'alignement sont utilisés en prenant en compte la position des points. On trouve des méthodes à deux (GODIN, RIOUX et BARIBEAU, 1994 ; BOSSE et ZLOT, 2008 ; TSAI et al., 2010) ou trois couches (BOSSE et ZLOT, 2009a) mais rien n'empêche d'imager des systèmes plus profonds.

Filtrage d'entrée

Il y a deux objectifs possibles au filtrage des nuages de points. On peut améliorer l'association en décorant les points avec des caractéristiques supplémentaires et l'on peut réduire la taille des nuages de points ce qui réduit le temps de calcul.

Décoration Les nuages de points issus des capteurs ne contiennent en général que la position des points et, dans le cas des caméras de profondeur, éventuellement la couleur. Un filtrage classique est de calculer le vecteur normal au plan local en chaque point (PULLI, 1999; JOST et HÜGLI, 2003; BOSSE et ZLOT, 2009a). On peut alors l'utiliser pour une meilleure association et une métrique d'erreur plan-à-plan ou point-à-plan plus robuste, surtout en environnement structuré.

On peut généraliser en utilisant des techniques d'inférence structurelle qui recouvrent, à partir des points, des caractéristiques géométriques d'une surface (ou, en 2D, d'une courbe). On peut s'intéresser au plan tangent à la surface, caractérisé par son vecteur normal ou même à une quadrique tangente avec les courbures suivant ses axes principaux. Il va de soi que plus l'ordre est élevé plus ces informations sont distinctives, mais aussi plus l'incertitude sur ces informations est grande et donc sensible au bruit.

Sous-échantillonnage En plus d'améliorer l'alignement, ces informations peuvent être utilisées pour sous-échantillonner les nuages de points de manière à réduire le temps de calcul. En effet, on peut choisir de supprimer les points les moins récents (car pouvant ne plus être d'actualité), ceux ayant été observés avec un grand angle d'incidence (et donc avec plus d'incertitude sur leur position), ou ceux ayant la plus grande densité (car plus redondants).

Enfin, on peut encore réduire le nombre de point de manière aléatoire (probabilité uniforme pour un point d'être choisi) (JOST et HÜGLI, 2003) ou essayer de rendre plus uniforme la densité par des méthodes géométriques comme des grilles uniformes (MAGNUSSON et al., 2009; BOSSE et ZLOT, 2009a) ou des *octrees* (FAIRFIELD et WETTERGREEN, 2009; HORNING et al., 2013).

Association

Lors de l'association, on suppose que les nuages de points sont exprimés dans le même repère et le but est d'obtenir un ensemble de paires de points.

Méthodes d'association L'association est généralement effectuée en cherchant, pour chaque point du nuage d'entrée, le plus proche dans le nuage de référence. La mesure de distance utilisée peut être simplement la distance euclidienne entre les positions des points (PULLI, 1999) mais aussi la distance entre le point d'entrée et le plan (un point avec son vecteur normal) (CHAMPLEBOUX et al., 1992) ou la quadrique (un point et ses courbures) du nuage de référence (FELDMAR et AYACHE, 1996). On trouve aussi des distances de Mahalanobis (STEWART, TSAI et ROYSAM, 2003) ou la statistique du test du χ^2 (TSAI et al., 2010).

L'association peut se faire aussi à partir des informations non géométriques (BOSSE et ZLOT, 2009a) C'est le cas le plus couramment en vision avec les descripteurs attachés aux

points d'intérêt (LOWE, 2004). L'avantage est alors d'être indépendant de la transformation initiale, mais on risque d'obtenir des paires incorrectes en cas d'éléments répétitifs dans l'environnement (échiquier d'étalonnage mais aussi des façades de bâtiments avec des fenêtres identiques).

On peut aussi construire une distance basée à la fois sur les informations géométriques et non géométriques, comme l'orientation (JOST et HÜGLI, 2003 ; BOSSE et ZLOT, 2009a) ou l'intensité du laser (GODIN, RIOUX et BARIBEAU, 1994 ; YOSHITAKA et al., 2006). On introduit alors un facteur de conversion entre la distance des deux espaces.

Un autre élément à prendre en compte dans la phase d'association est la direction d'association et le nombre de ses associations. Si l'on cherche souvent le point le plus proche dans le nuage de référence pour chaque point d'entrée on peut aussi faire dans l'autre sens (ce qui peut donner des résultats différents) voire aussi utiliser les deux directions (PULLI, 1999 ; GODIN, RIOUX et BARIBEAU, 1994). On peut aussi prendre plus que simplement le point le plus proche, par exemple tous les points plus proches d'un certain facteur que le point le plus proche (STEWART, TSAI et ROYSAM, 2003). On trouve parfois la matrice complète des associations, par exemple pour la fermeture de boucle (BOSSE et ZLOT, 2009b).

Optimisation À part dans ce dernier cas, l'association est une instance du problème de la recherche du point le plus proche (*nearest neighbor search*) dont la complexité est en $O(n \times m)$ où n et m sont respectivement les nombres de points dans les nuages d'entrée et de référence. C'est *a priori* la partie la plus lente de l'algorithme et c'est pourquoi de nombreux papiers s'intéressent à des variantes de cette recherche.

La méthode la plus courante est d'utiliser une structure de donnée de partition de l'espace, typiquement le *kD tree* (arbre k -dimensionnel). La complexité de l'association est alors réduite à $O(n \log m)$ une fois la structure construite en $O(m \log m)$. La recherche se fait en descendant jusqu'à la feuille correspondant au point dont on cherche les voisins et ensuite en remontant l'arbre pour savoir s'il faut regarder dans les feuilles voisines (si le plan de séparation est plus proche du point cherché que le point trouvé, il faut vérifier qu'un point plus proche n'existe pas de l'autre côté). Pour réduire la complexité de cette phase de vérification, on utilise parfois une méthode approchée en garantissant que le point retourné n'est pas plus éloigné du point demandé d'un certain facteur par rapport au point le plus proche dans la structure (ARYA et MOUNT, 1993).

Cette méthode se base sur la distance euclidienne dans un certain espace. Lorsque la dimension de l'espace devient trop grande, comme c'est le cas pour les descripteurs visuels, les *kD tree* deviennent moins efficaces. On peut alors utiliser des méthodes approchées comme le *best-bin-first* (LOWE, 2004).

Filtrage des paires aberrantes

Malgré le soin apporté au choix de la méthode d'association, il est attendu que les paires ne correspondent pas toujours parfaitement. C'est le cas typiquement lorsque les deux nuages de points ne se recouvrent pas totalement, par exemple durant l'exploration d'un nouvel endroit de l'environnement ou lorsqu'il y a des objets dans la scène qui se sont déplacés. Les paires de points mal associés peuvent tout de même ralentir la convergence voire parfois entraîner l'ICP vers un optimum local.

Rejet Une première méthode est simplement d'identifier les aberrations et de les supprimer de l'association. C'est utilisé dans le cas d'une association basée sur les descripteurs, comme les descripteurs *scale-invariant feature transform* (SIFT) où l'on rejette les paires pour lesquelles la distance du deuxième voisin le plus proche est inférieure à un facteur donné du premier voisin (LOWE, 2004).

Lorsque l'association est basée sur des caractéristiques géométriques, les techniques de rejet se basent en général sur le dépassement d'un seuil, avec de nombreuses variations pour le fixer. On utilise parfois une distance maximale entre les points (SEGAL, HAEHNEL et THRUN, 2009) mais cela peut se faire aussi pendant l'association en bornant la distance de recherche du point le plus proche. De la même manière, on peut limiter l'angle entre les surfaces considérées en dessous d'un seuil donné (ZHANG, 1994 ; PULLI, 1999).

De nombreuses propositions préfèrent ne pas fixer un seuil absolu, mais plutôt relatif, basé sur la moyenne et l'écart-type des distances (ZHANG, 1994 ; DRUON, ALDON et CROSNIER, 2006), sur l'écart-type seulement (MASUDA, SAKAUE et YOKOYA, 1996), les quartiles (CHETVERIKOV et al., 2002 ; ARMESTO, MINGUEZ et MONTESANO, 2010) ou même un facteur de la médiane (DIEBEL et al., 2004). Il a été aussi proposé de réduire le seuil d'acceptation à chaque itération en fonction de la convergence, soit manuellement (PULLI, 1999), soit de manière automatique (POMERLEAU, COLAS, FERLAND et al., 2010).

Pondération À côté de ces méthodes qui gardent ou rejettent des paires de points en fonction d'un seuil (fixe ou adaptatif), on en trouve qui affectent un poids pour moduler l'influence des différentes associations. Ces fonctions de poids peuvent être *ad hoc* telles qu'un rapport entre la distance moyenne et la distance des points de la paire (PAN, DAI et PENG, 2010) où l'on voit que plus les points sont proches plus le poids est grand. Mais ces fonctions sont souvent basées sur un noyau tel que la densité d'une loi gaussienne (GODIN, RIOUX et BARIBEAU, 1994) ou de Cauchy (BOSSE et ZLOT, 2009a).

On peut aussi combiner la pondération et le rejet avec des fonctions de poids qui donnent une valeur nulle. On trouve ainsi par exemple la fonction $d \mapsto 1 - \frac{d}{D}$ si $d < D$ et 0 sinon où d est la distance entre les points d'une paire et D une distance maximale (DIEBEL et al., 2004) ou la fonction bicarrée de Tukey $x \mapsto (1 - x^2)^2$ sur $[-1, 1]$ et 0 sinon (MASUDA, 2001 ; STEWART, TSAI et ROYSAM, 2003).

Minimisation de l'erreur

L'étape de minimisation de l'erreur est celle qui calcule la meilleure transformation en fonction de l'association des points (avec ou sans poids).

Définition de l'erreur Comme indiqué plus haut, la forme générale de l'erreur est :

$$\text{error}(\mathcal{P}, \mathcal{Q}) = \sum_{(p,q) \in \mathcal{M}} w(p, q) d(p, q),$$

où d est une mesure de distance. Cette mesure de distance n'est pas nécessairement la même que celle qui a servi à trouver le plus proche voisin. Par exemple, lorsque l'on utilise des descripteurs visuels pour l'association, comme ils sont indépendants de la transformation, ils ne peuvent pas servir à la trouver, La mesure de distance utilisée dans l'erreur à minimiser doit donc se baser sur les caractéristiques géométriques.

L'exemple le plus simple est la distance euclidienne entre les positions des points (dite point-à-point) introduite par BESL et MCKAY (1992) et reprise ensuite dans de nombreuses solutions (GODIN, RIOUX et BARIBEAU, 1994 ; PULLI, 1999 ; DRUON, ALDON et CROSNIER, 2006 ; PAN, DAI et PENG, 2010).

En 2D, l'erreur point-à-ligne est ainsi beaucoup utilisée, notamment dans les environnements structurés (BOSSE et ZLOT, 2009b). L'erreur la plus utilisée en 3D est l'erreur point-à-plan introduite par CHEN et MEDIONI (1991). Mais certaines solutions proposent d'utiliser l'erreur plan-à-plan (SEGAL, HAEHNEL et THRUN, 2009) voire quadrique-à-quadrique (FELDMAR et AYACHE, 1996).

Minimisation Une fois l'erreur définie, le problème est la stratégie permettant d'obtenir la meilleure transformation (RUSINKIEWICZ et LEVOY, 2001). Dans le cas de l'erreur point-à-point, il existe des formes closes en utilisant la décomposition en valeurs singulières (ARUN, HUANG et BLOSTEIN, 1987), les quaternions (HORN, 1987), des matrices orthogonales (HORN, HILDEN et NEGAHDARIPOUR, 1988) ou des biquaternions (WALKER, SHAO et VOLZ, 1991). Dans le cas de l'erreur point-à-plan, on effectue souvent une approximation de petit angle (CHEN et MEDIONI, 1991). Il existe aussi des méthodes itératives ou des méthodes basées sur l'histogramme de corrélation (BOSSE et ZLOT, 2008) ou la transformée de Hough (LOWE, 2004).

Une fois trouvée la transformation minimisant l'erreur, on peut alors l'appliquer au nuage de points d'entrée, refaire une étape d'association, filtrer les paires de points aberrantes et calculer une nouvelle transformation. On continue typiquement à itérer jusqu'à ce que la nouvelle transformation soit assez petite et la transformation finale entre le nuage d'entrée d'origine et le nuage de points de référence est donc la composition itérative de toutes les transformations intermédiaires.

3.2.3 Méthodologie

Notre contribution à la méthodologie de comparaison d'ICP (POMERLEAU, COLAS, SIEGWART et MAGNENAT, 2013) consiste en trois éléments : une procédure de comparaison avec des métriques définies, des jeux de données couvrant une variété d'environnements et de caractéristiques et une bibliothèque modulaire libre en C++ permettant de programmer facilement de nouvelles variantes de l'algorithme.

Procédure

Notre procédure s'inspire de travaux d'évaluation que l'on trouve notamment en vision par ordinateur (SCHARSTEIN et SZELISKI, 2002 ; GEIGER, LENZ et URTASUN, 2012) où se posent des problèmes similaires. L'ICP étant un algorithme d'optimisation local, ses résultats sont très sensibles à la transformation initiale. Un des critères importants à l'évaluation d'une variante d'ICP est donc son domaine de convergence vers la bonne solution. Nous proposons donc, pour l'estimer, d'échantillonner l'espace des transformations initiales autour de la transformation réelle, en utilisant une distribution gaussienne multivariée en six dimensions.

Un deuxième facteur important dans l'alignement est la proportion de points sans correspondance, soit parce que l'objet a bougé, soit pour des raisons d'occultation et de

champ de vision. Pour contrôler cet effet, nous quantifions le recouvrement entre deux nuages de points par la proportion des points d'un nuage d'avoir un correspondant dans le second.

Enfin, le contenu des nuages de points a aussi une grande influence. Par exemple, un environnement intérieur présente souvent de nombreuses surfaces planes qui rendent moins sensibles une mauvaise association. À l'opposé, un environnement naturel avec des arbres, des buissons et des herbes sera plus difficile à aligner si l'association est mauvaise. Un environnement pauvre en contraintes, tel qu'un couloir ou un terrain plat, rendra aussi plus sensible l'ICP au bruit des capteurs, aux variations de densité ou à une mauvaise association. Pour cela, nous proposons d'effectuer la comparaison dans une variété d'environnements.

L'idée est d'obtenir la distribution sur l'erreur sur le résultat en échantillonnant à la fois les paires de nuages de points et les transformations initiales. Cette distribution porte sur l'espace des transformations rigides (de dimension 6 en 3D) et est donc complexe. Pour une meilleure appréhension, nous la projetons sur les erreurs en translation et en rotation. L'erreur sur la transformation ΔT est de la forme :

$$\Delta T = T_r T_g^{-1} = \begin{bmatrix} \Delta R & \Delta t \\ \mathbf{0} & 1 \end{bmatrix},$$

avec T_r le résultat de l'algorithme et T_g la vérité terrain. ΔR et Δt sont respectivement une matrice de rotation et un vecteur de translation. Nous définissons, sans originalité, l'erreur de translation comme la norme de ce vecteur : $e_t = \|\Delta t\|$. Pour les rotations, nous utilisons la distance géodésique sur la sphère unité (HUYNH, 2009) : $e_r = \arccos \frac{\text{trace}(\Delta R) - 1}{2}$.

Plutôt que de comparer directement les distributions, nous nous intéressons à des indicateurs robustes, tels que la médiane et les quantiles. Nous reprenons le choix des quantiles à 50 %, 75 % et 95 % de BAKER et al. (2010). Le choix, pourtant courant, de la moyenne et de l'écart-type de l'erreur est très discutable puisque les distributions ne sont pas gaussiennes et que la moyenne et l'écart-type empiriques ne fournissent pas une bonne représentation de la distribution, en plus de ne pas être robustes. À l'opposé, le choix de la médiane et des quantiles permet d'avoir une meilleure idée des valeurs typiques des erreurs. En particulier, si ces trois valeurs sont proches il est facile de voir que la technique est fiable et répétable et si ces valeurs sont faibles, la technique est précise.

Jeux de données

Nous avons mis à disposition de la communauté des jeux de données de manière à la fois à expérimenter mais aussi à évaluer les algorithmes d'alignement de nuages de points (POMERLEAU, LIU et al., 2012). Un aspect important de ces jeux de données est la présence de la vérité terrain, mesurée à l'aide d'un théodolite avec une précision de l'ordre du millimètre. Les nuages de points ont été recueillis à l'aide d'un scanner laser Hokuyo UTM-30LX monté sur un axe de rotation horizontal transversal permettant d'obtenir un nuage de point 3D en environ 20 s.

Les environnements ont été choisis de manière à couvrir une large gamme de caractéristiques. En premier lieu, il y a des environnements d'intérieur et d'extérieur. Ensuite les environnements peuvent être structurés (des primitives géométriques), non-structurés

TAB. 3.1: Jeux de données pour l'évaluation de l'alignement de nuages de points.

Nom	Nb. scans	Nb. points par scan	Volume des poses ($x \times y \times z$ en m)	Volume de la scène ($x \times y \times z$ en m)
<i>ETH Hauptgebäude</i>	36	191 000	$24 \times 2 \times 0,50$	$62 \times 65 \times 18$
<i>Apartment</i>	45	365 000	$5 \times 5 \times 0,06$	$17 \times 10 \times 3$
<i>Stairs</i>	31	191 000	$10 \times 3 \times 2,50$	$21 \times 111 \times 27$
<i>Gazebo Summer</i>	32	170 000	$5 \times 4 \times 0,07$	$35 \times 45 \times 16$
<i>Gazebo Winter</i>	32	153 000	$4 \times 5 \times 0,09$	$72 \times 70 \times 19$
<i>Mountain Plain</i>	31	102 000	$18 \times 6 \times 2,70$	$36 \times 40 \times 8$
<i>Wood Summer</i>	37	182 000	$10 \times 15 \times 0,50$	$30 \times 53 \times 20$
<i>Wood Autumn</i>	32	178 000	$6 \times 12 \times 0,50$	$36 \times 60 \times 22$

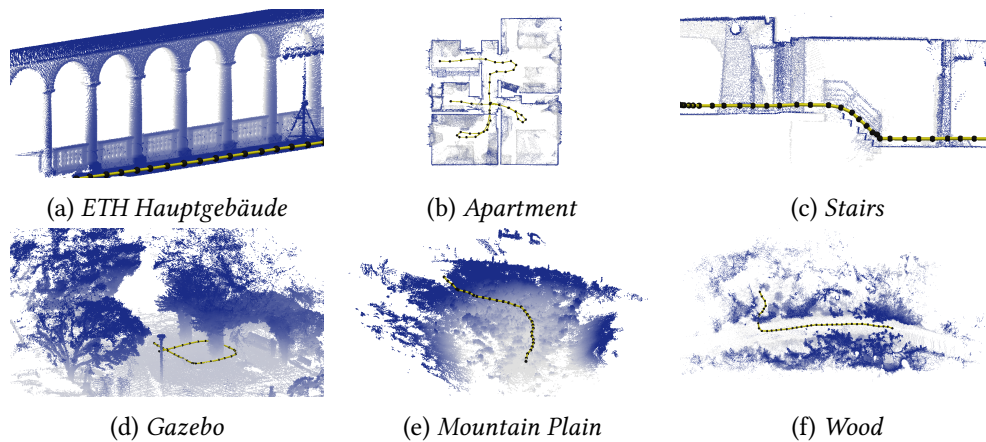


FIG. 3.2: Nuages de points de chaque environnement. En haut : environnements structurés ; en bas : environnements non- ou semi-structurés. Points noirs : poses du capteur dans l'environnement. (Adapté de (POMERLEAU, LIU et al., 2012).)

(objets complexes comme des buissons ou des arbres) ou semi-structurés (comprenant les deux types d'objets). Enfin, les scènes observées peuvent être statiques ou contenir des éléments dynamiques à l'intérieur d'un nuage de point (mouvement capturé durant un scan), entre des nuages de points d'un même jeu de données (comme une porte qui s'ouvre) ou des changements globaux entre des jeux de données (par exemple des changements saisonniers).

La [table 3.1](#) résume les caractéristiques principales des jeux de données proposés. La [figure 3.2](#) montre les nuages de points de chacun des environnements.

libpointmatcher

Notre troisième contribution pour améliorer la comparaison entre les différentes variantes est une bibliothèque d'ICP en C++ : `libpointmatcher`.² Sa conception est modulaire et suit la structure présentée dans la [section 3.2.1](#) et la [figure 3.1](#). Chacun des modules

2. <https://github.com/ethz-asl/libpointmatcher>

TAB. 3.2: Liste des variantes disponibles pour chaque type de module.

Module	Variantes
Filtres d'entrée	BoundingBox, CutAtDescriptorThreshold, FixStepSampling, Identity, MaxDensity, MaxDist, MaxPointCount, MaxQuantileOnAxis, MinDist, ObservationDirection, OrientNormals, RandomSampling, RemoveNaN, SamplingSurfaceNormal, Shadow, SimpleSensorNoise, SurfaceNormal, VoxelGrid
Association	KDTree, KDTreeVarDist, Null
Filtres des paires aberrantes	GenericDescriptor, MaxDist, MedianDist, MinDist, Null, SurfaceNormal, TrimmedDist, VarTrimmedDist
Minimisation de l'erreur	Identity, PointToPlaneWithCov, PointToPoint, PointToPointWithCov
Critères d'arrêt	Bound, Counter, Differential

détaillés dans la [section 3.2.2](#) correspond à une classe spécialisant une interface définie qui permet de composer une solution particulière d'ICP en choisissant les modules et leurs paramètres. En appui de cette bibliothèque, nous fournissons des exécutables capables de lire une configuration en YAML (BEN-KIKI, EVANS et NET, 2009) et de traiter des nuages de points avec la solution d'ICP ainsi définie.

Cette bibliothèque est libre et accueille les contributions sous la forme de nouvelles implémentations de modules d'ICP. La [table 3.2](#) liste les différents modules actuellement disponibles dans `libpointmatcher`.

3.3 Expériences

3.3.1 Comparaison sur les jeux de données

En premier lieu, nous sommes revenus sur deux variantes classiques qui semblent largement utilisées et qui utilisent respectivement la distance point-à-point (BESL et MCKAY, 1992) et point-à-plan (CHEN et MEDIONI, 1991), toutes deux combinées avec le rejet des paires aberrantes basé sur les quartiles (CHETVERIKOV et al., 2002). Le choix de ces deux variantes semble correspondre aux solutions de base les plus utilisées dans la littérature pour la comparaison des nouvelles contributions. Cela permet aussi de proposer à la communauté une implémentation de ces méthodes pour faciliter la comparaison et le développement de nouveaux modules. La [table 3.3](#) résume, pour chacune des deux méthodes, les modules et leur paramètres utilisés.

Nous avons testé ces deux solutions sur les jeux de données présentés précédemment en faisant varier la position relative initiale des nuages de points. Ces perturbations initiales ont été échantillonnées suivant trois gaussiennes multivariées de dimension 6 pour simuler trois niveaux de difficulté distincts : facile, moyen et difficile. Les répartitions d'erreur sont montrées dans la [figure 3.3](#).

La [table 3.4](#) compare les résultats entre les solutions point-à-point et point-à-plan pour chacun des jeux de données et chaque niveau de difficulté. On remarque que l'erreur est souvent plus faible pour les bas quantiles (A50 et A75) pour la solution point-à-plan que pour point-à-point mais que c'est souvent l'inverse pour A95. Utiliser la métrique point-à-plan semble donc améliorer la précision typique mais amener à des erreurs plus grandes

TAB. 3.3: Configuration des variantes d'ICP choisies. DF_{read} : filtres du nuage d'entrée ; DF_{ref} : filtres du nuage de référence ; MF : fonction d'association ; OF : filtres des paires aberrantes ; EM : minimisation d'erreur ; TC : vérification de la transformation.

	Étape	Module	Description
Point-à-point	DF_{read}	MinDist	Garde les points au-delà de 1 m.
		RandomSampling	Garde 5 % des points aléatoirement.
	DF_{ref}	MinDist	Garde les points au-delà de 1 m.
		RandomSampling	Garde 5 % des points aléatoirement.
	MF	KDTree	Utilise un <i>kD tree</i> avec un facteur d'approximation $\epsilon = 3,16$.
	OF	TrimmedDist	Garde les 75 % de paires dont les points sont les plus proches.
EM	PointToPoint	Erreur point-à-point.	
	TC	Counter Differential	Arrête lorsque le nombre d'itérations atteint 150. Arrête lorsque l'incrément de transformation est en dessous de 0,01 m et 0,001 rad.
	Étape	Module	Description
Point-à-plan	DF_{read}	MinDist	Garde les points au-delà de 1 m.
		RandomSampling	Garde 5 % des points aléatoirement.
	DF_{ref}	MinDist	Garde les points au-delà de 1 m.
		SamplingSurfaceNormal	Garde 10 % des points aléatoirement et calcule les vecteurs normaux.
	MF	KDTree	Utilise un <i>kD tree</i> avec un facteur d'approximation $\epsilon = 3,16$.
	OF	TrimmedDist	Garde les 70 % de paires dont les points sont les plus proches.
EM	PointToPlane	Erreur point-à-plan.	
	TC	Counter Differential	Arrête lorsque le nombre d'itérations atteint 150. Arrête lorsque l'incrément de transformation est en dessous de 0,01 m et 0,001 rad.

en cas d'échec.

C'est confirmé si on accumule les distributions pour tous les niveaux de perturbation (figure 3.4). On remarque ainsi que la métrique point-à-plan est généralement meilleure, comme attendu, mais que son avantage disparaît pour les environnements non-structurés (*wood* et *plain*).

3.3.2 Application réelle

En plus de ces comparaisons, nous avons déployé notre logiciel d'ICP sur plusieurs robots et pour plusieurs applications : l'inspection industrielle avec un petit robot à roues magnétiques, la cartographie et l'inspection des berges avec un catamaran autonome et la recherche et le sauvetage (*search and rescue*) avec un robot à chenilles. C'est cette dernière application que je vais présenter ici (pour les autres, voir (POMERLEAU, COLAS et SIEGWART, 2015)).

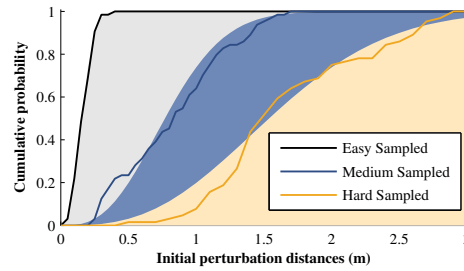


FIG. 3.3: Répartition de l'erreur en translation pour les échantillons de perturbation initiale. Les zones colorées correspondent à la répartition théorique et la différence est due au faible nombre d'échantillons (64) relativement à la dimension du support de la distribution (6). (Repris de (POMERLEAU, COLAS, SIEGWART et MAGNENAT, 2013).)

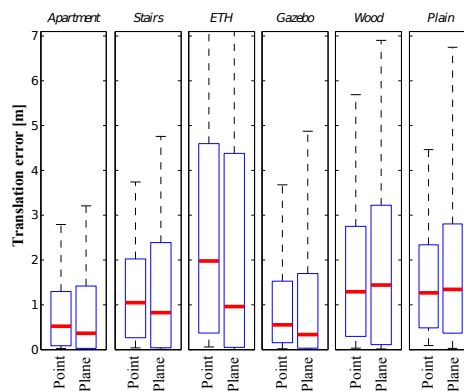


FIG. 3.4: Comparaison des distributions des solutions point-à-point et point-à-plan pour chaque environnement. Trait rouge : A50 ; haut de la boîte : A75 ; turet : A95. (Repris de (POMERLEAU, COLAS, SIEGWART et MAGNENAT, 2013).)

Présentation du robot

Ce travail a été effectué dans le cadre du projet européen NIFTi³ pour assister des pompiers ou les premiers répondants pour les situations de recherche et sauvegarde, telles qu'un tremblement de terre, un accident nucléaire, un danger chimique ou même un incendie. Dans toutes ces conditions, il est nécessaire de faire une reconnaissance pour connaître la dangerosité et l'accessibilité des lieux et planifier au mieux l'intervention. Ces situations sont en plus souvent techniquement difficiles pour un robot avec d'une part un terrain complexe avec des débris et des éléments dynamiques et d'autre part une portée de communication limitée et l'absence de ligne de vue directe avec le robot. Ces éléments font qu'il faut à la fois permettre au robot d'évoluer en 3D et qu'il faut le doter de plus d'autonomie. En particulier il devient important de pouvoir faire la localisation, la cartographie et la navigation de manière embarquée.

Le robot NiftiBot, présenté dans la [figure 3.5](#), est équipé de deux chenilles reliées

3. *Natural Human-Robot Cooperation in Dynamic Environments*, FP7-ICT-247870, 2010–2013 ; <http://www.nifti.eu>.

TAB. 3.4: Résumé des distributions d'erreur obtenues pour les solutions point-à-point et point-à-plan pour chaque jeu de donnée et chaque niveau de difficulté (facile (F), moyen (M) et difficile (D)). En haut : l'erreur en translation (en m) ; en bas : l'erreur en rotation (en rad). Une case plus foncée correspond à une plus grande erreur. (Adapté de (POMERLEAU, COLAS, SIEGWART et MAGNENAT, 2013).)

		Apartment			Stairs			ETH			Gazebo			Wood			Plain			
		A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95	
Translation	F	Plan	0,06	0,47	2,11	0,09	1,17	3,49	0,10	0,44	6,06	0,11	0,38	2,08	0,25	1,55	4,75	0,42	1,54	4,15
		Point	0,13	0,54	1,54	0,35	1,29	2,57	0,47	2,23	6,86	0,28	0,60	1,71	0,39	1,48	4,21	0,51	1,46	3,09
	M	Plan	0,20	1,04	2,98	0,61	2,08	4,64	0,60	4,06	16,3	0,28	0,96	3,51	1,25	2,92	6,62	1,30	2,58	5,58
		Point	0,46	1,03	2,32	0,94	1,86	3,38	1,92	4,29	11,2	0,49	1,13	3,18	1,19	2,52	5,15	1,21	2,17	3,76
	D	Plan	1,35	2,18	3,66	2,05	3,28	5,50	4,18	8,55	19,6	1,87	3,33	6,95	2,79	4,52	7,86	2,35	4,13	8,85
		Point	1,29	1,99	3,24	1,81	2,78	4,75	3,84	7,06	14,8	1,58	2,79	4,57	2,32	3,73	6,82	2,02	3,14	6,33
Rotation	F	Plan	0,02	0,20	1,14	0,02	0,31	1,58	0,01	0,02	0,61	0,02	0,08	0,48	0,05	0,34	0,95	0,07	0,20	0,60
		Point	0,07	0,25	0,97	0,12	0,39	1,22	0,05	0,22	0,83	0,04	0,17	0,41	0,09	0,29	0,77	0,09	0,20	0,44
	M	Plan	0,08	0,47	1,80	0,16	1,08	2,09	0,01	0,25	2,91	0,04	0,35	0,97	0,31	0,78	1,53	0,19	0,38	0,99
		Point	0,20	0,61	1,49	0,33	0,78	1,63	0,14	0,59	1,82	0,15	0,35	0,80	0,32	0,69	1,22	0,20	0,37	0,77
	D	Plan	1,01	1,72	2,95	1,48	1,91	2,94	1,31	2,09	3,11	0,58	1,31	2,88	1,05	1,56	2,53	0,50	1,09	3,05
		Point	1,04	1,60	2,53	1,10	1,64	2,53	0,97	1,73	3,05	0,58	1,20	2,59	0,97	1,44	2,35	0,46	0,99	2,09

par un système de différentiel qui leur permet de s'adapter passivement au terrain. De plus, chaque chenille est munie de deux *flippers* qui permettent une adaptation active à l'environnement comme l'extension pour avoir une plus grande surface de traction ou au contraire réduire l'encombrement pendant les virages. Ces *flippers* permettent aussi d'augmenter sensiblement la capacité de franchissement de ce robot. Il peut en effet grimper des obstacles jusqu'à 30 cm de haut, franchir des fossés de 50 cm de large, monter des pentes de 45° et négocier des escaliers.

En termes de capteurs, il est équipé d'un laser 2D Sick LMS-151 monté de manière à tourner autour de l'axe de déplacement du robot. L'agrégation des scans 2D permet de construire un nuage de points 3D couvrant tout l'hémisphère avant du robot à une fréquence d'environ 0,35 Hz (pour environ 55 000 points). En plus du laser, le robot est équipé d'une caméra omnidirectionnelle PointGrey Ladybug2 et une centrale inertielle avec GPS Xsens MTi-G.

Solution d'ICP

Nous avons spécialisé l'algorithme générique d'ICP formalisé plus haut pour les besoins spécifiques de ce robot. Le but est d'aligner successivement les nuages de points 3D assemblés à partir des scans du laser de manière à obtenir une carte 3D de l'environnement exploré sous la forme d'un nuage de points.

En premier lieu, la transformation initiale de l'ICP est fournie par un filtre de Kalman faisant la fusion d'informations entre la centrale inertielle et les informations d'odométrie. Il faut noter que l'odométrie d'un robot à chenille est significativement plus mauvaise que celle d'un robot différentiel puisque dès qu'il y a une composante de rotation, l'hypothèse de frottement sans glissement n'est plus valide et le centre de rotation instantané est alors difficile à prédire. De plus, l'hypothèse de planarité du sol n'est pas toujours



FIG. 3.5: Photographie du robot NiftiBot. Le capteur laser est monté sur l'axe pointant vers l'avant du robot et tourne alternativement à droite et à gauche de 90° pour obtenir un balayage complet de l'avant du champ de vision. (Crédit Photo : François Pomerleau.)

satisfaite et les lamelles des chenilles, nécessaires pour augmenter l'adhérence du robot, induisent des vibrations qui dégradent les performances de la centrale inertielle. Ainsi, l'ICP est réellement nécessaire à la localisation du robot, malgré l'utilisation en amont de l'odométrie et d'une centrale inertielle.

L'erreur choisie est point-à-plan, et les nuages de points doivent donc contenir le vecteur normal. Nous utilisons donc des filtres d'entrée pour rajouter cette information. La géométrie du capteur induit aussi des différences de densité surfacique des points avec notamment de très nombreux points pour les surfaces proches et celles directement face au robot. On réduit ainsi le nombre de points en limitant la densité pour éliminer les points redondants.

L'association se fait à l'aide de la distance euclidienne entre les positions des points, mais on filtre les paires en fonction de l'angle entre les normales de manière à éviter notamment qu'une face d'un mur soit associée à l'autre face, ce qui décalerait le nuage de point de l'épaisseur du mur.

Enfin, on arrête les itérations si la nouvelle transformation est trop faible, s'il y a eu trop d'itération ou si la transformation finale est trop grande, ce qui indique que l'ICP s'est trouvé dans un minimum local. La configuration complète de cette solution est résumée dans la [table 3.5](#).

Résultats

Nous avons appliqué cette configuration d'ICP dans plusieurs environnements qui présentent des caractéristiques différentes. Ces environnements sont relativement représentatifs d'un certains nombres de difficultés que serait amené à rencontrer un robot d'intervention pour la recherche et le sauvetage : un bâtiment de plusieurs étages, un environnement industriel, ou une église partiellement effondrée à la suite d'un tremblement de terre.

TAB. 3.5: Configuration de la solution d'ICP pour NiftiBot. Les noms des modules sont ceux de la bibliothèque `libpointmatcher`. DF_{read} : filtres du nuage d'entrée ; DF_{ref} : filtres du nuage de référence ; MF : fonction d'association ; OF : filtres des paires aberrantes ; EM : minimisation d'erreur ; TC : vérification de la transformation.

Étape	Module	Description
DF_{read}	<code>SimpleSensorNoise</code>	Ajoute l'information d'incertitude pour le capteur.
	<code>SamplingSurfaceNormal</code>	Garde 80 % des points et extrait les vecteurs normaux à l'aide des 20 plus proches voisins.
	<code>ObservationDirection</code>	Ajoute le vecteur pointant vers l'origine du capteur.
	<code>OrientNormals</code>	Orienté les normales à l'aide de la direction d'observation.
	<code>MaxDensity</code>	Supprime des points pour ne pas dépasser une densité de 100 points/m ³ .
DF_{ref}	<code>SurfaceNormal</code>	Calcule les vecteurs normaux à l'aide des 20 plus proches voisins.
	<code>MaxDensity</code>	Supprime des points pour ne pas dépasser une densité de 100 points/m ³ .
	<code>MaxPointCount</code>	Supprime 30 % des points s'il y en a plus de 600 000.
MF	<code>KDTree</code>	Utilise un <i>kD tree</i> avec une distance maximale de 0,5 m et un facteur d'approximation $\epsilon = 3,16$.
OF	<code>TrimmedDist</code>	Garde les 80 % de paires dont les points sont les plus proches.
	<code>SurfaceNormal</code>	Supprime les paires de points dont l'angle entre les normales est plus grand que 50°.
EM	<code>PointToPlane</code>	Erreur point-à-plan.
TC	<code>Differential</code>	Arrête lorsque l'incrément de transformation est en dessous de 0,01 m et 0,001 rad.
	<code>Counter</code>	Arrête lorsque le nombre d'itérations atteint 40.
	<code>Bound</code>	Arrête si la transformation totale est plus grande que 5,0 m et 0,8 rad.

Intérieur Le premier exemple d'application est un environnement structuré intérieur démontrant la nécessité d'un algorithme en 3D: la cage d'escalier du bâtiment CLA de l'ETH Zürich. Le robot était commandé à l'aide d'une manette de jeu par un opérateur qui le suivait. Le robot a débuté dans un bureau à l'étage E, est descendu jusqu'à l'étage C avant de remonter jusqu'à l'étage J, soit 7 étages.

Pour cette application les nuages de points n'ont pas été obtenus en continu mais à l'arrêt à la demande de l'opérateur, environ un tous les deux mètres. L'alignement des points et donc la localisation et la carte ont été faits sur l'ordinateur embarqué du robot.

La [figure 3.6](#) présente la carte finale.

Chantier ferroviaire Il a été possible en deux occasions de tester le robot dans un chantier ferroviaire de Prague avec la permission du *Depo kolejových vozidel Praha* (dépôt de véhicules ferroviaires de Prague). Dans les deux expériences, le robot était aussi commandé par un opérateur qui déclenchait l'acquisition des nuages de points manuellement. L'opérateur n'était pas un expert en cartographie et les nuages de points ont été pris de manière plus irrégulière avec parfois jusqu'à 8 m d'écart.

Dans la première expérience, illustrée dans la [figure 3.7a](#), le robot a fait le tour d'un

3 Alignement de nuages de points en robotique

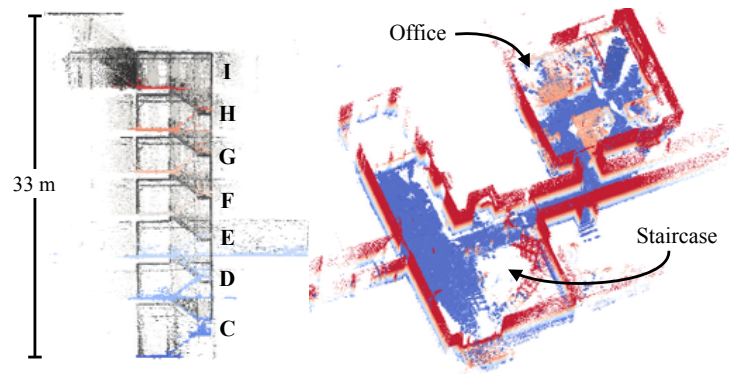


FIG. 3.6: Cartographie d'une cage d'escalier de 7 étages avec NiftiBot. À gauche : vue de côté de la carte avec les points du sol colorés en fonction de l'élévation. À droite : vue de haut de l'étage E avec les points colorés en fonction de l'élévation : de bleu en bas à rouge en haut. (Repris de (POMERLEAU, COLAS, SIEGWART et MAGNENAT, 2013).)

wagon. Malgré l'absence de clôture de boucle dans l'algorithme d'ICP, la précision était suffisante pour que les nuages de points de la fin puissent s'aligner avec ceux du début.

Dans la deuxième expérience, illustrée dans la [figure 3.7b](#), il s'agissait d'explorer l'intérieur d'un wagon désaffecté dans lequel une personne se tenait immobile (pour tester en parallèle les capacités d'une caméra thermique). Ensuite le robot est ressorti, a traversé de la végétation et a longé un autre wagon.

Pour des raisons techniques pendant ces deux expériences, les nuages de points ont été seulement enregistrés sur le terrain. L'alignement s'est fait dans un second temps, à une vitesse de quatre fois le temps réel sur un ordinateur similaire à celui embarqué. Il aurait donc été tout à fait possible de générer ces cartes directement sur le robot.

Église San Francesco d'Assisi à Mirandola Une série de tremblements de terre d'une magnitude jusqu'à 5,8 a frappé l'Émilie-Romagne au nord de l'Italie en mai 2012. Trois mois plus tard, des partenaires du projet européen Nifti ont eu l'autorisation de déployer NiftiBot avec l'assistance des *Vigili del Fuoco* (les pompiers italiens) et du *Beni Culturali* (le ministre de la culture en Italie) pour une mission d'évaluation des dommages au patrimoine. L'un des sites visités était l'église *San Francesco d'Assisi* à Mirandola (XIII^e–XV^e siècles) dont le toit de la nef était partiellement effondré.

Le robot a commencé à l'extérieur de l'église, franchi la porte de gauche a parcouru la galerie ouest ([figure 3.8](#)). Les dommages à la nef et la taille des débris étaient trop importants pour explorer plus avant ([figure 3.9](#)).

L'intérieur de l'église était interdit pour des raisons de sécurité et le robot a donc été télé-opéré depuis un poste de contrôle à l'extérieur. Les nuages de points ont été acquis cette fois-ci de manière continue durant le mouvement du robot. Comme pour le déploiement précédent, l'alignement a été effectué hors-ligne mais aurait pu être fait directement en embarqué.

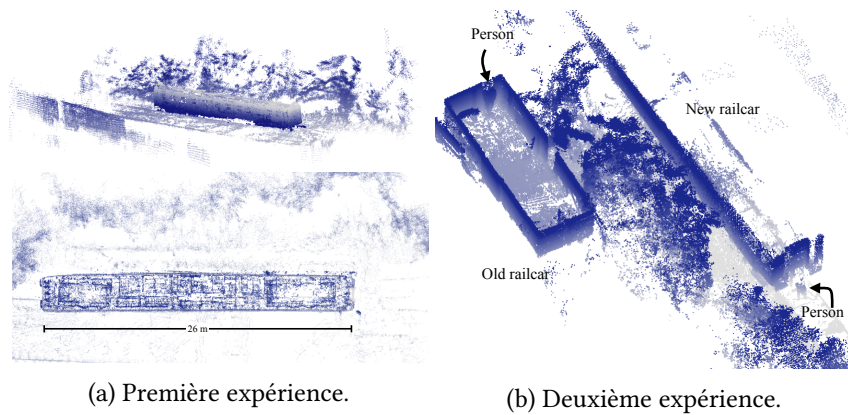


FIG. 3.7: Déploiement de NiftiBot dans un chantier ferroviaire avec un wagon et de la végétation. En haut : vue de côté de l'environnement reconstruit ; en bas : vue de haut de l'environnement ; à droite : vue de la deuxième expérience. (Adapté de (POMERLEAU, COLAS et SIEGWART, 2015).)

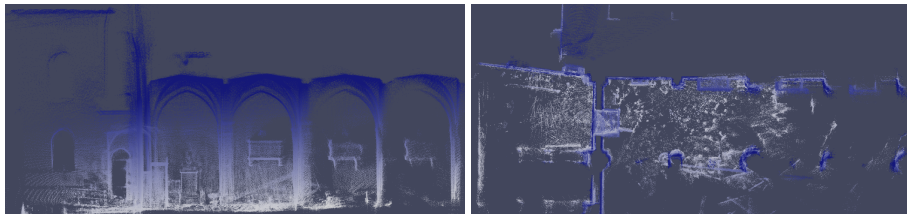


FIG. 3.8: Reconstruction de l'église *San Francesco d'Assisi* à Mirandola. En haut : vue de côté ; en bas : vue de dessus. (Repris de (POMERLEAU, COLAS et SIEGWART, 2015).)

3.4 Discussion et conclusion

L'alignement de nuages de points est un problème que l'on retrouve dans plusieurs communautés de recherche pour diverses applications. En robotique, l'ICP est appliqué surtout pour des questions de reconstruction de l'environnement et d'estimation du déplacement du robot. Ainsi, l'ICP est à la base d'un certain nombre d'algorithmes de SLAM et le nombre de publications portant sur un de ses aspects atteste son importance.

Pour autant, il est difficile de toujours évaluer les contributions tant les justifications ou les expérimentations sont variées et peu mises en relation. Nous avons donc cherché à consolider le champ de l'ICP en revenant à une méthodologie plus scientifique. Nous avons ainsi proposé une synthèse de la structure des algorithmes d'ICP dans laquelle les variantes peuvent être exprimées. Nous avons aussi défini une méthode de comparaison et publié une bibliothèque logicielle pour y aider. Nous avons enfin enregistré des jeux de données avec la vérité terrain pour servir de bancs de tests entre les algorithmes. La prochaine étape, qui reste encore à faire, est de reprendre la littérature à l'aide de ces outils pour faire une évaluation globale d'une grande partie des variantes.

En parallèle de ce travail pour une meilleure fondation et compréhension de cet algorithme, nous avons pu le déployer sur plusieurs plateformes. Cela nous a permis de montrer qu'à l'heure actuelle, avec une implémentation de qualité, il était possible d'ali-



FIG. 3.9: Comparaison entre une photographie et la reconstruction 3D. À gauche : photographie de la galerie ouest pris depuis l'entrée ; à droite : vue de face de la reconstruction. (Repris de (POMERLEAU, COLAS et SIEGWART, 2015).)

gner de gros nuages de points pour obtenir une carte 3D de l'environnement calculée en temps réel à bord du robot. Pour cela il n'a pas été nécessaire de faire appel à des variantes particulièrement sophistiquées de l'algorithme générique. Dans des environnements structurés à semi-structurés, il est avantageux d'extraire les vecteurs normaux locaux pour utiliser la métrique d'erreur point-à-plan. Pour une application robotique où il peut exister des objets mobiles, on peut aussi utiliser un filtrage des paires aberrantes basé sur la distance entre les points pour rejeter celles dont les points sont trop éloignés. Enfin, on remarque que l'information entre les points est très redondante dans les gros nuages de points et qu'alors il est faisable de les décimer pour diminuer grandement le coup de calcul sans un impact majeur sur la qualité de l'alignement.

En conclusion, nous avons montré qu'une approche scientifique de l'algorithme d'ICP en permet une meilleure appréhension, qui rend alors possible de sélectionner des variantes de manière à rendre possible l'alignement de nuages de points 3D en ligne pour localiser un robot et reconstruire son environnement.

4 Estimation d'état en robotique

4.1 Introduction du problème

L'estimation d'état peut être vue comme le pendant en robotique de la proprioception en biologie. Plus précisément, pour un système donné, on cherche à estimer son état en prenant en compte des observations et son évolution. Pour un système robotique, on s'intéresse ainsi typiquement à la pose du robot, c'est-à-dire sa position dans l'espace, son orientation et les positions et orientations relatives de chacune des parties articulées par rapport aux autres. L'estimation d'état est parfois simplifiée lorsqu'un capteur permet de mesurer directement et avec les caractéristiques voulues (précision, fréquence) l'état ou certaines de ses dimensions. C'est le cas par exemple de bras articulés en robotique industrielle dotés d'encodeurs absolus suffisamment précis pour être considérés comme la vérité de la position du bras.

Je me suis intéressé à la robotique mobile en extérieur et en particulier à deux applications sur des plateformes différentes. La première consiste en l'inspection et le suivi de berges à l'aide d'un bateau autonome ou *unmanned* ou *autonomous surface vessel* (ASV) (voir la [figure 4.1](#)). La seconde se place dans le cadre de la recherche et du sauvetage avec NiftiBot.

Dans les deux cas, le processus repose sur le filtrage bayésien qui s'intéresse à calculer la distribution de probabilité sur une variable d'état à un instant donné à partir de l'ensemble des observations effectuées jusque-là et éventuellement des actions entreprises. Grâce à l'hypothèse de Markov, indiquant une indépendance conditionnelle de l'état suivant vis-à-vis des états précédents connaissant l'état courant, il est possible d'obtenir une formulation itérative de l'inférence. De plus, ce formalisme peut s'adapter à la prise en compte des capteurs de modalité sensorielles différentes. Il est donc facile de combiner la fusion multi-capteurs avec le filtrage temporel.

Dans le cas des ASVs, l'estimation d'état a déjà été considérée sur un certain nombre de plateformes, telles que *Charlie* (BIBULI et al., 2009), un monocoque (FEEMSTER et ESPOSITO, 2011) ou de petits hydroglisseurs (DHARIWAL et SUKHATME, 2007). Ces travaux visent au déploiement de ces véhicules dans de grands espaces ouverts pour lesquels ils se restreignent à un espace 2D en négligeant les mouvements de roulis et tangage. FOSSEN (2011) présente un modèle de masse ponctuelle complet qui est utilisé tant dans les approches 2D que la nôtre. D'autres travaux mentionnent l'utilisation de matériel commercial fournissant une estimée de l'attitude du robot (SUBRAMANIAN et al., 2006 ; NEAL et al., 2012 ; ALMEIDA, SILVESTRE et PASCOAL, 2010). Cependant, il n'est pas évident de savoir quels capteurs et algorithmes sont employés dans ces solutions. Dans tous les cas, il n'y a pas beaucoup de travaux traitant de l'estimation de l'attitude malgré les nombreux ASVs développés depuis plus de deux décennies.

Les robots à roues ont l'avantage de pouvoir utiliser l'odométrie pour estimer leur mouvement. On ajoute souvent un gyroscope ou un magnétomètre pour stabiliser la mesure

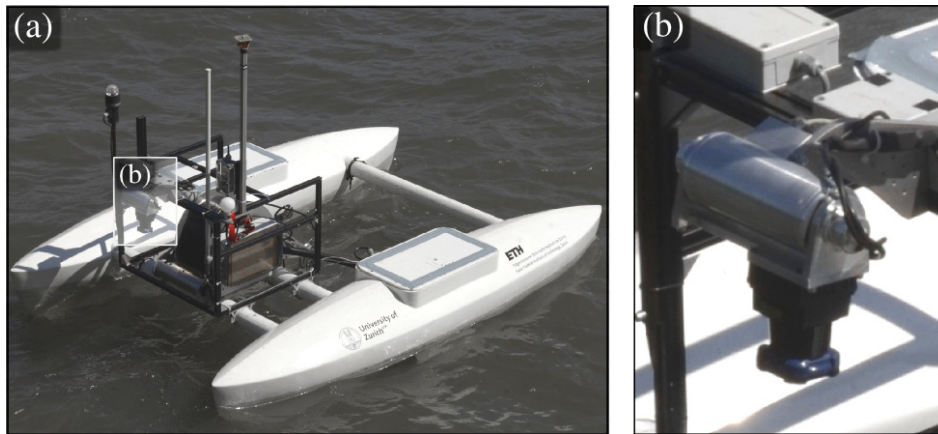


FIG. 4.1: Catamaran d'inspection de berges Lizbeth. (a) vue générale ; (b) zoom sur le scanner. (Repris de (HITZ et al., 2015).)

de l'orientation (le cap). On peut alors même estimer les glissements pour les véhicules à chenilles (ENDO et al., 2007). Là encore, l'hypothèse a longtemps été faite d'une évolution en 2D donc sur un sol parfaitement plan. Le passage en 3D nécessiterait une connaissance du terrain sur lequel évolue le robot. En l'absence d'un tel modèle, on utilise des capteurs extéroceptifs tels que des caméras (KONOLIGE, AGRAWAL et SOLÀ, 2010 ; SCARAMUZZA et FRAUNDORFER, 2011 ; FRAUNDORFER et SCARAMUZZA, 2012) ou des lasers (MORALES et al., 2009 ; BACHRACH et al., 2011 ; KOHLBRECHER et al., 2011).

L'estimation d'état en 3D a été par contre plus étudiée pour les *unmanned aerial vehicles* (UAVs). Les solutions reposent souvent sur un filtre de Kalman étendu centré sur une centrale inertielle ou *inertial measurement unit* (IMU) sans avoir à expliciter le modèle dynamique (TRAWNY et ROUMELIOTIS, 2005). C'est une solution parfois couplée à de l'odométrie visuelle ou même au GPS (LEUTENEGGER et SIEGWART, 2012 ; S. WEISS et al., 2013). Nous nous sommes aussi intéressés au *complementary filter* (CF) de MAHONY, HAMEL et PFLIMLIN (2008) qui estime l'attitude d'un UAV à l'aide de gyroscopes et d'accéléromètre en séparant vitesses et angles.

Ce chapitre synthétise deux séries de travaux respectivement sur un bateau et NiftiBot (KUBELKA et al., 2015 ; HITZ et al., 2015) et en présente en parallèle les modèles (section 4.2) et les expériences (section 4.3).

4.2 Modèles

4.2.1 Modèles pour bateau

Le but est d'estimer la position et l'attitude du repère du robot \mathbb{B} par rapport au repère inertiel du monde \mathbb{W} . Comme pour beaucoup d'ASVs, Lizbeth est équipé d'un GPS, d'une IMU et d'une boussole. Le GPS mesure la position du bateau par rapport à \mathbb{W} , la boussole l'orientation du nord magnétique dans le repère \mathbb{B} et l'IMU les accélérations et vitesses de rotation dans le repère \mathbb{B} . On suppose que toutes les mesures des capteurs sont sujettes à un bruit gaussien additif. Nous supposons de plus que les accéléromètres et les gyroscopes ont un biais additif, variable dans le temps et qu'il faut donc estimer avec les autres états

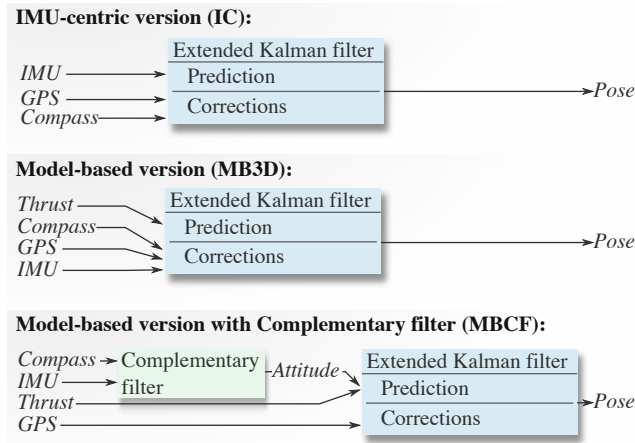


FIG. 4.2: Variantes de filtres pour l'estimation d'état sur un bateau (repris de (HITZ et al., 2015)).

du système.

Dans de grands espaces ouverts, l'erreur du GPS de quelques mètres est négligeable par rapport aux distances en jeu ; mais pour des applications d'inspection des berges, il devient nécessaire de gagner en précision et en continuité.

Nous comparons trois variantes d'*extended Kalman filter* (EKF) avec des hypothèses différentes. La première est un EKF centré sur l'IMU (IC), c'est-à-dire que les informations inertielles servent au modèle de prédiction et que les observations sont le GPS et la boussole. La seconde variante, basée sur un modèle (MB3D), utilise un modèle dynamique du bateau à partir des commandes de poussée ; l'IMU est utilisée comme observation aux côtés du GPS et de la boussole. La troisième variante (MBCF) combine un filtre complémentaire (CF) pour estimer l'attitude à l'aide de l'IMU et de la boussole et un EKF avec modèle dynamique combinant cette attitude et la commande de poussée pour la prédiction et le GPS comme observation. La figure 4.2 présente ces trois variantes avec les informations dont elles se servent. Dans cette première partie, l'accent est mis en particulier sur l'étape de prédiction et c'est là que les spécificités entre les algorithmes se font jour.

EKF centré sur l'IMU (IC)

Ce filtre adapte l'état de l'art de l'estimation d'état utilisée sur les UAVs et notre formulation suit plus précisément le travail de LEUTENEGGER et SIEGWART (2012). Le vecteur d'état est défini de la façon suivante :

$$\mathbf{x} = [\mathbf{p}, \mathbf{q}, \mathbf{v}, \mathbf{b}_g, \mathbf{b}_a]^\top,$$

avec $\mathbf{p} \in \mathbb{R}^3$ la position dans le repère du monde \mathbb{W} , $\mathbf{q} \in SO(3)$ l'attitude représentée par un quaternion unitaire, $\mathbf{v} \in \mathbb{R}^3$ la vitesse linéaire dans \mathbb{W} et $\mathbf{b}_g \in \mathbb{R}^3$ et $\mathbf{b}_a \in \mathbb{R}^3$ les biais respectifs des gyroscopes et accéléromètres. Le quaternion $\mathbf{q} = [q_1, \boldsymbol{\rho}^\top]^\top$ définit une rotation d'angle α autour d'un axe $\mathbf{a} \in \mathbb{R}^3$ tels que $q_1 = \cos(\alpha/2)$ et $\boldsymbol{\rho} = \sin(\alpha/2) \cdot \mathbf{a}$.

La dynamique de l'état est alors décrite par le système d'équations suivant :

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{q}} = \frac{1}{2}\Xi(\mathbf{q})(\mathbb{B}\tilde{\boldsymbol{\omega}} - \mathbf{b}_g) \\ \dot{\mathbf{v}} = \mathbf{R}_{\mathbb{W}\mathbb{B}}(\mathbf{q})(\mathbb{B}\tilde{\mathbf{a}} - \mathbf{b}_a) + \mathbb{W}\mathbf{g} \\ \dot{\mathbf{b}}_g = \mathbf{0} \\ \dot{\mathbf{b}}_a = -\frac{1}{\tau}\mathbf{b}_a \end{cases}$$

avec $\mathbb{B}\tilde{\boldsymbol{\omega}}$ et $\mathbb{B}\tilde{\mathbf{a}}$ sont les valeurs respectives des gyroscopes et des accéléromètres, $\mathbb{W}\mathbf{g}$ le vecteur de la gravité, $\mathbf{R}_{\mathbb{W}\mathbb{B}}$ la matrice de rotation entre \mathbb{B} et \mathbb{W} et :

$$\Xi(\mathbf{q}) = \begin{bmatrix} \boldsymbol{\rho}^\top \\ q_1\mathbb{I}_{3\times 3} + [\boldsymbol{\rho}]_\times \end{bmatrix} \in \mathbb{R}^{4\times 3},$$

où $[\cdot]_\times$ sert à noter l'opérateur de produit vectoriel défini tel que $[\mathbf{a}]_\times\mathbf{b} = \mathbf{a} \wedge \mathbf{b}$ avec $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$:

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

La matrice de covariance sur le quaternion représentant l'attitude n'est pas bien définie à cause de la contrainte unitaire (TRAWNY et ROUMELIOTIS, 2005). Pour contourner ce problème, le filtre est adapté pour travailler sur l'état d'erreur $\Delta\mathbf{x}$ plutôt que sur l'état lui-même. Ainsi le quaternion d'erreur $\delta\mathbf{q}$ peut être remis en 3D avec l'approximation de petit angle (TRAWNY et ROUMELIOTIS, 2005). L'expression complète des matrices jacobiennes peut être trouvée dans LEUTENEGGER et SIEGWART (2012).

EKF basé sur un modèle (MB3D)

Contrairement au précédent, ce modèle met en œuvre un modèle dynamique du système à partir des commandes motrices tel qu'attendu pour un EKF. Le modèle suit les dérivations de FOSSEN (2011). Il se compose d'une partie cinématique (équation 4.1) et d'une partie dynamique (équation 4.2) :

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu} \quad (4.1)$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_m + \boldsymbol{\tau}_e \quad (4.2)$$

avec $\boldsymbol{\eta} = [\mathbf{p}, \mathbf{q}]^\top$ la pose du bateau dans le repère \mathbb{W} , $\mathbf{J}(\boldsymbol{\eta})$ est la transformation entre \mathbb{B} et \mathbb{W} , $\boldsymbol{\nu} \in \mathbb{R}^6$ les vitesses linéaires et angulaires dans le repère \mathbb{B} , \mathbf{M} la matrice d'inertie, \mathbf{C} la matrice de Coriolis, \mathbf{D} la matrice d'amortissement, \mathbf{g} le vecteur de forces hydrostatiques, $\boldsymbol{\tau}_m$ les forces motrices d'entrée et $\boldsymbol{\tau}_e$ des perturbations extérieures inconnues (vent, courant, etc.). Les matrices d'inertie et de Coriolis peuvent être calculées à partir de la connaissance de la distribution de masse du bateau. Par contre, la matrice d'amortissement est difficile à connaître, contient *a priori* 36 éléments si on la considère linéaire ($\mathbf{D}(\boldsymbol{\nu}) = \mathbf{D}_l\boldsymbol{\nu}$), mais possiblement plus en tenant compte d'effets non-linéaires. Le problème est le même avec les forces hydrostatiques $\mathbf{g}(\boldsymbol{\eta})$ que l'on peut considérer comme linéaires ($\mathbf{g}(\boldsymbol{\eta}) = \mathbf{G}\boldsymbol{\eta}$).

Nous simplifions ces matrices pour réduire le nombre de paramètres à 17 (voir (HITZ et al., 2015) pour plus de détails) :

- la masse m du bateau ;
- les moments d'inertie I_ϕ, I_θ, I_ψ ;
- des facteurs d'amortissement $D_x, D_y, D_z, D_\phi, D_\theta, D_\psi, D_{x\psi}$;
- des facteurs hydrostatiques G_z, G_ϕ, G_θ ;
- un facteur d'inefficacité β pour tenir compte de l'asymétrie des pales des moteurs, optimisés pour la propulsion vers l'avant ;
- les valeurs des forces $f_{1,0}$ et $f_{0,5}$ mesurées respectivement pour la poussée maximale et à la moitié de la poussée.

EKF avec filtre complémentaire (MBCF)

La troisième approche se base sur une estimation séparée de l'attitude du bateau et de sa position. Nous utilisons le filtre complémentaire (CF) (MAHONY, HAMEL et PFLIMLIN, 2008), qui intègre les mesures gyroscopiques ω et les corrige avec des directions connues c_i . Dans notre cas, les directions mesurées nous sont fournies par la boussole pour le nord et l'accéléromètre pour la direction de la gravité. L'état du CF consiste en l'attitude représentée par un quaternion unitaire q et un vecteur de biais des gyroscopes b_g . Le filtre est alors défini comme suit :

$$\begin{cases} \dot{\omega}_c = -\text{vex} \left(\sum_{i=1}^N \frac{k_i}{2} (c_i \hat{c}_i^\top - \hat{c}_i c_i^\top) \right) \\ \dot{q} = \frac{1}{2} \hat{q} \otimes (\omega - b_g + k_P \omega_c) \\ \dot{b}_b = -k_I \omega_c \end{cases}$$

avec vex l'opérateur inverse de celui du produit vectoriel : $\text{vex}([a]_\times) = a$ pour $a \in \mathbb{R}^3$. Le terme de correction ω_c dépend des directions mesurées c_i , de leurs estimations \hat{c}_i , et de leurs poids k_i . Les coefficients k_P et k_I sont des paramètres du filtre.

Une fois l'attitude estimée par le CF, le reste de l'état, à savoir la position, est estimée à l'aide d'un EKF qui correspond aux trois premières dimensions du modèle précédent (équations (4.1) et (4.2)).

4.2.2 Modèles pour robot au sol

Pour le robot au sol, nous comparons différentes combinaisons de capteurs. Comme précédemment, le robot au sol est doté d'une IMU, de l'odométrie de ses chenilles, d'une odométrie visuelle et d'une odométrie laser à l'aide de l'ICP (voir la section 3.3.2). Pour le modèle de prédiction, nous suivons les équations dynamiques du modèle d'EKF centré sur l'IMU (LEUTENEGGER et SIEGWART, 2012) formulé comme au-dessus dans l'espace d'état d'erreur. La seule différence est que les vitesses angulaires et les accélérations sont estimées dans l'état plutôt que prises dans les mesures. Le vecteur d'état est donc augmenté de la vitesse angulaire ω et de la force spécifique f (SAVAGE, 1998) :

$$x = [p, q, v, \omega, f, b_g, b_a]^\top.$$

Pour ne pas alourdir les équations, les repères dans lesquels les différentes quantités sont exprimées sont omis. En pratique, il faut prêter attention aux transformations entre ces différents repères (repère du monde, de l'IMU, des odométries) et en faire un étalonnage.

Modèle d'observation de l'IMU

Les gyroscopes de l'IMU donnent une mesure, avec biais, des vitesses angulaires. On peut donc écrire :

$$z_g = \omega + b_g + \eta_g,$$

avec z_g l'observation prévue par le modèle et η_g un bruit gaussien.

Les accéléromètres mesurent la force spécifique, c'est-à-dire l'accélération subie par rapport à la chute libre. Ainsi, un accéléromètre placé sur une table mesure une force vers le haut, celle de la table qui l'empêche de suivre le champ gravitationnel. Le modèle d'observation est donc :

$$z_a = f + b_a + \eta_a.$$

Modèle d'observation de l'odométrie

Pour l'odométrie des chenilles, on fait l'hypothèse que le robot se déplace localement sur un plan. Ainsi il n'y a pas de composante verticale à la vitesse instantanée dans le référentiel \mathbb{B} du robot. On considère aussi, ce qui est moins vrai en pratique, qu'il n'y a pas de glissement latéral.

On écrit alors :

$$z_O = v + \eta_O,$$

en considérant que l'observation mesurée sera $z_{O,x} = \frac{v_r + v_l}{2}$ avec v_r et v_l les vitesses respectives des chenilles droite et gauche.

Modèle d'observation de l'ICP

Comme expliqué plus haut, l'ICP permet de calculer la transformation entre deux nuages de points. Cela permet donc d'évaluer le déplacement relatif entre les deux instants où ont été pris les nuages de points. Contrairement au GPS, il ne s'agit pas d'une mesure de position absolue mais d'une mesure de position relative (différence de positions). L'odométrie classique est dans ce cas aussi en ce que la mesure de vitesse des roues ou des chenilles est obtenue par différence de position entre deux instants. Elle est cependant considérée comme une vitesse par approximation de la dérivée par la différence finie divisée par l'intervalle de temps. Lorsque l'intervalle de temps est court, l'approximation est bonne et l'on n'a pas de problème. En revanche dans notre cas il faut près de 3 s pour relever un nuage de points ; nous proposons donc de comparer trois approches dans la prise en compte d'information d'ICP.

Position incrémentale La première approche est de considérer l'information issue de l'ICP comme une information de position relative :

$$z_{ICP} = p_{t(i)} - p_{t(i-1)} + \eta_{ICP},$$

avec $t(i)$ et $t(i - 1)$ qui correspondent aux instants auxquels les nuages de points ont été pris.

Une telle approche se retrouve dans la littérature (MA et al., 2012) mais viole l'hypothèse de Markov d'ordre un.

Vitesse Une deuxième approche est de transformer l'information d'ICP en information de vitesse, simplement en calculant la vitesse instantanée qui mène à la transformation calculée après l'écart de temps $t(i) - t(i - 1)$. On peut alors écrire :

$$z_{ICP} = [v, \omega]^T.$$

Cette approche, dont on trouve des variantes dans la littérature (ALMEIDA et SANTOS, 2013), ne viole pas les hypothèses du filtre mais suppose que le mouvement est uniforme, ce qui peut poser problème lorsque la fréquence de mesure est assez faible, ce qui est notre cas.

Trajectoire Nous avons donc proposé une troisième approche qui consiste à déformer la trajectoire estimée sans l'ICP de manière à correspondre à la différence calculée par l'ICP. Plus précisément, on crée une trajectoire entre les instants $t(i - 1)$ et $t(i)$ en interpolant deux trajectoires : celle estimée à l'origine et la même trajectoire déplacée pour s'aligner avec l'estimation de l'ICP (plus de détails dans (KUBELKA et al., 2015)).

Modèle d'observation de l'odométrie visuelle

L'odométrie visuelle est similaire à l'ICP en ce qu'elle ne calcule qu'une position relative. Les mêmes problèmes d'intégration de cette information se posent donc et nous avons utilisé l'approche dite trajectoire.

Une difficulté de l'odométrie visuelle est que le facteur d'échelle n'est pas observable. En effet, une caméra monoculaire ne peut pas faire la différence entre une scène et un mouvement donnés et la même scène et le même mouvement agrandis d'un même facteur. C'est un problème restreint uniquement aux translations donc, dans ce travail, nous nous sommes contentés d'utiliser l'estimation d'attitude par l'odométrie visuelle.

4.3 Expériences

4.3.1 Bateau autonome

Évaluation

Que ce soit pour estimer les paramètres des filtres ou simplement l'évaluation de leur résultat, il est nécessaire d'avoir une vérité terrain. Dans des environnements intérieurs, on dispose de systèmes de capture de mouvement permettant de mesurer la position et l'attitude. Malheureusement ces systèmes proposent un espace de travail bien délimité de quelques mètres de côté et ne fonctionnent pas toujours en extérieur.

Nous avons donc découpé l'évaluation de l'attitude et celle de la position. Pour la position, nous utilisons un théodolite qui peut mesurer la position d'un point en continu

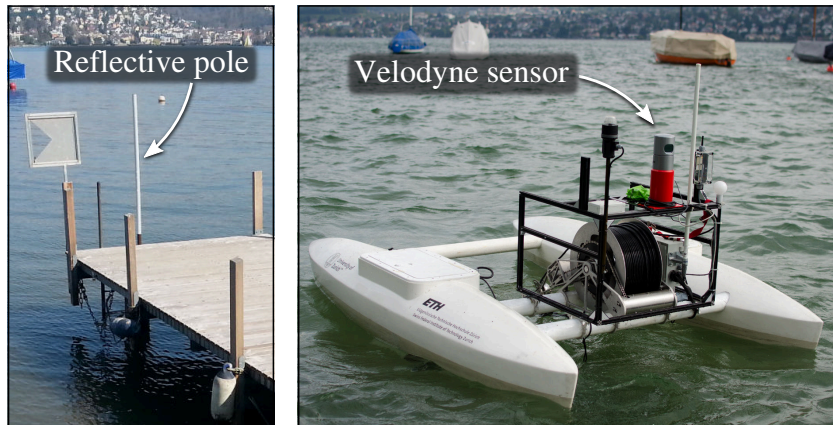


FIG. 4.3: Référence pour l'estimation d'orientation. À gauche : le poteau réfléchissant sur le ponton ; à droite : le capteur Velodyne HDL-32 monté sur Lizbeth. (Repris de (Hirtz et al., 2015).)

avec une précision d'un millimètre sur une portée de l'ordre du kilomètre. Pour l'attitude, nous avons placé dans l'environnement un poteau vertical recouvert d'une surface réfléchissante permettant son extraction sans ambiguïté dans un scan laser (figure 4.3). Si l'attitude est exactement estimée, la reconstruction de ce poteau par les données laser sera bien droite et verticale ; à l'opposé, si l'attitude est mauvaise, il devrait y avoir des distorsions et un défaut d'inclinaison. Deux capteurs ont été utilisés pour cette expérience : un Velodyne HDL-32 qui donne une capture de la scène 3D à une fréquence de 10 Hz et un Hokuyo UTM-30LX-EW monté sur un axe, moins cher et plus représentatif, qui fait une capture en 2 s (0,5 Hz, voir la figure 4.1(b)).

Les paramètres des différents modèles ont été optimisés sur des jeux de données spécifiques à l'aide de l'algorithme d'optimisation stochastique *covariance matrix adaptation evolution strategy* (CMA-ES) (HANSEN, 2006).

Estimation

La figure 4.4 compare l'estimation de position des filtres IC et MB3D par rapport à la trajectoire de référence donnée par le théodolite. On remarque que, si la trajectoire globale est correcte, il y a des décalages de plusieurs mètres par endroits. Nous avons ajouté la trajectoire brute indiquée par le GPS où il est apparent que le décalage des estimations est causé par celui, erratique, du GPS (voir en particulier le premier insert où ce décalage change de côté). On remarque aussi que la trajectoire centrée sur l'IMU semble très discontinue lorsque le décalage du GPS change (trajectoire en dents de scie dans le premier insert). Le filtre complémentaire utilisant le même modèle de déplacement que celui centré sur le modèle, les trajectoires sont très proches.

La figure 4.5 présente une évaluation plus quantitative de ces trajectoires en s'intéressant à plusieurs critères. Les premiers sont la précision de l'estimation, indiquée par les erreurs sur la position et la vitesse estimées. Les erreurs en position ne sont pas significativement différentes entre les filtres, essentiellement car la seule information absolue dont disposent les filtres vient du GPS dont l'erreur est montrée pour référence. Par contre,

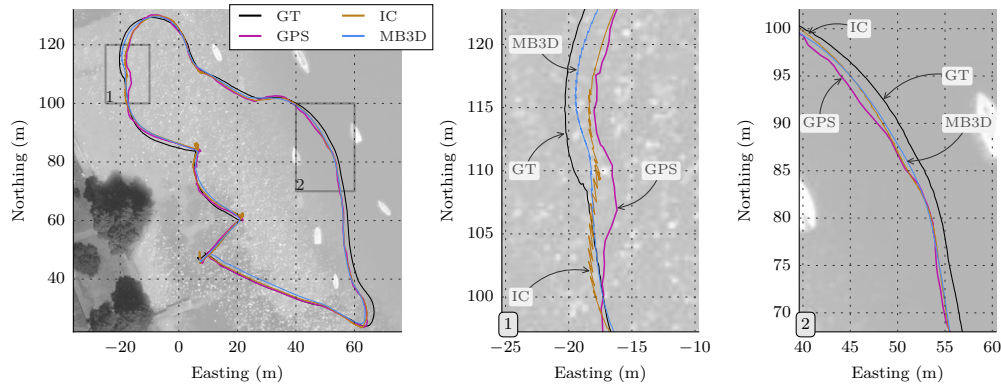


FIG. 4.4: Comparaison des trajectoires pour Lizbeth. À gauche : trajectoire globale projetée sur une image aérienne ; à droite : agrandissement de deux endroits de la trajectoire. En noir : vérité terrain ; en magenta : GPS seul ; en marron : estimation par le filtre centré sur l'IMU ; en bleu : estimation par le filtre basé sur un modèle (le filtre complémentaire en est assez proche et est omis pour clarifier le graphique). (Repris de (HIRTZ et al., 2015).)

on remarque que le filtrage d'une part et l'intégration d'un modèle dynamique du bateau d'autre part permettent de réduire l'erreur sur l'estimation de la vitesse. Enfin, notamment pour reconstruire des nuages de points, il peut être important que les trajectoires estimées soient lisses. L'estimation des discontinuités en position, vitesse et attitude par différence finie montre une corrélation entre l'erreur absolue et la discontinuité. Par contre, on remarque que le filtre basé sur un modèle est moins lisse dans son estimation de l'attitude. C'est probablement dû à une formulation linéarisée et trop simpliste des forces hydrostatiques.

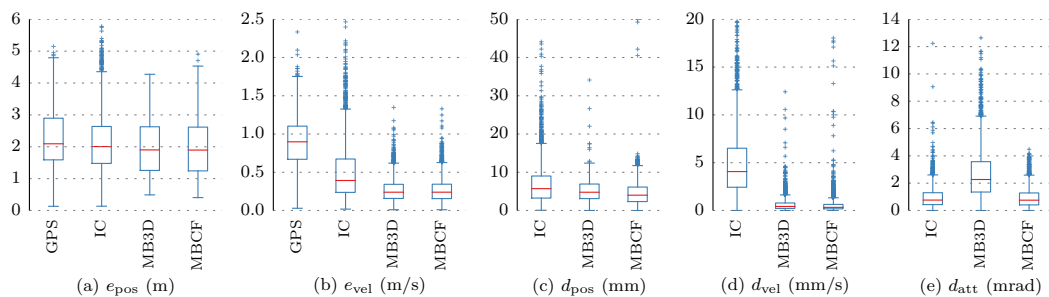


FIG. 4.5: Comparaison quantitative des filtres à l'aide des données du théodolite. Les métriques sont, de gauche à droite, l'erreur en position e_{pos} , l'erreur en vitesse (e_{vel}), la discontinuité (différence finie) en position (d_{pos}), la discontinuité en vitesse (d_{vel}), la discontinuité en attitude (d_{att}). (Repris de (HIRTZ et al., 2015).)

On peut confirmer ces derniers résultats en regardant l'inclinaison du poteau. La [figure 4.6](#) présente les inclinaisons évaluées à l'aide de chacun des trois filtres, IC, MB3D et MBCF, à l'aide du Velodyne ou de l'Hokuyo. Pour référence nous avons ajouté les inclinaisons pour un modèle complètement 2D. On remarque que le filtre complémentaire est confirmé comme le plus robuste pour l'estimation de l'attitude. La [figure 4.6d](#) et la

4 Estimation d'état en robotique

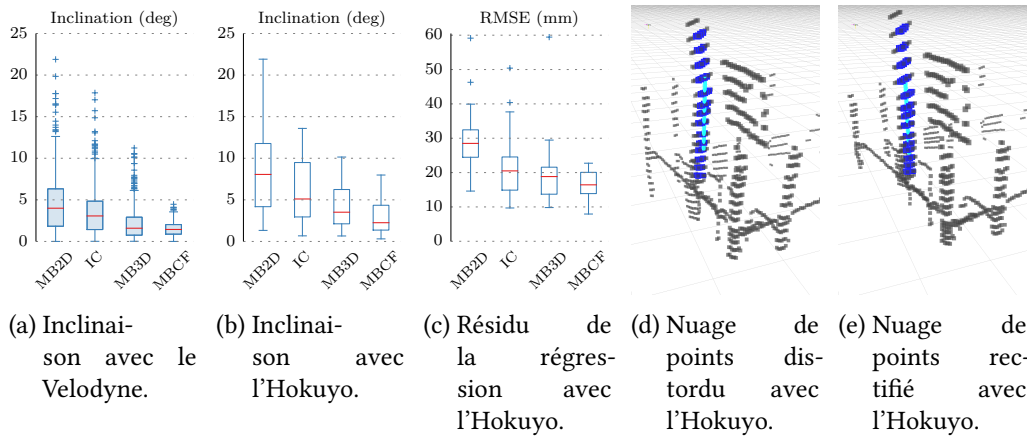


FIG. 4.6: Résultats de l'évaluation avec le poteau. (Repris de (HITZ et al., 2015).)

figure 4.6e montrent l'impact d'une bonne estimation d'attitude sur le nuage de points.

4.3.2 Robot au sol

Expériences

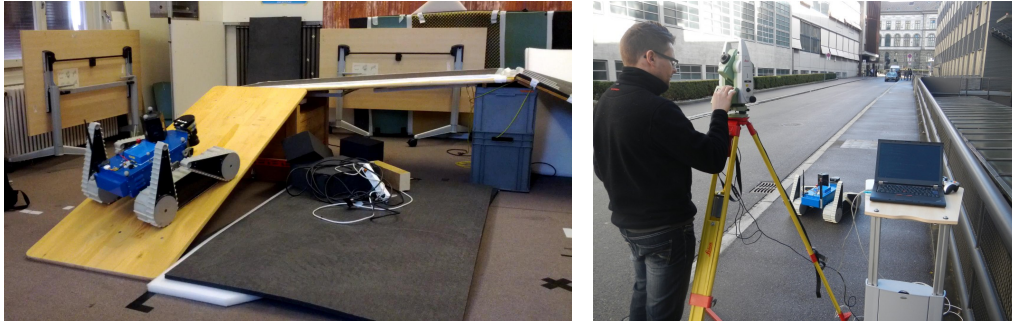
Pour évaluer les algorithmes d'estimation du robot au sol, nous avons enregistré ses données capteurs ainsi que la vérité terrain pendant plusieurs trajectoires. Une première partie des expériences a été réalisée en intérieur avec un système de capture de mouvement (Vicon) de manière à obtenir la vérité terrain sous la forme de la position et de l'orientation du robot. L'environnement (figure 4.7a) a été installé de manière à proposer une diversité de déplacements (rampes, pont, etc.). Dans cet environnement, nous avons enregistré l'équivalent de 2,4 km en 64 trajectoires.

Les dispositifs de capture du mouvement ont souvent un volume utile assez limité. Pour évaluer les algorithmes sur de plus grands espaces, nous avons effectué des tests en extérieur avec un théodolite pour mesurer la vérité terrain de la position (figure 4.7b). Ces tests ont eu lieu dans deux environnements différents : un canyon urbain et un parc avec des arbres et des escaliers. Dans ces environnements, nous avons enregistré environ 1,6 km en 10 trajectoires.

Modèles d'observation

À partir de ces données, le premier test qui nous intéresse est de comparer les différents modèles d'observation définis pour l'ICP. Pour cela, nous utilisons l'erreur relative e_{rel} , c'est-à-dire l'erreur finale en position divisée par la distance parcourue (en pourcentage), et l'erreur relative moyenne e_{avg} , soit la moyenne, pendant toute la trajectoire, de l'erreur relative.

La table 4.1 montre les résultats des différents modèles en séparant les trajectoires en intérieur et en extérieur. On remarque que la méthode en vitesse, la plus utilisée dans la littérature, est la moins précise, que ce soit en intérieur, ou en extérieur. La figure 4.8 montre un exemple des différences fines où l'on voit que la méthode en vitesse prend des raccourcis dans les coins.



(a) Environnement semi-structuré d'intérieur avec système de capture de mouvement. (b) Canyon urbain avec théodolite.

FIG. 4.7: Environnements des expériences pour NiftiBot. (Adapté de (KUBELKA et al., 2015).)

Modèle	Intérieur		Extérieur	
	e_{rel}	e_{avg}	e_{rel}	e_{avg}
Position incrémentale	0,4 0,7 1,2	0,1 0,1 0,2	0,8 1,5 11,0	0,7 2,4 6,1
Vitesse	1,0 1,3 2,3	0,1 0,1 0,3	0,9 1,8 12,2	0,8 2,5 6,1
Trajectoire	0,7 1,2 2,1	0,0 0,1 0,2	0,6 1,4 11,5	0,6 2,2 6,1

TAB. 4.1: Comparaison des modèles d'observation pour l'ICP. Pour chaque modèle et chaque métrique, les résultats sont présentés sous la forme premier|médiane|dernier quartile. (Adapté de (KUBELKA et al., 2015).)

La méthode de position incrémentale est sensiblement meilleure en intérieur où l'ICP dispose de suffisamment d'information dans l'environnement pour pouvoir complètement déterminer la position mais la méthode trajectoire est meilleure en extérieur, sur de plus grands environnements moins contraints.

Comparaison des modalités

Le deuxième test que nous souhaitons réaliser est celui de l'impact des différentes modalités. Plus précisément, parmi toute la combinatoire des modalités, nous nous intéressons à l'odométrie visuelle et à l'ICP; ainsi les informations de l'odométrie et l'IMU seront toujours utilisées. On va donc comparer quatre filtres : sans ICP ni odométrie visuelle, avec l'odométrie visuelle, avec l'ICP et avec les deux. Comme pour au-dessus, nous séparons les expériences en intérieur de celles en extérieur et nous utilisons l'erreur relative en pourcentage.

La table 4.2 montre les erreurs relatives pour chacune des trajectoires d'intérieur. On remarque une certaine variabilité avec certaines trajectoires qui présentent de plus grandes valeurs d'erreur. Plus précisément, un test de Wilcoxon¹ nous permet de montrer que l'ajout de l'information d'ICP améliore significativement les résultats alors que l'odométrie visuelle n'apporte pas de changement significatif. En effet, le problème principal de la

1. Wilcoxon signed-rank test avec $p < 0,05$.

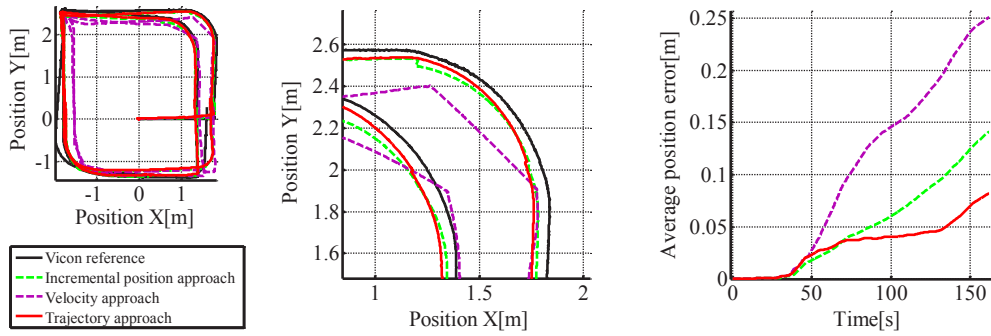


FIG. 4.8: Comparaison des modèles d'observation pour l'ICP sur une trajectoire donnée. À gauche : trajectoire globale ; au milieu : zoom sur un coin ; à droite : évolution de l'erreur moyenne au cours de la trajectoire. Ligne noire : vérité terrain ; tirets verts : méthode position incrémentale ; tirets violets : méthode vitesse ; ligne rouge : méthode trajectoire. (Repris de (KUBELKA et al., 2015).)

combinaison odométrie et IMU est le glissement des chenilles, notamment dans les pentes. L'odométrie visuelle que nous utilisons ne permet de corriger que les orientations et c'est pourquoi elle n'améliore pas significativement la performance. À l'opposée, l'ICP permet une correction des 6 degrés de liberté de la pose du robot et améliore bien les performances.

La [table 4.3](#) montre les erreurs relatives pour les trajectoires d'extérieur. Là encore on remarque une grande variabilité et, de manière plus marquée qu'en intérieur, on remarque que l'ICP améliore significativement les résultats du filtre. Par contre, l'odométrie visuelle, qui ne changeait pas significativement les résultats en intérieur, détériore maintenant grandement les performances. D'un point de vue théorique, la covariance d'une estimation effectuée à l'aide de plus de modalités sensorielles ne peut que diminuer si les hypothèses sont valides (en particulier bruit gaussien sans biais). La raison de cette détérioration se trouve justement en ce que l'odométrie visuelle casse sa représentation interne et ses estimations ne se rapportent plus correctement à l'état à estimer. En pratique, les caractéristiques des environnements, et en particulier la similarité des textures (fenêtres ou feuillage), expliquent les nombreuses estimations erronées de l'odométrie visuelle. Ainsi, l'ajout d'une modalité sensorielle n'est pas un gage d'amélioration de la performance du filtre et il est nécessaire de vérifier la qualité des entrées (plus de détails dans (KUBELKA et al., 2015)).

4.4 Discussion et conclusion

Nous avons montré dans ce chapitre qu'un outil commun d'estimation pouvait être adapté au traitement de deux types de robots assez différents dans leur conception, leurs applications mais surtout dans les caractéristiques de leurs mouvements. Le filtre de Kalman étendu, variante du filtre bayésien pour lequel les distributions de probabilité sont gaussiennes, permet de calculer une approximation de la distribution recherchée sous forme close. La représentation de la distribution est réduite à ses statistiques suffisantes,

Exp.	Dist. (m)	Durée (s)	Erreur relative (%)			
			O,IMU	O,IMU,VO	O,IMU,ICP	O,IMU,VO,ICP
1	47,42	254	2,17	2,30	1,71	0,79
2	36,52	186	1,99	2,21	0,36	0,14
3	48,74	244	3,15	2,63	0,50	0,18
4	29,40	237	2,22	2,06	0,42	0,45
5	82,10	585	2,51	2,24	0,90	0,71
6	74,64	452	2,05	3,64	0,98	1,24
7	74,65	387	1,70	1,72	2,28	0,58
8	30,57	194	1,98	3,42	1,59	2,29
9	26,58	287	2,67	2,23	1,90	1,19
10	26,57	236	1,53	3,94	0,77	2,11
11	26,96	208	1,25	1,20	0,95	0,66
12	29,13	211	1,27	1,29	0,88	0,87
13	26,35	180	1,37	1,25	0,94	0,77
14	40,23	240	6,58	6,70	0,88	0,99
15	21,01	167	5,26	5,27	0,61	0,57
16	19,04	209	5,94	5,95	0,55	0,60
17	10,95	405	3,44	2,89	2,15	2,05
18	8,65	238	2,87	2,77	1,36	1,38
19	9,36	284	4,14	3,91	1,83	1,85
20	9,02	282	2,90	3,36	2,73	2,65
21	10,82	308	3,79	3,23	1,43	1,41
22	9,45	237	5,36	5,45	2,66	2,68
23	12,75	204	2,65	2,84	2,66	1,79
24	7,81	179	1,58	1,83	2,82	3,06
25	10,85	165	3,85	4,14	3,25	2,17
26	10,83	163	2,36	1,84	0,62	0,68
27	12,79	237	15,42	14,95	2,48	2,53
28	12,07	239	28,42	27,07	2,89	2,98
Quartiles			2,0 2,7 4,0	2,1 2,9 4,0	0,8 1,4 2,4	0,7 1,2 2,1

TAB. 4.2: Erreurs relative pour chacune des combinaisons de modalités et chacune des trajectoires en intérieur. (Adapté de (KUBELKA et al., 2015).)

Exp.	Dist. (m)	Durée (s)	Erreur relative (%)			
			O,IMU	O,IMU,VO	O,IMU,ICP	O,IMU,VO,ICP
1	120,62	825	2,08	26,61	1,83	17,84
2	175,67	853	1,37	12,53	2,42	5,91
3	159,42	738	1,10	20,48	0,43	12,22
4	135,18	584	2,78	0,72	0,24	0,62
5	259,86	992	9,74	0,80	0,26	0,80
6	145,31	918	2,65	2,66	1,03	1,76
7	88,20	601	1,94	1,60	1,25	0,97
8	99,29	560	1,20	20,18	0,62	11,50
9	238,28	1010	9,10	0,62	0,58	0,43
10	203,23	1107	3,23	6,79	0,51	0,42
Quartiles			1,4 2,4 3,2	0,8 4,7 20,2	0,4 0,6 1,2	0,6 1,4 11,5

TAB. 4.3: Erreurs relative pour chacune des combinaisons de modalités et chacune des trajectoires en extérieur. (Adapté de (KUBELKA et al., 2015).)

vecteur moyenne et matrice de covariance, ce qui en fait une représentation compacte et efficace en termes de mémoire.

Par contre, même si le cadre général de filtrage est le même, il est nécessaire d'adapter les modèles dynamique et d'observation. Ainsi, le problème principal d'un ASV réside en la complexité des forces hydrostatiques et hydrodynamiques auxquelles il est soumis. En comparant des modèles différents par le niveau de détail du modèle, nous avons montré l'intérêt d'un modèle spécialisé par rapport à la simple intégration générique de l'IMU mais aussi l'intérêt d'une estimation d'attitude assez générique par rapport à un modèle complet. Nous avons ainsi une estimation lisse à la fois sur la vitesse et sur l'orientation.

Pour les robots au sol, nous nous sommes intéressés aux modèles d'observation des capteurs incrémentaux tels que l'ICP et l'odométrie visuelle qui fonctionnent en estimant le déplacement relatif entre deux captures de données. Le problème survient essentiellement lorsque la fréquence des capteurs est très différente. Dans un tel cas, il devient problématique de considérer l'observation comme relevant de l'instant présent alors qu'il s'agit en pratique de l'accumulation de tous les changements depuis l'évènement précédent. Nous avons montré que le meilleur modèle était de répartir explicitement la correction due à cette observation en veillant à maintenir la continuité.

Enfin nous avons mis en lumière le problème de la fusion multimodale d'information : chacune des modalités est supposée ne pas avoir de biais. Sous cette hypothèse, rajouter des capteurs amène nécessairement une amélioration de l'estimation, En pratique, cependant, il existe des cas pour lesquels cette hypothèse n'est plus vérifiée pour un capteur. Dans ces cas, en l'absence d'un meilleur modèle, plus complexe, dans lequel l'hypothèse serait à nouveau vérifiée, la solution serait d'ignorer la valeur du capteur. Mais il n'est pas aisé de détecter les cas pour lesquels c'est nécessaire (voir (KUBELKA et al., 2015)).

En conclusion, même si le modèle générique de fusion de capteurs pour l'estimation d'état est le même, il n'est pas possible d'utiliser un modèle identique pour tous les robots et les combinaisons de capteurs. La modélisation du robot et de ses capteurs est toujours nécessaire pour une estimation d'état satisfaisante.

5 Navigation en robotique

5.1 Introduction du problème

Autant la localisation et la cartographie sont importantes pour la robotique mobile, autant la capacité de planifier et exécuter des trajectoires est nécessaire à l'autonomie des robots mobiles. Lors de missions impliquant des robots, les comportements autonomes peuvent permettre aux opérateurs de se consacrer à d'autres tâches. De plus, lors d'opérations de recherche et sauvetage dans des zones dangereuses telles qu'à Fukushima (NAGATANI et al., 2013), avoir la capacité de planifier son chemin sur le robot, permettrait des procédures de secours telles que le retour à la base en cas de perte de communication.

La planification de chemin en 2D pour des robots d'intérieur peut être considérée comme largement résolue avec des logiciels libres disponibles (MONTEMERLO et al., 2002 ; MARDER-EPPSTEIN et al., 2010). Des algorithmes similaires peuvent s'appliquer à des scénarios d'extérieur tels que les voitures autonomes mais le problème devient alors essentiellement celui de la perception et de l'application des règles de conduite (CAMPBELL et al., 2010). Une fois l'espace libre identifié, on peut utiliser des algorithmes génériques de recherche de chemin dans des graphes tels que *Dijkstra* (E. W. DIJKSTRA, 1959), A^* (HART, NILSSON et RAPHAEL, 1968) ou D^* -Lite (KOENIG et LIKHACHEV, 2002). Il est possible d'adapter cette planification aux contraintes cinématiques du robot soit en utilisant des *state lattices* (treillis d'états) (PIVTORAIKO, KNEPPER et KELLY, 2009) ou par la spécification dans la fonction de cout de connexion entre les nœuds du graphe (RICHTER et WESTPHAL, 2010). Dans un espace de recherche en 2D, il y a même des solutions d'interpolation qui permettent de résoudre le problème commun aux grilles de discrétisation d'angle (FERGUSON et STENTZ, 2007 ; PHILIPPSEN et SIEGWART, 2005).

D'un autre côté, la planification de trajectoire en 3D a été beaucoup étudiée pour des robots aériens (BORTOFF, 2000 ; HE, PRENTICE et ROY, 2008 ; TISDALE, KIM et HEDRICK, 2009) ou sous-marins (PÊTRÈS et al., 2007). On trouve surtout des approches stochastiques permettant de s'arranger avec la plus grande dimension de l'espace de recherche comme les *probabilistic roadmaps* (feuilles de route probabilistes) (KAVRAKI et al., 1996) et les *rapidly exploring random trees* (arbres aléatoires d'exploration rapide) (LAVALLE et KUFFNER, 2001 ; KARAMAN et FRAZZOLI, 2011). Ces approches stochastiques permettent l'exploration rapide d'un espace de recherche de grande dimension pour obtenir une première solution qui peut ensuite être affinée.

Le problème des robots au sol est que l'on peut considérer qu'ils évoluent sur une variété 2D dans un espace 3D. Pour les robots d'exploration spatiale, on ajoute souvent l'information d'élévation sur une carte 2D (on parle alors parfois de 2,5D) (TOMPKINS, STENTZ et WETTERGREEN, 2004). Mais cette approche est insuffisante pour les environnements de recherche et sauvegarde où l'on peut être confronté à des environnements à plusieurs étages, des escaliers ou des tunnels. Il y a donc une distinction supplémentaire à faire entre les configurations libres, c'est-à-dire pas en collision avec un obstacle, et les

configurations acceptables pour le robot ; distinction qui n'est pas présente en 2D ou pour les drones aériens ou sous-marins. Cette distinction ne permet pas non plus l'utilisation efficace de méthodes de planification stochastique puisqu'elles supposent qu'il est relativement facile de décider si une nouvelle configuration échantillonnée est atteignable depuis une autre ou pas.

Certains robots d'inspection sont confrontés à des problèmes similaires. STUMM et al. (2012) se sont intéressés à un petit robot à roues magnétiques pour inspecter des canalisations en acier dans des centrales thermiques. Ce robot se déplace exclusivement sur la surface de ces canalisations (l'intérieur) qui est un objet 3D complexe ne pouvant pas se représenter correctement avec une carte d'élévation. Ils proposent en premier lieu une reconstruction 3D dense de l'environnement à l'aide du *tensor voting* (MEDIONI, TANG et LEE, 2000) à partir de nuages de points. Ensuite, ils font une recherche de chemin classique à l'aide d'A*. Pendant l'exécution, ils déroulent la trajectoire 3D sur des plans 2D locaux de manière à utiliser un algorithme de suivi de trajectoire 2D. Cette approche n'est pas directement applicable à notre robot qui n'adhère pas aux surfaces avec une orientation quelconque. De plus, certaines trajectoires nécessitent l'utilisation des *flippers* et notre robot a donc plus de degrés de liberté que le leur. Enfin, ils utilisent un système de localisation extérieur au robot (capture de mouvement) et font la reconstruction de l'environnement et la planification hors-ligne.

Nous avons donc adapté cette approche à notre robot (voir la [figure 3.5](#)) en couplant fortement la reconstruction de la représentation et la planification, en séparant l'espace des configurations et en couplant exécution et planification. Ce travail a été débuté par Srivatsa Mahesh lors de son travail de master et continué pour un article de conférence (COLAS, MAHESH et al., 2013).

5.2 Système

Notre système est fortement couplé dans son exécution ; pour autant, les concepts de perception, planification et exécution sont toujours opérants. Par perception, j'entends ici spécifiquement le traitement effectué sur les données sensorielles de manière à construire une représentation qui pourra être utilisée pour la planification. La planification sera l'exploitation de cette représentation pour décider des commandes que le robot devra exécuter dans la phase d'exécution.

5.2.1 Perception

Inférence structurelle

Les données sensorielles que nous avons à traiter sont des nuages de points 3D issus du capteur et alignés avec la variante d'ICP présentée dans la [section 3.3.2](#). C'est une information creuse, au sens où un nuage de points nous renseigne sur ce qui existe en chaque point du nuage, mais ne dit rien entre eux.¹ Nous avons besoin d'une représentation qui nous permette de décider :

- si une pose particulière est acceptable (libre de collision, avec assez de surface de support et avec une inclinaison limitée) ;

1. Par opposition, une grille d'occupation est dense au sens où à chaque endroit, l'information est présente.

– et s’il est possible de se déplacer entre deux poses voisines.

En particulier, il va falloir pouvoir répondre à ces questions n’importe où dans l’espace et il faut donc une représentation dense en ce sens. De plus, nous devons construire cette représentation en ligne, sur le robot, ce qui exclut des traitements trop coûteux en espace mémoire et surtout en temps de calcul.

Le *tensor voting* permet justement d’inférer, en un endroit quelconque de l’espace, la structure géométrique à partir d’un nuage de point. Plus précisément, les tenseurs votent pour estimer les caractéristiques et la saillance de primitives géométriques (plan, ligne, ou sphère dans notre cas). Ce processus est souvent effectué en deux étapes avec une première phase, *sparse voting* (vote creux), où l’on estime la structure (le tenseur) en chacun des points d’entrée et une deuxième phase, *dense voting* (vote dense), où l’on peut calculer le tenseur aux endroits désirés, typiquement des cellules dans une grille régulière. On obtient alors une grille 3D de cellules contenant chacune la saillance et les paramètres de chaque tenseur : vecteur normal pour un plan et direction principale pour une ligne.

Nous nous sommes intéressés essentiellement aux tenseurs de plans pour estimer les surfaces locales avec leur orientation. Un paramètre clé du *tensor voting* est la taille du noyau qui définit l’échelle à laquelle sont considérés les tenseurs. Nous avons choisi cette taille de manière à correspondre à la taille du robot, c’est-à-dire en pratique, de manière à ce que l’orientation du plan inféré correspondra en général à celle du robot à cet endroit.

Filtre de distance

Le nombre de votes de la phase de *sparse voting* est quadratique en nombre de points du nuage puisque chaque point est censé voter pour tous les autres. Nous avons proposé une implémentation sur carte graphique dans un autre travail (LIU et al., 2012) mais le robot ciblé n’en est pas pourvu. On peut néanmoins réduire à un nombre linéaire en combinant un noyau à support borné et une limitation du nombre de point sur ce support, c’est-à-dire de la densité des points. Le nombre de votes devient alors $k \times N$ avec N le nombre de points et k le nombre de point maximal sur le support du noyau. Le problème est que le nuage de points peut présenter une densité de points arbitrairement grande à certains endroits et faible dans d’autres. On ne peut donc pas se contenter de sous-échantillonner de manière uniforme puisque l’on perdrait de l’information là où peu de points la présentent avant de limiter utilement la complexité.

Nous proposons donc de réduire à la fois le nombre de points et la densité maximale en supprimant les points qui sont trop proches les uns des autres. L’[algorithme 2](#) montre l’algorithme, avec d la distance la plus proche acceptée.

ALG. 2 Algorithme de filtrage des points par la distance.

```

for all  $p \in$  nuage de point do
  for all  $n \in$  nuage de point tel que  $\|n - p\| \leq d$  do
    supprime  $n$ 
  end for
end for

```

Cet algorithme peut être réalisé efficacement à l’aide d’une recherche de plus proche voisins rapide (ELSEBERG et al., 2012) et en marquant les points à supprimer plutôt que

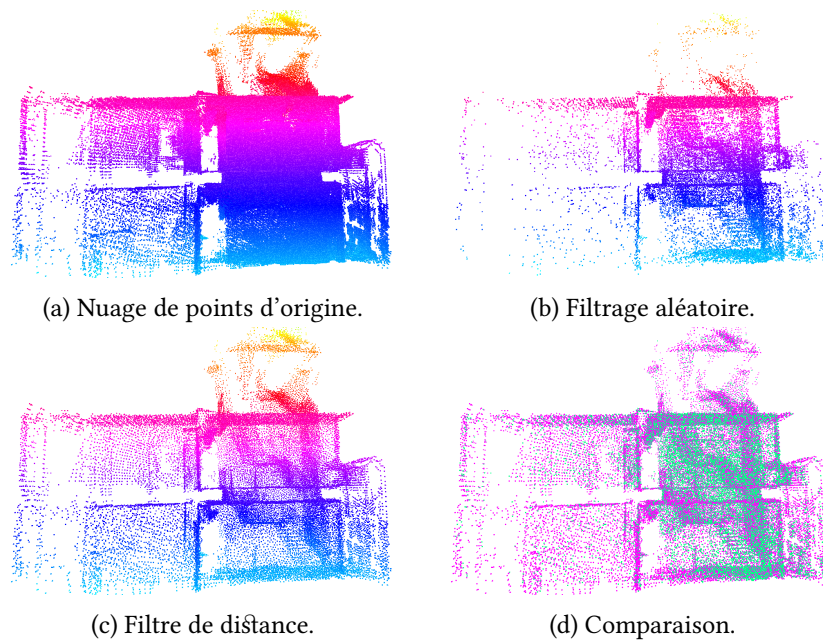


FIG. 5.1: Comparaison du filtre de distance avec un filtre aléatoire. En haut à gauche : nuage de points complet représentant deux étages avec la cage d'escalier à gauche ; les points (488 918) sont colorés en fonction de leur élévation. En haut à droite : nuage de points après filtrage aléatoire (36 014 points). En bas à gauche : nuage de points après filtrage de distance (36 014 points). En bas à droite : superposition des nuages de points ; en vert le filtre aléatoire et en magenta le filtre de distance. (Adapté de (COLAS, MAHESH et al., 2013).)

les supprimant directement pour ne pas avoir à reconstruire la kD tree. La complexité de cette phase passe donc de $O(N^2)$ à $O(N \log N)$. Il ne s'agit pas ici de construire une représentation optimale en minimisant le nombre de points, mais plutôt d'avoir un compromis raisonnable pour le temps de calcul du *sparse voting*.

La figure 5.1 montre une comparaison entre le filtre de distance et le sous-échantillonnage aléatoire. Comme prévu, ce dernier perd de nombreux points dans les zones de faible densité alors que le filtre de distance garde une représentation plus équilibrée.

Tensor voting dense et paresseux

Après la phase de *sparse voting*, on peut faire la phase de *dense voting* ; c'est-à-dire calculer, en chaque endroit, les tenseurs d'après le vote de chaque point. La complexité est donc en $O(N \times G)$ avec G le nombre de cellule dans la grille dense. Ce nombre de cellule est cubique en fonction de l'inverse de la taille du côté de la cellule, ce qui peut être prohibitif même pour des environnements d'intérieur : un quart de milliards de cellules de 5 cm dans un bâtiment de $100 \times 20 \times 16$ m.

Cependant la plupart de ces cellules sont vides et, contrairement aux robots volants, beaucoup plus ne sont pas pertinentes pour un robot au sol (plafonds, lampes, parties

TAB. 5.1: Évaluation du *dense voting* paresseux (adapté de (COLAS, MAHESH et al., 2013)).

Environnement	nombre de cellules		moyenne par cellule	
	total	nécessaire	nombre de requêtes	temps de calcul
Escalier	5 509 350	131 586	1805	0,474 ms
Bureau	1 140 700	38 070	1016	0,577 ms

supérieures des murs et des obstacles). Ainsi, nous proposons de n'effectuer le *dense voting* qu'aux endroits où c'est nécessaire. Ce ne peut être déterminé que pendant la phase de planification et d'expansion de l'espace atteignable. Plutôt que de tout calculer *a priori*, nous effectuons le *dense voting* à la demande en utilisant un mécanisme de mémoïsation à l'aide d'une table de hachage pour ne pas le faire plusieurs fois.

La [table 5.1](#) compare le nombre de cellules dans des environnements avec celui nécessaire à la planification. Pour donner un élément de comparaison, la boîte englobante du robot est de l'ordre de 400 cellules. On remarque donc que le calcul à la demande réduit le nombre de cellules calculées d'un ordre de grandeur. De plus, le nombre moyen de requêtes pour chaque cellule en regard du temps de calcul démontre l'intérêt de la mémoïsation.

5.2.2 Planification

Réduction de dimension

Les techniques de recherche de chemin dans un graphe de la famille de Dijkstra et A^* ont des problèmes de passage à l'échelle lors de l'augmentation de la dimension de l'espace de planification à cause de l'augmentation exponentielle du nombre de voisins, c'est-à-dire du facteur de branchement. L'espace des configurations statiques de notre robot est de dimension 11, avec les 6 dimensions de la pose du corps central, les 4 angles des *flippers* et l'angle entre les bogies.

Faire une recherche exhaustive dans cet espace serait trop coûteux pour une planification en ligne, mais on peut réduire la complexité à l'aide des observations suivantes :

- l'assiette (tangage et roulis) et l'angle entre les bogies sont complètement déterminés par la position du robot et le terrain autour de cette position ;
- l'orientation (cap) est déterminée par la séquence des positions puisque le robot n'est pas holonome ;
- les angles des *flippers* peuvent être déterminés de manière à aider le robot à effectuer la séquence de positions.

On s'aperçoit alors qu'il est possible de planifier un chemin dans l'espace à trois dimensions de la position du robot et ensuite de recouvrir les dimensions restantes grâce à la connaissance du terrain. Pour cela, il est nécessaire de s'assurer que le chemin est faisable en termes d'assiette, d'orientation et d'angles des *flippers* en prenant en compte les contraintes que ça induit dans la connectivité du graphe de planification. Cela se traduit aussi par une fonction de cout entre des nœuds voisins qui exprime les différences de difficulté liées à ces dimensions dépendantes.

Pour l'algorithme spécifique de recherche de chemin, nous avons choisi D*-Lite (KOE-NIG et LIKHACHEV, 2002). Il s'agit d'un algorithme de recherche de chemin de moindre cout dans un graphe à l'aide d'une heuristique comme A* mais, à la différence de ce dernier, il peut être utilisé pour adapter le chemin dans le cas d'un changement dans les couts ou dans la position du robot.

Connectivité

La planification de chemin en 2D se fait généralement sans référence explicite aux obstacles mais en n'introduisant pas les régions occupées dans le graphe. Dans le même esprit, nous recherchons les chemins faisables (pas seulement libres de collision) en restreignant les connexions aux cellules faisables. Ainsi, nous considérons une connectivité générique de 26 voisins (les directions cardinales et les diagonales en 3D) et nous vérifions, durant l'expansion d'une cellule, lesquels de ces 26 voisins sont faisables depuis cette cellule.

Plusieurs critères entrent en jeu dans la définition d'un voisin acceptable :

- le robot n'est pas en collision ;
- il y a une surface pour supporter le robot ;
- la surface est orientée de manière à ce que le robot ne se renverse pas.

Le critère de non-collision est vérifié en regardant l'intersection entre le nuage de points laser et la boîte englobante du robot positionné à l'aide de la surface en cette cellule. Le critère de support est assuré grâce à un seuil sur la saillance de la surface de support telle que calculée par le *dense voting*. Enfin le critère d'orientation se base sur l'angle entre la verticale et le vecteur normal à la surface, estimé lui aussi par le *dense voting*.

Fonctions de cout

D*-Lite fonctionne, comme Dijkstra et A*, avec une liste de nœuds en attente qui sont considérés en fonction du cout total depuis la position de départ :

$$g(c) = \min_{n \in \mathcal{N}} \{g(n) + \text{cost}(n, c)\},$$

avec g le cout depuis le départ, \mathcal{N} la liste des voisins acceptables de la cellule c considérée et $\text{cost}(n, c)$ le cout de transition de la cellule n à la cellule c .

Pour cette dernière fonction de cout, nous définissons les éléments suivants :

- saillance : les voisins les plus saillants sont préférés ;
- stabilité : les voisins avec les pentes les moins fortes sont préférés ;
- distance : les voisins les plus proches sont préférés ;
- orientation : les voisins pour lesquels la direction du robot suit la plus grande pente sont préférés.

Ces critères sont assez évidents à l'exception peut-être du dernier qui est motivé par la conception du robot. En effet les *flippers* permettent de franchir un fossé en agrandissant le polygone de sustentation, mais ils permettent aussi une plus grande stabilité dans l'axe de déplacement du robot. Cela se traduit par la possibilité d'une inclinaison plus grande en tangage qu'en roulis. En pratique notre robot peut grimper une pente de 45° mais pas s'y déplacer en diagonale. C'est une caractéristique commune aux robots équipés de *flippers* qui allongent le rectangle de support dans la seule direction du mouvement.

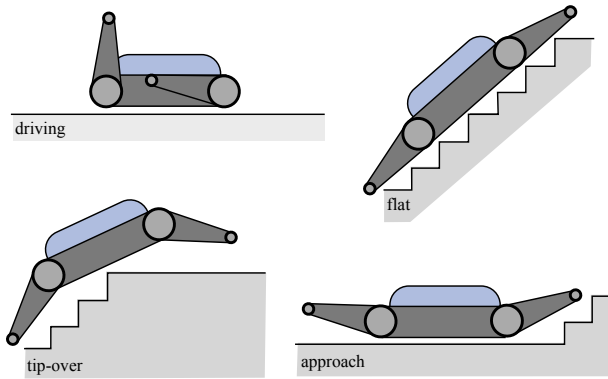


FIG. 5.2: Configurations des *flippers* pour franchir des obstacles positifs ou négatifs (repris de (COLAS, MAHESH et al., 2013)).

Pour obtenir une pondération entre ces différents éléments, nous avons optimisé ces paramètres sur des trajectoires de test en visant à réduire conjointement la longueur et la courbure du chemin. Cela conduit à un poids assez grand sur la composante de distance pendant que les autres composantes contribuent à un chemin plus lisse.

Configuration

Une fois le chemin obtenu sous la forme d'une suite de positions il est nécessaire de déterminer les angles des *flippers*. Une technique serait de définir ces angles à l'aide de la surface de support de manière à ce que chaque *flipper* ait le meilleur contact possible. Cette idée vient du rôle stabilisateur des *flippers*. Mais ils ont aussi un rôle pour aider aux transitions. Par exemple, lorsque le robot doit grimper sur un obstacle, les *flippers* avant doivent rester fixes et non pas chercher à s'adapter à la surface en se repliant progressivement. De la même manière, en s'engageant dans une pente descendante, les *flippers* arrière peuvent faciliter le basculement en poussant plutôt qu'en restant sur la surface.

Nous avons donc regroupé les configurations des *flippers* en quatre postures suivant quatre situations typiques (figure 5.2) :

- **driving** (conduite) : les *flippers* sont repliés pour réduire les frottements au sol et libérer le champ de vision des capteurs pour de la navigation en terrain facile ;
- **flat** (plat) : les *flippers* sont étendus à plat pour augmenter la stabilité et la traction dans les pentes ;
- **approach** (approche) : les *flippers* avant sont levés pour négocier un obstacle positif pendant que les *flippers* arrière sont étendus pour prévenir un basculement ;
- **tip-over** (basculement) : les *flippers* sont dirigés vers le sol, à l'avant pour amortir et à l'arrière pour aider au basculement lors d'obstacles négatifs.

Concrètement, une posture est associée à chaque position du chemin planifié. Cela est effectué en deux passes : d'abord les postures permanentes, **driving** et **flat**, en fonction de la pente ; ensuite les postures de transitions, **approach** et **tip-over**, en fonction du signe du changement de pente.

5.2.3 Exécution

L'exécution en boucle ouverte de commandes pré-calculées n'est pas faisable à cause de l'impact croissant des incertitudes : le robot n'effectue jamais exactement les commandes données et n'arrivera donc pas à l'endroit escompté. L'approche classique est alors de suivre une trajectoire figée à l'issue de l'étape de planification, en calculant les commandes à effectuer en fonction de la position actuelle du robot par rapport à la trajectoire. On trouve alors un compromis entre suivre de près la trajectoire et avancer vers le but lorsque l'on s'écarte de la référence.

Pour les robots au sol, la plupart des algorithmes de suivi se placent en terrain plat en 2D. Le suivi en 3D sur des surfaces quelconques nécessite un calcul plus compliqué de la distance entre la position actuelle du robot et la trajectoire pour ensuite savoir comment s'en rapprocher. STUMM et al. (2012) ont proposé une approche complexe avec des points de contrôle autour desquels l'espace est supposé localement plat et où la trajectoire 3D est projetée en 2D pour permettre la mise en œuvre d'algorithmes de suivi de trajectoire 2D.

À l'opposé, notre approche se base sur la capacité de l'algorithme D*-Lite de recherche de chemin de faire de la replanification assez vite en ne recalculant que les nœuds nécessaires. Cette capacité permet de réagir lorsque les coûts entre les nœuds changent, c'est-à-dire en cas de changement de l'environnement, mais aussi lorsque la position de départ du robot change, c'est-à-dire au cours de son déplacement. Lorsque le robot suit principalement le chemin planifié initialement, la replanification est instantanée puisque les nœuds sont déjà explorés. En cas de déviation du chemin, l'algorithme doit alors étendre quelques nœuds supplémentaires ce qui peut nécessiter des vérifications de collision et du calcul de *dense voting* sur quelques cellules ce qui n'est pas prohibitif pour une exécution en ligne.

Le chemin, généré à partir d'une grille 3D avec 26 voisins présente un crénelage reconnaissable puisque les changements de direction ne peuvent pas être moins de $\frac{\pi}{4}$. Ces changements de direction assez soudains ne sont pas idéaux pour les robots mobiles et en particulier sur des terrains difficiles tels que des escaliers puisqu'ils peuvent entraîner plus de glissement. D'un autre côté, il faut éviter de trop prendre de raccourcis de peur d'entraîner des collisions. Nous calculons donc la commande qui permet d'aller le plus loin dans le chemin planifié sans jamais s'éloigner de plus d'une distance donnée du chemin.

La **figure 5.3** présente le diagramme général de notre système. Les données des capteurs (laser, odométrie et centrale inertielle) sont utilisées pour la localisation du robot basée sur l'ICP. Les scans laser sont aussi accumulés en une carte 3D sous la forme d'un nuage de points. Cette carte est alors passée à notre algorithme en tant qu'entrée du *sparse voting*. La planification peut ensuite, à mesure des besoins de l'expansion des nœuds du graphe de recherche, effectuer des requêtes de *dense voting*. Une fois le chemin initial, on peut commencer la boucle d'exécution qui effectue successivement une planification à partir de la position actuelle, la décoration du chemin pour ajouter les postures des *flippers* et l'exécution proprement dite. Cette boucle est effectuée à une fréquence fixe (5 Hz) et se termine lorsque le robot est assez proche du but.

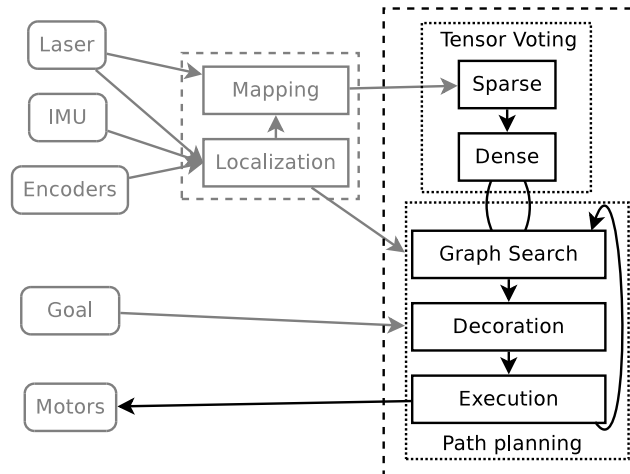


FIG. 5.3: Diagramme général du système (repris de (COLAS, MAHESH et al., 2013)).

5.3 Expériences

Tous les tests commencent par une phase de cartographie pendant laquelle le robot est télé-opéré de manière à découvrir son environnement. Ensuite le but est défini dans la représentation 3D à l'aide d'un marqueur interactif (Gossow et al., 2011). Le but est validé par le planificateur et éventuellement légèrement déplacé de manière à ce qu'il soit bien exempt de collision, sur une surface assez saillante et avec l'orientation correspondant à la surface. Cela permet d'éviter de rechercher un chemin vers un but non atteignable, et aussi de faciliter l'utilisation du système puisqu'il n'est pas toujours évident d'orienter correctement un marqueur 3D sur une vue projetée.

5.3.1 Terrain plat

Le premier test s'apparente plus à un test de régression vis-à-vis de la planification 2D et vise à valider notre système dans un environnement plat. Ces expériences permettent aussi d'évaluer la performance générale du système. L'environnement est un bureau avec des tables, des chaises et des poubelles. Le but et le point de départ sont positionnés alternativement d'un coin à l'autre.

La figure 5.4 montre une vue de dessus de l'environnement. Le chemin calculé par la planification est indiqué en vert présente bien l'aspect crénelé dû à la grille. L'exécution de chemin produit des trajectoires, en rouge, plus lisses.

La table 5.2 présente les temps de calcul sur un processeur d'ordinateur portable (core i7-2640M). À l'exception de la planification initiale du premier trajet effectué, les planifications initiales des trajets suivants prennent de l'ordre de la seconde. Cela s'explique parce que l'essentiel des calculs du *dense voting* ont été effectués pour le premier trajet et sont ensuite réutilisés pour les trajets suivants. De plus, le temps moyen de replanification est suffisamment en dessous des 200 ms de la période de notre boucle de contrôle, ce qui valide notre approche de replanification rapide à chaque étape de temps.

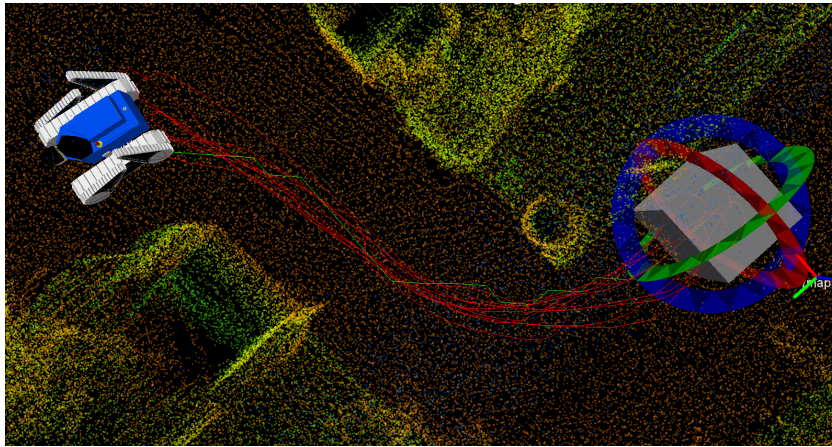


FIG. 5.4: Vue de dessus des expériences en terrain plat. Le robot est représenté sur la gauche et le marqueur indiquant le but est sur la droite. La ligne verte montre le chemin calculé par la planification initiale. La ligne rouge montre le chemin parcouru par le robot au cours de dix aller-retours. Les points du nuage de points utilisés pour la planification sont colorés en fonction de l'altitude avec le sol en orange, les tables en vert et le plafond en bleu. (Repris de (COLAS, MAHESH et al., 2013).)

TAB. 5.2: Temps de calcul et de trajet pendant les expériences en terrain plat (adapté de (COLAS, MAHESH et al., 2013)).

	Trajet									
	1	2	3	4	5	6	7	8	9	10
Planification initiale (s)	29,63	0,34	0,57	0,28	0,46	0,37	1,04	0,94	0,36	1,34
Durée du trajet (s)	51,6	73,4	56,0	52,2	51,6	54,0	51,0	54,2	56,0	56,0
Replanification (total en s)	6,71	5,01	3,27	1,07	1,56	1,35	0,65	0,32	3,39	0,25
Replanif. (moyenne en ms)	26,0	13,9	11,7	4,1	6,1	5,0	2,6	1,2	12,1	0,9
Nombre d'étapes	258	367	280	261	258	270	255	271	280	280

5.3.2 Escalier tournant

L'avantage de notre approche comparée à de la planification en 2D ou 2,5D est de pouvoir traiter des environnements complexes en 3D de la même manière qu'un environnement plat tel qu'un bureau. Par définition, les techniques 2D ne peuvent pas grimper des escaliers puisque ceux-ci ne sont pas traités différemment des murs. Les techniques 2,5D ne sont, quant à elles, pas capables de gérer correctement des environnements multi-niveaux tel qu'un bâtiment avec plusieurs étages ou même simplement une cage d'escalier. Avec notre approche, la seule différence par rapport à un environnement 2D est l'utilisation active des *flippers*.

Nous avons testé notre algorithme dans plusieurs cages d'escalier. L'une d'entre elles est un escalier légèrement tournant qui présente donc la difficulté que monter ou descendre en ligne droite n'y est pas possible. Il faut donc tourner, ce qui implique une plus grande chance de dérapage et donc de sortir de la trajectoire prévue initialement.

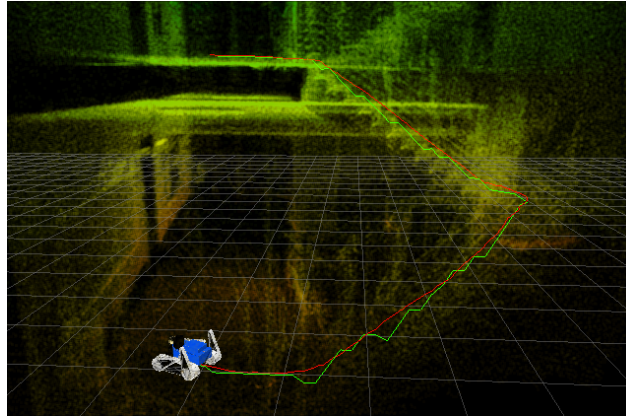
La [figure 5.5](#) montre un exemple de planification et d'exécution de chemin dans cet escalier. Sur cet exemple, le robot a mis 387 s pour monter et 260 s pour descendre. La différence est réellement due au dérapage puisque dans les deux cas la vitesse linéaire de consigne était fixée à 5 cm s^{-1} . Les temps de planification initiales étaient respectivement de 27,3 s et 20,5 s, avec des temps moyens de replanification de respectivement 11,0 ms et 21,0 ms. Ces nombres sont similaires à ceux présentés dans la [table 5.2](#) ce qui indique que le temps de calcul n'est pas directement corrélé à la complexité de l'environnement.

La [figure 5.6](#) montre une superposition de photos du robot dans l'escalier à des phases différentes de l'exécution, mettant en avant les différentes configurations des *flippers* nécessaires au franchissement d'un escalier.

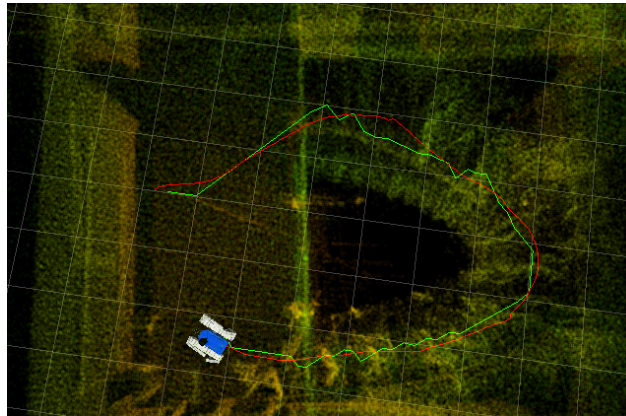
5.4 Discussion et conclusion

Dans ce chapitre, nous avons proposé un système permettant à un robot au sol de se déplacer dans des environnements 3D complexes et, comme exemple typique, de grimper des escaliers. Ce système part de données sous forme de nuages de points et ne prétend pas construire une reconstruction riche et complète de l'environnement mais simplement évaluer la possibilité de franchissement à l'aide d'une inférence structurelle locale et paresseuse. Nous avons évalué notre approche dans un environnement facile 2D et dans une cage d'escalier difficile pour montrer que notre robot est capable de le monter et le descendre.

Jusqu'à présent, notre approche n'a été testée et programmée que pour des environnements statiques. Ainsi, la carte est acquise avant la planification initiale et n'est jamais mise à jour. Le passage à un environnement dynamique présente deux difficultés différentes : les obstacles dynamiques à éviter à court terme, et les changements significatifs d'environnement tel que l'effondrement d'une partie d'un bâtiment ou, plus fréquemment, la fermeture d'une porte ou le déplacement d'un meuble. Ces deux types de changement doivent être traités correctement par le logiciel de cartographie en amont. Si ce dernier est capable d'indiquer l'ajout ou la suppression dans la carte d'un ensemble de points alors le résultat du *tensor voting* peut être mis à jour pour les cellules touchées et les modifications de leurs valeurs de coût qui en résultent peuvent être prises en compte pour la planification. Le choix de D*-Lite comme algorithme de planification permet ainsi



(a) Vue de côté



(b) Vue de dessus

FIG. 5.5: Exemple de franchissement d'un escalier. Le chemin initial est indiqué en vert. Le chemin effectué est indiqué en rouge. Les points du nuage de points sont colorés par leur altitude : en orange le niveau inférieur et en vert le niveau supérieur. (Adapté de (COLAS, MAHESH et al., 2013).)

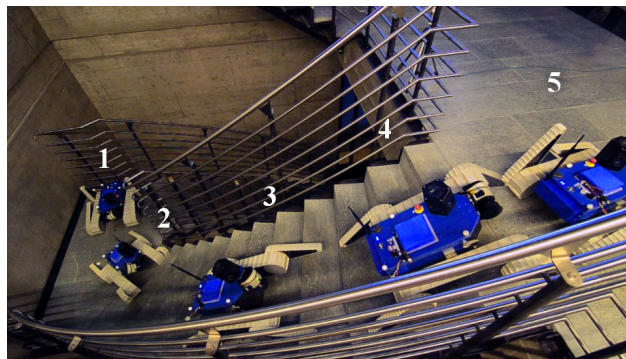


FIG. 5.6: Différentes phases de franchissement d'un escalier. 1 : conduite à plat sur le palier (driving) ; 2 : entrée sur l'escalier (approach) ; 3 : montée (flat) ; 4 : basculement (tip-over) ; 5 : conduite à plat sur le palier (driving). (Repris de (COLAS, MAHESH et al., 2013).)

de ne refaire que les calculs nécessaires suite à ces changements plutôt que d'invalider tout le plan et de devoir le recalculer.

Plus généralement, notre système est encore sensible à la qualité de la représentation. Les vecteurs normaux aux surfaces, par exemple, jouent un rôle central à la fois pour savoir si une cellule est traversable ou non, mais aussi pour choisir la configuration des *flippers*. La saillance calculée par le *tensor voting* est aussi importante pour déterminer si la surface est suffisamment plane. Cependant, la valeur de cette saillance dépend fortement de la densité locale des points du nuage et il faut donc mettre en œuvre des paramètres pour avoir une échelle raisonnable sur cette information.

Enfin, le choix de la replanification à chaque étape de l'exécution, plutôt qu'essayer de suivre une trajectoire n'est pas sans compromis. Il permet d'éviter le problème géométrique complexe de se situer par rapport à une trajectoire 3D sur une surface implicite avec des obstacles et de calculer des commandes motrices pour se rapprocher de cette trajectoire. Cependant, replanifier génère un nouveau chemin sans rapport nécessairement à l'exécution passée. Plus précisément, il ne prend pas en compte le fait que le robot vienne juste de franchir une arête dans l'escalier et doive rester encore dans la configuration de transition pour les *flippers*. Pour résoudre ce problème, nous avons ajouté une mémoire des éventuelles transitions en cours d'exécution.

En conclusion, en combinaison avec une estimation d'état et une cartographie embarquée, cette approche peut permettre à des robots au sol de naviguer dans des environnements complexes en 3D.

6 Projet de recherche

6.1 Introduction

6.1.1 Motivation

L'objectif premier de la robotique est de développer des artefacts à même de faciliter la vie des humains. Il existe de nombreux robots actuellement en usage, pour la plupart dans l'industrie (automobile, électronique et chimie en majorité) puis pour des usages professionnels (défense, agriculture en majorité). Le marché de la robotique domestique, bien qu'en forte expansion, est pour l'instant largement limité aux aspirateurs autonomes, aux drones personnels ou autres gadgets (IFR, 2015).

Pourtant la robotique domestique pourrait jouer un rôle important, notamment dans le cadre de l'assistance à la personne qui est l'un des axes prioritaires du plan France Robots Initiatives (MRP, 2013) mais aussi dans les domaines d'application et les marchés mis en avant pour le plan de financement de la recherche en robotique de l'union européenne (SPARC, 2014). Que ce soit pour des personnes âgées en perte d'autonomie ou souffrant de pathologies longues pouvant s'accommoder d'une médicalisation légère, il y aurait de nombreux avantages, tant économiques qu'humains, de permettre à ces personnes de rester à leur domicile. Pour cela, il est nécessaire de garantir un suivi adapté à leur condition, mais aussi d'étendre leur autonomie par l'entremise d'une adaptation de leur environnement et du déploiement de robots idoines (CAMARINHA-MATOS et VIEIRA, 1999 ; BROEKENS, HEERINK et ROSENDAL, 2009 ; OKAMURA, MATARIC et CHRISTENSEN, 2010).

Or, si la robotique est présente dans les chaînes de production, c'est qu'il y a été possible de concevoir des environnements dédiés aux robots en cantonnant les opérateurs qualifiés à des zones limitées et sécurisées. La simplification de l'environnement (éclairage constant, absence d'imprévu...) a été la clé de l'émergence de la robotique industrielle. À l'opposé, la recherche en robotique domestique a souvent eu pour objectif de s'adapter à l'habitat tel quel. Cela implique une grande diversité de problématiques comme la perception d'une scène complexe avec une large gamme de conditions (distance, point de vue, éclairage...), la navigation en environnement encombré, inconnu et dynamique et la manipulation fine et *compliant*.

Pour ces raisons, je pense qu'il est utile d'envisager le problème sous l'angle de la robotique située, c'est-à-dire des interactions entre les robots, l'environnement et les utilisateurs. La question n'est plus de réussir à obtenir des robots qu'ils réalisent seuls une tâche prédéfinie, mais plutôt d'intégrer les robots dans un environnement adapté pour leur permettre d'aider l'utilisateur à une variété de tâches. De la même manière, il est important de rendre l'utilisateur acteur dans le processus d'adaptation du système à ses propres besoins.

6.1.2 Objectifs

Mon but est de faciliter l'accès et l'utilisation de robots au quotidien pour tous types d'utilisateurs. Pour cela, il faut en premier lieu que les robots puissent fonctionner correctement, en continu et sur une longue durée dans les environnements dans lesquels on souhaite les placer. Ainsi, la robotique située me semble être le compromis adéquat entre la recherche d'une autonomie complète dans n'importe quel environnement et la conception complète de l'environnement autour d'une tâche robotique restreinte. Ma vision est qu'il est plus avantageux pour la robotique de service de déployer quelques capteurs dans l'environnement que d'essayer d'augmenter la complexité et l'autonomie des seuls robots.

En second lieu, il faut que les robots puissent s'adapter à ce que l'on souhaite leur voir faire. Les robots peuvent potentiellement résoudre différentes tâches mais, à l'heure actuelle, la tendance reste à les cantonner à un usage prédéfini qui aura été spécifié et programmé a priori. Pour étendre la gamme d'utilisation de ces systèmes, une solution est alors de permettre la programmation de nouvelles tâches directement par les utilisateurs. Ainsi les robots seront non seulement plus utiles puisque plus adaptés à l'usage souhaité, mais aussi mieux compris et acceptés par leurs utilisateurs.

Mon projet de recherche s'articule donc essentiellement autour des deux questions scientifiques suivantes.

- *Comment tirer parti de la combinaison d'une infrastructure de capteurs et de robots mobiles ?*
- *Comment permettre à un utilisateur non-expert d'enseigner une tâche robotique ?*

Je me propose de répondre à ces questions principalement dans le cadre applicatif de l'habitat intelligent composé d'un réseau de capteurs et d'un ou plusieurs robots mobiles au domicile d'habitants non-experts. Cependant ni ces questions ni mes contributions ne seront réduites à cette application particulière et pourront par exemple s'appliquer au suivi environnemental ou à l'inspection robotique.

Mon projet de recherche vise donc les objectifs suivants :

1. *augmenter les capacités de robots mobiles à l'aide d'un réseau de capteurs distribués dans l'environnement ;*
2. *augmenter les capacités d'un réseau de capteurs à l'aide de robots mobiles ;*
3. *faciliter l'apprentissage par les robots de comportements complexes.*

Grâce à la complémentarité des capteurs, il sera possible d'améliorer l'analyse de la situation courante et de l'état de l'environnement. L'apprentissage de comportements permettra quant à lui d'agir de manière plus adaptée aux besoins des utilisateurs. Ainsi, ces objectifs sont, pour moi, les premières étapes à l'amélioration générale de l'intelligence des systèmes robotiques à la fois dans la perception et dans l'action.

6.1.3 Approche

Mon approche s'intéresse à la fois au robot lui-même et à son environnement par le biais de l'exploitation de réseaux de capteurs. Dans les deux cas, un problème fondamental est celui du modèle de l'environnement : quelle que soit sa qualité, il ne pourra être complet. Une manière de gérer cette incomplétude est de la transformer en incertitude qui peut alors être représentée par des distributions de probabilités (LEBELTEL et al., 2004). C'est la base de l'approche bayésienne illustrée par les chapitres précédents et dont l'ap-

plication à la robotique s'est établie il y a plus d'une dizaine d'années (BESSIÈRE, LAUGIER et SIEGWART, 2008 ; COLAS, DIARD et BESSIÈRE, 2010).

Avec ce cadre théorique général, je compte orienter ma recherche concrètement en suivant deux axes. Le premier axe se rapporte à la première question scientifique et s'attache aux interactions entre un robot mobile et un réseau de capteurs. On y trouvera le principal des contributions aux objectifs 1 et 2. Le deuxième axe se rapporte à la deuxième question scientifique et s'attache à l'apprentissage automatique. On y trouvera l'essentiel des contributions à l'objectif 3.

6.2 Robotique située

La robotique située repose sur l'idée d'une interaction entre un environnement dit intelligent, c'est-à-dire en particulier incluant des capteurs et des unités de traitement, un ou plusieurs robots, là encore avec des capteurs, des unités de traitement et des actionneurs, et un ou plusieurs utilisateurs humains. Dans ce contexte, mon travail se structure selon le cycle de vie d'un tel système, depuis l'installation jusqu'à la maintenance en passant par l'utilisation.

6.2.1 Estimation

Un premier aspect de l'exploitation de capteurs divers est la fusion des données de caractéristiques différentes pour l'estimation d'un état cohérent du système. Ainsi un sol équipé de capteurs de pression comme celui de l'appartement intelligent, permet théoriquement de renseigner sur la position de robots ou autres objets mobiles. De même, des microphones distribués dans l'environnement donnent, sur des sources sonores, des informations telles que la position mais aussi le type de source et son activité. Enfin, un réseau de caméras permet d'obtenir des informations riches sur la scène visuelle : couleur, position, taille apparente, etc.

Ces différents capteurs se basent sur des processus physiques différents et mesurent des quantités de nature différente, à des fréquences d'échantillonnage différentes et dans des conditions différentes. Ainsi, les caméras mesurent les couleurs projetées sur une grille régulière de pixels à une fréquence de l'ordre de 30 Hz. Le sol instrumenté de l'appartement intelligent donne environ 400 mesures unidimensionnelles de force couvrant une surface d'environ 36 m² à une fréquence de 20 Hz. Enfin des systèmes de microphones peuvent typiquement indiquer la direction de la source sonore à une dizaine de Hertz.

Pour prendre en compte ces différents capteurs, il existe deux approches principales. La première consiste en des filtres de fusion forte qui définissent un état unique et des modèles d'observation prenant en compte conjointement l'ensemble des capteurs. Cette approche a une complexité assez importante (cubique) en fonction de la taille de l'espace d'état et de l'espace d'observation et a des difficultés à passer à l'échelle pour des observations complexes ou nombreuses. Ainsi elle est très souvent utilisée avec des étapes de prétraitement visant à réduire l'espace d'observation. Il s'agit, par exemple, le passage d'une image couleur, en grande dimension, à une liste de positions de cibles extraites de l'image.

La deuxième approche, la fusion faible, considère des filtres indépendants travaillant chacun sur un capteur couplé à un sous-espace de l'espace d'état. La complexité de chaque

filtre reste limitée et il suffit ensuite de fusionner les estimations sur les états. Le problème principal de cette approche est qu'elle ignore les dépendances entre les informations des différents capteurs et, par conséquent, mésestime l'incertitude globale sur l'état. D'autre part, les travaux de fusion faible travaillent tous avec une représentation homogène de l'état, typiquement des distributions gaussiennes ou discrétisées. Or, les caractéristiques des modalités différentes amènent souvent à des choix de représentation distincts et incompatibles. C'est le cas par exemple des travaux sur la localisation et la cartographie qui, actuellement, reposent en majorité sur des filtres à particules (GRISSETTI, STACHNISS et BURGARD, 2007) alors que la localisation de sources sonores utilise des mixtures de gaussiennes (NGUYEN et al., 2016) et que la localisation de robot à l'aide d'un sol intelligent peut se satisfaire d'un filtre de Kalman (RIO et al., 2016). Je souhaite donc travailler sur des structures de filtres hiérarchiques et modulaires qui permettent de combiner des espaces d'état et des représentations différentes en une estimation d'état persistante.

Enfin, un dernier défi sur cette partie serait l'intégration entre les techniques basées sur des modèles telles qu'exposées et mises en œuvre jusqu'à présent et des techniques sans modèles et basées sur l'apprentissage. Ce défi peut être scindé en deux parties distinctes. La première, que j'appelle en série, vise à essayer d'apprendre l'écart entre le modèle utilisé et les données réelles. Ainsi le modèle de bruit gaussien centré, à la base des calculs des filtres de Kalman et de ses dérivés, est soit une approximation d'un modèle de bruit plus complexe mais pour lequel on ne saurait pas faire les calculs, soit une hypothèse non vérifiée a priori et qui n'est validée à l'usage que par l'utilité du résultat. Mon approche ici est d'apprendre la régression entre l'innovation (écart entre l'observation et sa prédiction par le modèle) et l'état précédent en fonction de la commande. Une fois cette régression apprise, par exemple à l'aide de processus gaussiens, on devrait alors pouvoir la rétropropager à travers le modèle d'observation de manière à obtenir une meilleure estimation.

La seconde partie de ce défi consiste plutôt en l'augmentation de l'espace d'état par de nouvelles dimensions apprises de manière à réduire l'erreur de prédiction de l'observation. L'outil à mettre en œuvre serait plutôt des modèles d'apprentissage profonds tels que des réseaux récurrents au sein desquels serait intégré le modèle via ses prédictions comme une nouvelle entrée du réseau. Cet effort serait dans la lignée des travaux que nous avons débutés avec Emmanuel Vincent autour du stage de Mohammad Rami Koujan.

6.2.2 **Étalonnage**

Une autre problématique rencontrée par tout ensemble de capteurs est celui de l'étalonnage, en particulier par des utilisateurs non-experts. Concrètement, pour des caméras par exemple, il est nécessaire d'estimer les paramètres intrinsèques (distance focale, position du centre optique...) et extrinsèques (position et orientation relatives à un repère extérieur). Il existe de nombreuses méthodes d'étalonnage, mais qui reposent principalement sur la présentation d'un modèle connu (tel un échiquier) à la caméra (ABRAHAM et FÖRSTNER, 2005 ; REMONDINO et FRASER, 2006).

La première interaction entre un robot et son environnement que je propose est donc l'aide, par un robot mobile, à l'étalonnage d'un réseau de caméras. Cette application permet de mettre en avant plusieurs problématiques scientifiques distinctes. Dans un premier temps, on peut imaginer un utilisateur plaçant itérativement un robot équipé d'un échi-

quier dans le champ de vision de chaque caméra à son tour. Le premier problème est alors de planifier les mouvements du robot de manière à couvrir l'ensemble du champ de vision de la caméra. En l'absence de carte de l'environnement et pour des caméras grand angle utilisées dans ce genre d'application, une simple planification locale ne fonctionne plus. Il faut à la fois mettre en œuvre une cartographie et localisation du robot mobile et prendre en compte la position dans l'image, mais aussi la distance à la caméra. Une estimation en ligne des paramètres de la caméra est alors nécessaire pour commencer à projeter les contours de l'exploration de l'environnement par le robot, en prenant en compte à la fois la position dans l'image et la distance à la caméra.

Il est facile d'imaginer répéter cette procédure pour chaque caméra de l'environnement et d'en estimer les positions respectives vis-à-vis de la carte construite en parallèle par le robot mobile. Une deuxième approche serait d'utiliser l'intersection entre les champs de vision des caméras durant l'étalonnage. En introduisant les contraintes épipolaires dérivées de l'observation conjointe d'un même objet par plusieurs caméras, il est possible, par exemple par la méthode de l'ajustement de faisceaux, d'obtenir une estimation du maximum de vraisemblance des positions relatives des caméras. Là encore, se pose le problème de la planification de mouvements du robot de manière à pouvoir insérer les contraintes les plus variées possible pour réduire l'incertitude de l'estimation. De plus, il n'est pas immédiat d'intégrer cette estimation des poses des caméras avec celle de la première méthode.

Enfin, à mesure de la réduction de l'incertitude sur les paramètres d'étalonnage, il est possible de mieux estimer la pose du robot mobile. Cependant, des approches naïves du problème, consistant soit à coupler fortement les paramètres des caméras, la pose du robot et la carte de l'environnement, soit au contraire à découpler étalonnage et cartographie amèneraient respectivement à une explosion de la complexité, ou à l'impossibilité de mettre en relation le résultat d'une approche de relaxation globale avec celui d'un processus incrémental. À l'image de la Rao-Blackwellisation qui a permis le développement de solutions efficaces dans le domaine de la cartographie et localisation simultanée (MONTEMERLO et al., 2002), il faut donc réussir à intégrer le minimum d'information sur les caméras permettant d'estimer l'état conjoint du système. Ces exemples, basés sur un réseau de caméras, illustrent concrètement comment je compte avancer l'état de l'art de la robotique située, en m'appuyant notamment sur la complémentarité entre un robot mobile et des capteurs fixes couvrant un espace donné.

6.2.3 Planification et observation

Un des objectifs à l'ajout de capteurs dans un environnement est d'y faciliter l'intégration de robots. Dans un environnement encombré, il arrive souvent que l'on ne puisse obtenir des observations fiables qu'à courte portée. Dans un environnement statique, la carte construite lors de l'exploration de l'environnement est suffisante pour la planification des actions du robot (LATOMBE, 1991 ; LAVALLE, 2006) mais ce n'est plus le cas en environnement dynamique où il peut se trouver un nouvel obstacle forçant une replanification tardive, comme une porte nouvellement fermée.

Des capteurs dans l'environnement peuvent alors pallier ce problème en fournissant des observations plus globales depuis des points de vue multiples et ainsi renseigner la planification le plus tôt possible. Le problème devient alors d'intégrer des informations

potentiellement de nature différente (image pour un réseau de caméras et capteur de distance pour le robot, par exemple) à la carte de l'environnement. On peut envisager d'essayer de projeter les observations des capteurs directement dans la carte avec le risque de la modifier de manière erronée. Je propose plutôt d'essayer d'apprendre des modèles, pour chaque endroit et chaque caméra, de l'environnement lorsqu'il est observé comme libre par le robot. Ensuite il devient possible de calculer la probabilité que chaque endroit soit encore libre, à partir des observations courantes des caméras. La planification globale du robot peut donc être facilitée par l'environnement.

D'un autre côté, les capteurs installés dans l'environnement peuvent également avoir un intérêt en eux-mêmes. C'est déjà le cas des caméras de surveillance mais aussi de réseaux de capteurs pour l'évaluation environnementale. Dans ces deux cas, il est possible de bénéficier de la complémentarité entre un réseau de capteurs fixes donnant une information à une échelle grossière et des plateformes mobiles permettant une investigation adaptative. En outre, un robot peut aider à combler les angles morts des champs de vision des caméras, qu'ils soient permanents en raison du placement initial, ou temporaires dans le cas d'une mise hors service ou d'une occlusion.

Le problème est alors de décider des points de vue où placer le robot mobile. On peut envisager de spécifier directement les patrouilles à suivre, mais il est plus intéressant d'estimer l'incertitude sur l'état global de l'environnement et de planifier les mouvements du robot de manière à réduire au mieux cette incertitude. De cette manière, on peut obtenir un comportement adaptatif du robot avec un minimum de spécification.

Cette application peut aussi trouver son intérêt avec des modalités différentes de capteurs, par exemple des microphones à la place de caméras fixes dans des lieux où de telles caméras seraient mal acceptées. Enfin, pour réduire les coûts de l'infrastructure, il est possible de baisser la qualité des capteurs de l'environnement et d'utiliser la plateforme mobile en complément. Ainsi, l'objectif est de pouvoir tirer parti de tous les types de complémentarité entre les capteurs fixes de l'environnement et ceux des robots mobiles : champ de vision, modalité, précision...

6.2.4 Maintenance

Enfin, le déploiement d'un système robotique situé pour des utilisateurs non experts pose de nombreuses contraintes pour son usage à long terme. Ainsi, contrairement à la pratique courante d'une simple démonstration ou d'un usage ponctuel, l'utilisation continue sur une longue durée pose la question de l'évolution des paramètres de configuration du système (étalonnage et carte en particulier). En effet, la majeure partie de la recherche en robotique résout des problèmes sur des échelles de temps assez courtes, au mieux de quelques heures. Estimer des paramètres ou même une carte en le moins de temps possible a un intérêt certain puisque cela implique une plus grande rapidité de déploiement dans de nouveaux environnements.

Cependant, dans le cadre de la robotique située et d'un environnement intelligent, il n'est pas nécessaire de refaire l'étalonnage ou la cartographie à chaque démarrage du système puisque l'environnement est largement inchangé depuis la dernière session. À l'opposé, il est important de ne pas totalement figer les paramètres du système ou la carte puisqu'il est toujours possible qu'ils changent. Ainsi, un meuble ou même un capteur peut avoir été déplacé, que ce soit volontairement ou non ; ou encore, certaines propriétés

peuvent changer à cause de l'usure : c'est le cas par exemple du diamètre des roues d'un robot.

Face à ce problème il y a plusieurs approches. On peut, en premier lieu, demander à l'utilisateur à l'origine d'un changement de configuration de relancer une procédure d'étalonnage. C'est ce qui est fait le plus souvent, par exemple pour les salles de *motion capture*, mais cela requiert l'intervention de l'utilisateur. Une deuxième approche consiste en ne pas cesser d'estimer tous les paramètres. Ainsi, en utilisant un algorithme d'estimation en ligne tel un filtre de Kalman pour l'étalonnage, il est possible de simplement ne pas arrêter l'estimation de paramètres ou même la cartographie.

En pratique cependant, cette approche peut s'avérer coûteuse en temps de calcul et en mémoire mais aussi se heurter à des observations aberrantes susceptibles de dégrader sévèrement la précision et la performance du système, du moins de manière temporaire. Il y a donc besoin d'une approche différente, soit en modifiant l'estimation pour être à la fois robuste et légère, soit en développant des algorithmes de détection d'erreur et de changement d'étalonnage pour déclencher un processus de correction.

En résumé de cette première partie sur la robotique, je souhaite développer et mettre en œuvre de nouveaux algorithmes qui permettent d'améliorer le déploiement, l'utilisation et la maintenance de systèmes de robotique située, comprenant à la fois une infrastructure de capteurs et des robots mobiles. Le point clé de ces innovations est la prise en compte des complémentarités entre les capteurs fixes et les plateformes mobiles afin d'augmenter leurs capacités respectives (objectifs 1 et 2).

6.3 Apprentissage automatique

Mon deuxième axe de recherche est plus fondamental que le premier dans le sens où il s'intéresse à des bases plus théoriques et des applications plus larges. En effet l'apprentissage automatique, dont le but est d'améliorer les performances d'un algorithme en prenant appui sur des données d'expérience, est utilisé pour de nombreuses applications comme la reconnaissance automatique de caractères, les moteurs de recherche ou la détection de courriers indésirables. Mon but dans ce domaine est d'utiliser l'apprentissage automatique pour rapprocher la robotique des utilisateurs non-experts.

6.3.1 Apprentissage par démonstration de comportements

Il existe plusieurs formes d'apprentissage automatique : non supervisé, supervisé, par renforcement, etc. Je me propose de travailler en premier lieu sur l'apprentissage par démonstration (ATKESON et SCHAAL, 1997 ; ARGALL et al., 2009). L'idée est de montrer à un système ce qu'il doit faire et ensuite de lui demander de le répéter. Par exemple pour un bras robotique, on peut le conduire à un endroit précis pendant que le système enregistre toutes les données pertinentes, comme les valeurs des encodeurs ou les valeurs d'actionnement. Si l'on ne fait que rejouer la séquence des commandes motrices, le système ne va pas pouvoir aller au même endroit s'il ne repart pas exactement du même état initial. Même dans ce cas, les incertitudes dans la chaîne d'actionnement ont de bonnes chances de ne pas l'amener au point voulu avec la précision souhaitée. Enfin, si une perturbation

intervient ou s'il faut éviter un obstacle, il n'est plus possible de corriger le mouvement. C'est pour cela qu'il faut développer des algorithmes à même de généraliser à partir de plusieurs démonstrations. De tels algorithmes existent déjà pour apprendre un mouvement particulier tel que celui d'un bras robotique dans un espace libre. L'idée générale est souvent de construire un modèle synthétique du mouvement à partir des données considérées comme bruitées. C'est un processus complexe qui dépend de la tâche, du système et qui exige souvent de régler de nombreux paramètres même pour apprendre un mouvement continu.

Cependant, de nombreuses applications robotiques nécessitent des séquences de mouvements à réaliser et coordonner à partir de données capteurs hétérogènes. C'est le cas par exemple d'applications de navigation s'appuyant à la fois sur des données de distance pour éviter les obstacles et sur des données de caméra afin de choisir la destination. Pour cela, je propose un algorithme bayésien de suivi d'exécution parmi un ensemble de trajectoires enregistrées (MAGNENAT, PRADALIER et COLAS, 2012). Cet algorithme ne construit pas de modèle de la tâche et se base directement sur les données d'apprentissage. L'objectif est de nécessiter le moins de paramètres possibles pour pouvoir s'adapter rapidement à une grande variété de tâches et de capteurs. Je me propose de faire un travail de caractérisation de l'algorithme proposé en vue d'évaluer ses capacités de généralisation, le nombre de trajectoires nécessaires à l'apprentissage, la précision des trajectoires reproduites, la robustesse vis-à-vis de trajectoires liées à d'autres tâches, etc.

Notre algorithme requiert actuellement de stocker et de parcourir toutes les données des capteurs et des actionneurs de toutes les trajectoires d'apprentissage. Pour des systèmes et des tâches complexes, cette quantité de données peut devenir difficile à gérer. Je prévois donc de proposer une méthode de réduction de la quantité de données en identifiant celles qui n'apportent pas d'information supplémentaire par rapport à celles déjà existantes. Cette identification peut s'appuyer sur le même algorithme de suivi dans l'espace des trajectoires grâce à l'observation des coactivations.

Ces avancées dans l'apprentissage par démonstration permettront de faciliter la programmation des robots par des utilisateurs non-spécialistes. En outre, l'implication de l'utilisateur dans le processus d'apprentissage devrait améliorer la compréhension des limites du robot d'une part, et aussi leur acceptation dans la sphère privée.

6.3.2 Interactions avec les utilisateurs

En plus d'impliquer l'utilisateur dans le processus d'apprentissage du robot, la robotique située peut, en parallèle, apprendre certaines habitudes de l'utilisateur dans le but de l'aider dans sa vie quotidienne. Il est ainsi déjà possible de se procurer des thermostats qui apprennent les consignes de température des différentes pièces d'une maison au cours du temps. Cela induit par exemple des économies d'énergie en ne faisant pas fonctionner le chauffage ou l'air conditionné durant la journée si personne n'est là, tout en permettant un niveau de confort suffisant en remettant les pièces à température pour l'heure d'arrivée habituelle des habitants.

De la même façon, on peut aussi développer des algorithmes de planification, de mouvements ou de tâches prenant en compte l'habitude des utilisateurs. En effet si, à un moment particulier de la journée, les personnes ont tendance à se déplacer d'une pièce à l'autre, il peut être intéressant pour le robot de choisir un chemin ayant moins de chances

d'être emprunté de manière à limiter la probabilité d'être bloqué mais surtout de gêner les habitants. De même, il est préférable de planifier une tâche telle que le ménage d'une pièce lorsqu'elle est le moins susceptible d'être utilisée.

De manière à apprendre ces modèles, il est nécessaire de recourir non seulement aux capteurs du robot, mais aussi au réseau de capteurs de l'environnement intelligent. Malgré les avancées importantes en vision par ordinateur, il n'est pas encore possible d'obtenir une détection parfaite des humains surtout si l'on prend en compte la variété d'activités et de positions à laquelle ils peuvent se livrer. Ainsi il faut là encore que les modèles d'apprentissage mis en œuvre soient robustes aux fausses alertes et aux échecs de détection.

Une autre piste est d'améliorer les performances en fusionnant différentes modalités sensorielles du réseau de capteurs et en prenant en compte à la fois la dimension temporelle et la dimension spatiale de la détection. Les détecteurs visuels sont souvent composés de cascades de détecteurs plus simples mais moins sélectifs. C'est le choix judicieux du sous-ensemble de détecteurs, réalisé à partir des données d'apprentissage souvent libellées manuellement, qui permet d'obtenir de bonnes performances. L'ajout de données issues de modalités différentes, telles que des capteurs de pression au sol, devrait permettre, de par leur complémentarité, à la fois d'augmenter la performance et de réduire la complexité. Enfin, un robot mobile peut aussi être utilisé pour obtenir un point de vue différent de manière à lever des ambiguïtés.

À plus long terme, ces modèles de présence et d'utilisation de l'habitat pourront être utilisés pour détecter des situations anormales, comme une intrusion ou la chute d'une personne âgée. Dans une application de suivi de personnes médicalisées à domicile, le même type de modèles peut être mis en œuvre à partir de mesures biologiques pour, là encore, lancer une alerte en cas de détection d'anomalie.

6.4 Conclusion

En conclusion, mon projet de recherche s'intéresse conjointement à la robotique située et à l'apprentissage automatique dans le cadre de la robotique de service, en particulier pour l'assistance à la personne. Je me propose de replacer les robots au sein d'un triptyque incluant en outre un environnement équipé de capteurs ainsi que les utilisateurs. En tirant parti des complémentarités au sein de cet ensemble, je compte atteindre à la fois une plus grande autonomie et une plus grande utilité. Pour cela, l'apprentissage de comportements et, plus généralement, de modèles est une pierre angulaire dans le déploiement et l'adaptabilité d'un tel système.

Au-delà de ces contributions, mon objectif est d'améliorer l'intelligence de systèmes robotiques, c'est-à-dire leur capacité à mobiliser leur expérience pour percevoir la situation courante et agir de manière appropriée. Je pense cela possible en développant notamment des mécanismes d'abstractions issus de l'apprentissage par démonstration.

Enfin, mon expérience m'a montré la nécessité, à terme, de mettre en rapport des spécialistes des interfaces homme-machine, des praticiens en contact avec le public ciblé et des industriels à même de produire ces systèmes. C'est par une expérimentation mettant en jeu des utilisateurs finaux que l'on pourra contribuer précisément à des solutions qui puissent aboutir à un impact dans la société.

Bibliographie

- Abraham, Steffen and Wolfgang Förstner (2005). “Fish-eye-stereo calibration and epipolar rectification”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 59.5, pp. 278–288. DOI: [10.1016/j.isprsjprs.2005.03.001](https://doi.org/10.1016/j.isprsjprs.2005.03.001).
- Almeida, Jorge and Vitor M. Santos (2013). “Real time egomotion of a nonholonomic vehicle using LIDAR measurements”. In: *Journal of Field Robotics* 30.1, pp. 129–141. DOI: [10.1002/rob.21441](https://doi.org/10.1002/rob.21441).
- Almeida, Jorge, Carlos Silvestre, and António Pascoal (2010). “Cooperative control of multiple surface vessels in the presence of ocean currents and parametric model uncertainty”. In: *International Journal of Robust and Nonlinear Control* 20.14, pp. 1549–1565. DOI: [10.1002/rnc.1526](https://doi.org/10.1002/rnc.1526).
- Alvarez, George A. and Patrick Cavanagh (2005). “Independent Resources for Attentional Tracking in the Left and Right Visual Hemifields”. In: *Psychological Science* 16.8, pp. 637–643. DOI: [10.1111/j.1467-9280.2005.01587.x](https://doi.org/10.1111/j.1467-9280.2005.01587.x).
- Alvarez, George A. and Steven L. Franconeri (2007). “How many objects can you track? Evidence for a resource-limited attentive tracking mechanism”. In: *Journal of Vision* 7.13, pp. 14.1–10. DOI: [10.1167/7.13.14](https://doi.org/10.1167/7.13.14).
- Amigoni, Francesco, Monica Reggiani, and Viola Schiaffonati (2009). “An insightful comparison between experiments in mobile robotics and in science”. In: *Autonomous Robots* 27.4, pp. 313–325. DOI: [10.1007/s10514-009-9137-8](https://doi.org/10.1007/s10514-009-9137-8).
- Argall, Brenna D., Sonia Chernova, Manuela Veloso, and Brett Browning (2009). “A survey of robot learning from demonstration”. In: *Robotics and Autonomous Systems* 57.5, pp. 469–483. DOI: [10.1016/j.robot.2008.10.024](https://doi.org/10.1016/j.robot.2008.10.024).
- Armesto, Leopoldo, Javier Minguez, and Luis Montesano (2010). “A generalization of the metric-based Iterative Closest Point technique for 3D scan matching”. In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1367–1372. DOI: [10.1109/ROBOT.2010.5509371](https://doi.org/10.1109/ROBOT.2010.5509371).
- Arun, K.S., Thomas S. Huang, and Steven D. Blostein (1987). “Least-Squares Fitting of Two 3-D Point Sets”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.5, pp. 698–700. DOI: [10.1109/TPAMI.1987.4767965](https://doi.org/10.1109/TPAMI.1987.4767965).
- Arya, Sunil and David M. Mount (1993). “Approximate Nearest Neighbor Queries in Fixed Dimensions.” In: pp. 271–280.
- Atkeson, Christopher G. and Stefan Schaal (1997). “Robot Learning From Demonstration”. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. ICML '97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 12–20.
- Bachrach, Abraham, Samuel Prentice, Ruijie He, and Nicholas Roy (2011). “RANGE-Robust autonomous navigation in GPS-denied environments”. In: *Journal of Field Robotics* 28.5, pp. 644–666. DOI: [10.1002/rob.20400](https://doi.org/10.1002/rob.20400).

- Baker, Simon, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski (2010). “A Database and Evaluation Methodology for Optical Flow”. In: *International Journal of Computer Vision* 92.1, pp. 1–31. DOI: [10.1007/s11263-010-0390-2](https://doi.org/10.1007/s11263-010-0390-2).
- Barash, Shabtai, R. Martyn Bracewell, Leonardo Fogassi, James W. Gnadt, and Richard A. Andersen (1991a). “Saccade-related activity in the lateral intraparietal area. I. Temporal properties; comparison with area 7a”. In: *Journal of Neurophysiology* 66.3, pp. 1095–1108.
- (1991b). “Saccade-related activity in the lateral intraparietal area. II. Spatial properties”. In: *Journal of Neurophysiology* 66.3, pp. 1109–1124.
- Ben Hamed, Suliann, Jean-René Duhamel, Frank Bremmer, and Werner Graf (2001). “Representation of the visual field in the lateral intraparietal area of macaque monkeys: a quantitative receptive field analysis”. In: *Experimental Brain Research* 140.2, pp. 127–144. DOI: [10.1007/s002210100785](https://doi.org/10.1007/s002210100785).
- Ben-Kiki, Oren, Clark Evans, and Ingy döt Net (2009). *YAML Ain’t Markup Language (YAML™) version 1.2*.
- Besl, Paul J. and Neil D. McKay (1992). “A method for registration of 3-D shapes”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 14.2, pp. 239–256. DOI: [10.1109/34.121791](https://doi.org/10.1109/34.121791).
- Bessière, Pierre, Christian Laugier, and Roland Siegwart (2008). *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*. Springer Science & Business Media.
- Bessière, Pierre, Emmanuel Mazer, Juan Manuel Ahuactzin, and Kamel Mekhnacha (2013). *Bayesian Programming*. CRC Press.
- Bibuli, Marco, Gabriele Bruzzone, Massimo Caccia, and Lionel Lapierre (2009). “Path-following algorithms and experiments for an unmanned surface vehicle”. In: *Journal of Field Robotics* 26.8, pp. 669–688. DOI: [10.1002/rob.20303](https://doi.org/10.1002/rob.20303).
- Bortoff, Scott A. (2000). “Path planning for UAVs”. In: *American Control Conference, 2000. Proceedings of the 2000*. Vol. 1, 364–368 vol.1. DOI: [10.1109/ACC.2000.878915](https://doi.org/10.1109/ACC.2000.878915).
- Bosse, Michael and Robert Zlot (2008). “Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM”. In: *The International Journal of Robotics Research* 27.6, pp. 667–691. DOI: [10.1177/0278364908091366](https://doi.org/10.1177/0278364908091366).
- (2009a). “Continuous 3D scan-matching with a spinning 2D laser”. In: *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, pp. 4312–4319. DOI: [10.1109/ROBOT.2009.5152851](https://doi.org/10.1109/ROBOT.2009.5152851).
- (2009b). “Keypoint design and evaluation for place recognition in 2D lidar maps”. In: *Robotics and Autonomous Systems*. Inside Data Association 57.12, pp. 1211–1224. DOI: [10.1016/j.robot.2009.07.009](https://doi.org/10.1016/j.robot.2009.07.009).
- Boxtel, Jeroen J. A. van, Mark Wexler, and Jacques Droulez (2003). “Perception of plane orientation from self-generated and passively observed optic flow”. In: *Journal of Vision* 3.5, p. 1.
- Bozsis, A. and Adonis K. Moschovakis (1998). “Neural network simulations of the primate oculomotor system III. An one-dimensional, one-directional model of the superior colliculus”. In: *Biological Cybernetics* 79.3, pp. 215–230. DOI: [10.1007/s004220050472](https://doi.org/10.1007/s004220050472).
- Broekens, Joošt, Marcel Heerink, and Henk Rosendal (2009). “Assistive social robots in elderly care: a review”. In: *Gerontechnology* 8.2. DOI: [10.4017/gt.2009.08.02.002.00](https://doi.org/10.4017/gt.2009.08.02.002.00).

- Camarinha-Matos, Luis M. and Walter Vieira (1999). "Intelligent mobile agents in elderly care". In: *Robotics and Autonomous Systems*. Multi-Agent Systems Applications 27.1-2, pp. 59-75. DOI: [10.1016/S0921-8890\(98\)00083-9](https://doi.org/10.1016/S0921-8890(98)00083-9).
- Campbell, Mark, Magnus Egerstedt, Jonathan P. How, and Richard M. Murray (2010). "Autonomous driving in urban environments: approaches, lessons and challenges". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 368.1928, pp. 4649-4672. DOI: [10.1098/rsta.2010.0110](https://doi.org/10.1098/rsta.2010.0110).
- Champleboux, Guillaume, Stéphane Lavallée, Richard Szeliski, and Lionel Brunie (1992). "From accurate range imaging sensor calibration to accurate model-based 3D object localization". In: , *1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92*, pp. 83-89. DOI: [10.1109/CVPR.1992.223223](https://doi.org/10.1109/CVPR.1992.223223).
- Chen, Yang and Gérard Medioni (1991). "Object modeling by registration of multiple range images". In: , *1991 IEEE International Conference on Robotics and Automation, 1991. Proceedings*, 2724-2729 vol.3. DOI: [10.1109/ROBOT.1991.132043](https://doi.org/10.1109/ROBOT.1991.132043).
- Chetverikov, Dmitry, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek (2002). "The Trimmed Iterative Closest Point algorithm". In: *16th International Conference on Pattern Recognition, 2002. Proceedings*. Vol. 3, 545-548 vol.3. DOI: [10.1109/ICPR.2002.1047997](https://doi.org/10.1109/ICPR.2002.1047997).
- Colas, Francis (2006). "Perception des objets en mouvement. Composition bayésienne du flux optique et du mouvement de l'observateur". fr. PhD thesis. Institut National Polytechnique de Grenoble - INPG.
- Colas, Francis, Julien Diard, and Pierre Bessière (2010). "Common Bayesian Models for Common Cognitive Issues". In: *Acta Biotheoretica* 58.2-3, pp. 191-216. DOI: [10.1007/s10441-010-9101-1](https://doi.org/10.1007/s10441-010-9101-1).
- Colas, Francis, Jacques Droulez, Mark Wexler, and Pierre Bessière (2007). "A unified probabilistic model of the perception of three-dimensional structure from optic flow". In: *Biological Cybernetics* 97.5-6, pp. 461-477. DOI: [10.1007/s00422-007-0183-z](https://doi.org/10.1007/s00422-007-0183-z).
- Colas, Francis, Fabien Flacher, Thomas Tanner, Pierre Bessière, and Benoît Girard (2009). "Bayesian models of eye movement selection with retinotopic maps". In: *Biological Cybernetics* 100.3, pp. 203-214. DOI: [10.1007/s00422-009-0292-y](https://doi.org/10.1007/s00422-009-0292-y).
- Colas, Francis, Srivatsa Mahesh, François Pomerleau, Ming Liu, and Roland Siegwart (2013). "3D path planning and execution for search and rescue ground robots". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 722-727. DOI: [10.1109/IROS.2013.6696431](https://doi.org/10.1109/IROS.2013.6696431).
- Cornilleau-Pérès, Valérie, Mark Wexler, Jacques Droulez, Emmanuel Marin, Christian Miège, and Bernard Bourdoncle (2002). "Visual perception of planar orientation: dominance of static depth cues over motion cues". In: *Vision Research* 42.11, pp. 1403-1412. DOI: [10.1016/S0042-6989\(01\)00298-X](https://doi.org/10.1016/S0042-6989(01)00298-X).
- Dhariwal, Amit and Gaurav S. Sukhatme (2007). "Experiments in robotic boat localization". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, pp. 1702-1708. DOI: [10.1109/IROS.2007.4399056](https://doi.org/10.1109/IROS.2007.4399056).
- Diebel, James, Kjell Reuterswärd, Sebastian Thrun, James Davis, and Rakesh Gupta (2004). "Simultaneous localization and mapping with active stereo vision". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings*. Vol. 4, 3436-3443 vol.4. DOI: [10.1109/IROS.2004.1389948](https://doi.org/10.1109/IROS.2004.1389948).

- Dijkstra, Edsger W. (1959). “A note on two problems in connexion with graphs”. In: *Numerische Mathematik* 1.1, pp. 269–271. DOI: [10.1007/BF01386390](https://doi.org/10.1007/BF01386390).
- Dijkstra, Tjeerd M. H., Valérie Cornilleau-Pérès, Constantinus C. A. M. Gielen, and Jacques Droulez (1995). “Perception of three-dimensional shape from ego- and object-motion: Comparison between small- and large-field stimuli”. In: *Vision Research* 35.4, pp. 453–462. DOI: [10.1016/0042-6989\(94\)00147-E](https://doi.org/10.1016/0042-6989(94)00147-E).
- Domini, Fulvio and Myron L. Braunstein (1998). “Recovery of 3-D structure from motion is neither euclidean nor affine”. In: *Journal of Experimental Psychology: Human Perception and Performance* 24.4, pp. 1273–1295. DOI: [10.1037/0096-1523.24.4.1273](https://doi.org/10.1037/0096-1523.24.4.1273).
- Domini, Fulvio and Corrado Caudek (1999). “Perceiving surface slant from deformation of optic flow”. In: *Journal of Experimental Psychology: Human Perception and Performance* 25.2, pp. 426–444. DOI: [10.1037/0096-1523.25.2.426](https://doi.org/10.1037/0096-1523.25.2.426).
- (2003). “3-D structure perceived from dynamic information: a new theory”. In: *Trends in Cognitive Sciences* 7.10, pp. 444–449. DOI: [10.1016/j.tics.2003.08.007](https://doi.org/10.1016/j.tics.2003.08.007).
- Droulez, Jacques and Alain Berthoz (1991). “A neural network model of sensoritopic maps with predictive short-term memory properties.” In: *Proceedings of the National Academy of Sciences* 88.21, pp. 9653–9657. DOI: [10.1073/pnas.88.21.9653](https://doi.org/10.1073/pnas.88.21.9653).
- Druon, Sébastien, Marie-José Aldon, and André Crosnier (2006). “Color Constrained ICP for Registration of Large Unstructured 3D Color Data Sets”. In: *2006 IEEE International Conference on Information Acquisition*, pp. 249–255. DOI: [10.1109/ICIA.2006.306004](https://doi.org/10.1109/ICIA.2006.306004).
- Elfes, Alberto (1989). “Using occupancy grids for mobile robot perception and navigation”. In: *Computer* 22.6, pp. 46–57. DOI: [10.1109/2.30720](https://doi.org/10.1109/2.30720).
- Elseberg, Jan, Stéphane Magnenat, Roland Siegwart, and Andreas Nüchter (2012). “Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration”. In: *Journal of Software Engineering for Robotics* 3.1, pp. 2–12.
- Endo, Daisuke, Yoshito Okada, Keiji Nagatani, and Kazuya Yoshida (2007). “Path following control for tracked vehicles based on slip-compensating odometry”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, pp. 2871–2876. DOI: [10.1109/IROS.2007.4399228](https://doi.org/10.1109/IROS.2007.4399228).
- Ernst, Marc O. and Martin S. Banks (2002). “Humans integrate visual and haptic information in a statistically optimal fashion”. In: *Nature* 415.6870, pp. 429–433. DOI: [10.1038/415429a](https://doi.org/10.1038/415429a).
- Fairfield, Nathaniel and David Wettergreen (2009). “Evidence grid-based methods for 3D map matching”. In: *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, pp. 1637–1642. DOI: [10.1109/ROBOT.2009.5152688](https://doi.org/10.1109/ROBOT.2009.5152688).
- Feemster, Matthew G. and Joel M. Esposito (2011). “Comprehensive framework for tracking control and thrust allocation for a highly overactuated autonomous surface vessel”. In: *Journal of Field Robotics* 28.1, pp. 80–100. DOI: [10.1002/rob.20369](https://doi.org/10.1002/rob.20369).
- Feldmar, Jacques and Nicholas Ayache (1996). “Rigid, affine and locally affine registration of free-form surfaces”. In: *International Journal of Computer Vision* 18.2, pp. 99–119. DOI: [10.1007/BF00054998](https://doi.org/10.1007/BF00054998).
- Ferguson, Dave and Anthony Stentz (2007). “Field D*: An Interpolation-Based Path Planner and Replanner”. In: *Robotics Research*. Ed. by Dr Sebastian Thrun, Dr Rodney Brooks, and Dr Hugh Durrant-Whyte. Springer Tracts in Advanced Robotics 28. Springer Berlin Heidelberg, pp. 239–253.

- Fossen, Thor I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons.
- Fraundorfer, Friedrich and Davide Scaramuzza (2012). “Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications”. In: *IEEE Robotics Automation Magazine* 19.2, pp. 78–90. DOI: [10.1109/MRA.2012.2182810](https://doi.org/10.1109/MRA.2012.2182810).
- Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). “Are we ready for autonomous driving? The KITTI vision benchmark suite”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361. DOI: [10.1109/CVPR.2012.6248074](https://doi.org/10.1109/CVPR.2012.6248074).
- Gnadt, James. W. and Richard A. Andersen (1988). “Memory related motor planning activity in posterior parietal cortex of macaque”. In: *Experimental Brain Research* 70.1, pp. 216–220. DOI: [10.1007/BF00271862](https://doi.org/10.1007/BF00271862).
- Godin, Guy, Marc Rioux, and Rejean Baribeau (1994). “Three-dimensional registration using range and intensity information”. In: vol. 2350, pp. 279–290. DOI: [10.1117/12.189139](https://doi.org/10.1117/12.189139).
- Goldberg, Michael E. and Charles. J. Bruce (1990). “Primate frontal eye fields. III. Maintenance of a spatially accurate saccade signal”. In: *Journal of Neurophysiology* 64.2, pp. 489–508.
- Gossow, David, Adam Leeper, Dave Hershberger, and Matei Ciocarlie (2011). “Interactive Markers: 3-D User Interfaces for ROS Applications [ROS Topics]”. In: *IEEE Robotics Automation Magazine* 18.4, pp. 14–15. DOI: [10.1109/MRA.2011.943230](https://doi.org/10.1109/MRA.2011.943230).
- Grisetti, Giorgio, Cyrill Stachniss, and Wolfram Burgard (2005). “Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling”. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, pp. 2432–2437. DOI: [10.1109/ROBOT.2005.1570477](https://doi.org/10.1109/ROBOT.2005.1570477).
- (2007). “Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters”. In: *IEEE Transactions on Robotics* 23.1, pp. 34–46. DOI: [10.1109/TRO.2006.889486](https://doi.org/10.1109/TRO.2006.889486).
- Hansen, Nikolaus (2006). “The CMA Evolution Strategy: A Comparing Review”. In: *Towards a New Evolutionary Computation*. Ed. by Jose A. Lozano, Pedro Larrañaga, Iñaki Inza, and Endika Bengoetxea. Studies in Fuzziness and Soft Computing 192. DOI: [10.1007/3-540-32494-1_4](https://doi.org/10.1007/3-540-32494-1_4). Springer Berlin Heidelberg, pp. 75–102.
- Hart, Peter E., Nils J. Nilsson, and Bertram Raphael (1968). “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4.2, pp. 100–107. DOI: [10.1109/TSSC.1968.300136](https://doi.org/10.1109/TSSC.1968.300136).
- He, Ruijie, Sam Prentice, and Nicholas Roy (2008). “Planning in information space for a quadrotor helicopter in a GPS-denied environment”. In: *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, pp. 1814–1820. DOI: [10.1109/ROBOT.2008.4543471](https://doi.org/10.1109/ROBOT.2008.4543471).
- Helmholtz, Hermann von (1867). *Handbuch der physiologischen Optik*. de. Voss.
- Henry, Peter, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox (2012). “RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments”. In: *The International Journal of Robotics Research* 31.5, pp. 647–663. DOI: [10.1177/0278364911434148](https://doi.org/10.1177/0278364911434148).

- Herrero, Luis, Fernando Rodríguez, Cosme Salas, and Blas Torres (1998). “Tail and eye movements evoked by electrical microstimulation of the optic tectum in goldfish”. In: *Experimental Brain Research* 120.3, pp. 291–305. DOI: [10.1007/s002210050403](https://doi.org/10.1007/s002210050403).
- Hitz, Gregory, François Pomerleau, Francis Colas, and Roland Siegwart (2015). “Relaxing the planar assumption: 3D state estimation for an autonomous surface vessel”. In: *The International Journal of Robotics Research* 34.13, pp. 1604–1621. DOI: [10.1177/0278364915583680](https://doi.org/10.1177/0278364915583680).
- Horn, Berthold K. P. (1987). “Closed-form solution of absolute orientation using unit quaternions”. In: *Journal of the Optical Society of America A* 4.4, p. 629. DOI: [10.1364/JOSAA.4.000629](https://doi.org/10.1364/JOSAA.4.000629).
- Horn, Berthold K. P., Hugh M. Hilden, and Shahriar Negahdaripour (1988). “Closed-form solution of absolute orientation using orthonormal matrices”. In: *Journal of the Optical Society of America A* 5.7, p. 1127. DOI: [10.1364/JOSAA.5.001127](https://doi.org/10.1364/JOSAA.5.001127).
- Hornung, Armin, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard (2013). “OctoMap: an efficient probabilistic 3D mapping framework based on octrees”. In: *Autonomous Robots* 34.3, pp. 189–206. DOI: [10.1007/s10514-012-9321-0](https://doi.org/10.1007/s10514-012-9321-0).
- Huynh, Du Q. (2009). “Metrics for 3D Rotations: Comparison and Analysis”. In: *Journal of Mathematical Imaging and Vision* 35.2, pp. 155–164. DOI: [10.1007/s10851-009-0161-2](https://doi.org/10.1007/s10851-009-0161-2).
- IFR (2015). *World Robotics 2015 – Executive summary*. Executive Summary. International Federation of Robotics.
- Jošt, Timothée and Heinz Hügli (2002). “A multi-resolution scheme ICP algorithm for fast shape registration”. In: *First International Symposium on 3D Data Processing Visualization and Transmission, 2002. Proceedings*, pp. 540–543. DOI: [10.1109/TDPVT.2002.1024114](https://doi.org/10.1109/TDPVT.2002.1024114).
- (2003). “A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images”. In: *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings*, pp. 427–433. DOI: [10.1109/IM.2003.1240278](https://doi.org/10.1109/IM.2003.1240278).
- Karaman, Sertac and Emilio Frazzoli (2011). “Sampling-based algorithms for optimal motion planning”. In: *The International Journal of Robotics Research* 30.7, pp. 846–894. DOI: [10.1177/0278364911406761](https://doi.org/10.1177/0278364911406761).
- Kavraki, Lydia E., Peter Svestka, Jean-Claude Latombe, and Mark H. Overmars (1996). “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”. In: *IEEE Transactions on Robotics and Automation* 12.4, pp. 566–580. DOI: [10.1109/70.508439](https://doi.org/10.1109/70.508439).
- Kersten, Daniel, Pascal Mamassian, and Alan Yuille (2004). “Object Perception as Bayesian Inference”. In: *Annual Review of Psychology* 55.1, pp. 271–304. DOI: [10.1146/annurev.psych.55.090902.142005](https://doi.org/10.1146/annurev.psych.55.090902.142005).
- Koenderink, Jan J. (1986). “Optic flow”. In: *Vision Research* 26.1, pp. 161–179. DOI: [10.1016/0042-6989\(86\)90078-7](https://doi.org/10.1016/0042-6989(86)90078-7).
- Koenig, Sven and Maxim Likhachev (2002). “D*Lite”. In: *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, pp. 476–483.
- Kohlbrecher, Stefan, Oskar von Stryk, Johannes Meyer, and Uwe Klingauf (2011). “A flexible and scalable SLAM system with full 3D motion estimation”. In: *2011 IEEE Inter-*

- national Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 155–160. DOI: [10.1109/SSRR.2011.6106777](https://doi.org/10.1109/SSRR.2011.6106777).
- Konolige, Kurt, Motilal Agrawal, and Joan Solà (2010). “Large-Scale Visual Odometry for Rough Terrain”. In: *Robotics Research*. Ed. by Makoto Kaneko and Yoshihiko Nakamura. Springer Tracts in Advanced Robotics 66. DOI: [10.1007/978-3-642-14743-2_18](https://doi.org/10.1007/978-3-642-14743-2_18). Springer Berlin Heidelberg, pp. 201–212.
- Krauzlis, Richard J. (2004). “Recasting the Smooth Pursuit Eye Movement System”. In: *Journal of Neurophysiology* 91.2, pp. 591–603. DOI: [10.1152/jn.00801.2003](https://doi.org/10.1152/jn.00801.2003).
- Kubelka, Vladimír, Lorenz Oswald, François Pomerleau, Francis Colas, Tomáš Svoboda, and Michal Reinstein (2015). “Robust Data Fusion of Multimodal Sensory Information for Mobile Robots”. In: *Journal of Field Robotics* 32.4, pp. 447–473. DOI: [10.1002/rob.21535](https://doi.org/10.1002/rob.21535).
- Latombe, Jean-Claude (1991). *Robot Motion Planning*. Springer Science & Business Media.
- LaValle, Steven M. (2006). *Planning Algorithms*. Cambridge University Press.
- LaValle, Steven M. and James J. Kuffner (2001). “Randomized Kinodynamic Planning”. In: *The International Journal of Robotics Research* 20.5, pp. 378–400. DOI: [10.1177/02783640122067453](https://doi.org/10.1177/02783640122067453).
- Lebeltel, Olivier, Pierre Bessière, Julien Diard, and Emmanuel Mazer (2004). “Bayesian Robot Programming”. In: *Autonomous Robots* 16.1, pp. 49–79. DOI: [10.1023/B:AUR0.0000008671.38949.43](https://doi.org/10.1023/B:AUR0.0000008671.38949.43).
- Leutenegger, Stefan and Roland Siegwart (2012). “A low-cost and fail-safe Inertial Navigation System for airplanes”. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 612–618. DOI: [10.1109/ICRA.2012.6225061](https://doi.org/10.1109/ICRA.2012.6225061).
- Liu, Ming, François Pomerleau, Francis Colas, and Roland Siegwart (2012). “Normal estimation for pointcloud using GPU based sparse tensor voting”. In: *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 91–96. DOI: [10.1109/ROBIO.2012.6490949](https://doi.org/10.1109/ROBIO.2012.6490949).
- Longuet-Higgins, H. Christopher (1984). “The Visual Ambiguity of a Moving Plane”. In: *Proceedings of the Royal Society of London B: Biological Sciences* 223.1231, pp. 165–175. DOI: [10.1098/rspb.1984.0088](https://doi.org/10.1098/rspb.1984.0088).
- Longuet-Higgins, H. Christopher and K. Prazdny (1980). “The Interpretation of a Moving Retinal Image”. In: *Proceedings of the Royal Society of London B: Biological Sciences* 208.1173, pp. 385–397. DOI: [10.1098/rspb.1980.0057](https://doi.org/10.1098/rspb.1980.0057).
- Lowe, David G. (2004). “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2, pp. 91–110. DOI: [10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94).
- Lu, F. and Evangelos Miliotis (1997). “Globally Consistent Range Scan Alignment for Environment Mapping”. In: *Autonomous Robots* 4.4, pp. 333–349. DOI: [10.1023/A:1008854305733](https://doi.org/10.1023/A:1008854305733).
- Ma, Jeremy, Sara Susca, Max Bajracharya, Larry Matthies, Matt Malchano, and Dave Wooden (2012). “Robust multi-sensor, day/night 6-DOF pose estimation for a dynamic legged vehicle in GPS-denied environments”. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 619–626. DOI: [10.1109/ICRA.2012.6225132](https://doi.org/10.1109/ICRA.2012.6225132).
- Magnenat, Stéphane, Cédric Pradalier, and Francis Colas (2012). “Towards Non-Parametric Bayesian Learning of Robot Behaviors from Demonstration”. In: *NIPS Work-*

- shop on "Bayesian Nonparametric Models (BNPM) For Reliable Planning And Decision-Making Under Uncertainty". Lake Tahoe, United States.
- Magnusson, Martin, Andreas Nüchter, Christopher Lörken, Achim J. Lilienthal, and Joachim Hertzberg (2009). "Evaluation of 3D registration reliability and speed - A comparison of ICP and NDT". In: *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, pp. 3907–3912. DOI: [10.1109/ROBOT.2009.5152538](https://doi.org/10.1109/ROBOT.2009.5152538).
- Mahony, Robert, Tarek Hamel, and Jean-Michel Pflimlin (2008). "Nonlinear Complementary Filters on the Special Orthogonal Group". In: *IEEE Transactions on Automatic Control* 53.5, pp. 1203–1218. DOI: [10.1109/TAC.2008.923738](https://doi.org/10.1109/TAC.2008.923738).
- Marder-Eppstein, Eitan, Eric Berger, Tully Foote, Brian Gerkey, and Kurt Konolige (2010). "The Office Marathon: Robust navigation in an indoor office environment". In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 300–307. DOI: [10.1109/ROBOT.2010.5509725](https://doi.org/10.1109/ROBOT.2010.5509725).
- Masuda, Takeshi (2001). "Generation of geometric model by registration and integration of multiple range images". In: *Third International Conference on 3-D Digital Imaging and Modeling, 2001. Proceedings*, pp. 254–261. DOI: [10.1109/IM.2001.924447](https://doi.org/10.1109/IM.2001.924447).
- Masuda, Takeshi, Katsuhiko Sakaue, and Naokazu Yokoya (1996). "Registration and integration of multiple range images for 3-D model construction". In: *Proceedings of the 13th International Conference on Pattern Recognition, 1996*. Vol. 1, 879–883 vol.1. DOI: [10.1109/ICPR.1996.546150](https://doi.org/10.1109/ICPR.1996.546150).
- Maurer, Calvin R. Jr., Georges B. Aboutanos, Benoit M. Dawant, Robert J. Maciunas, and J. Michael Fitzpatrick (1996). "Registration of 3-D images using weighted geometrical features". In: *IEEE Transactions on Medical Imaging* 15.6, pp. 836–849. DOI: [10.1109/42.544501](https://doi.org/10.1109/42.544501).
- Mays, Lawrence E. and David L. Sparks (1980). "Dissociation of visual and saccade-related responses in superior colliculus neurons". In: *Journal of Neurophysiology* 43.1, pp. 207–232.
- McIlwain, James T. (1983). "Representation of the visual streak in visuotopic maps of the cat's superior colliculus: Influence of the mapping variable". In: *Vision Research* 23.5, pp. 507–516. DOI: [10.1016/0042-6989\(83\)90125-6](https://doi.org/10.1016/0042-6989(83)90125-6).
- Medioni, Gérard, Chi-Keung Tang, and Mi-Suen Lee (2000). "Tensor Voting: Theory and Applications". In: Paris.
- Mitchell, Jude F and David Zipser (2003). "Sequential memory-guided saccades and target selection: a neural model of the frontal eye fields". In: *Vision Research* 43.25, pp. 2669–2695. DOI: [10.1016/S0042-6989\(03\)00468-1](https://doi.org/10.1016/S0042-6989(03)00468-1).
- Montemerlo, Michael, Sebastian Thrun, Daphne Koller, and Ben Wegbreit (2002). "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem". In: *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, pp. 593–598.
- Moosmann, Frank and Christoph Stiller (2011). "Velodyne SLAM". In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 393–398. DOI: [10.1109/IVS.2011.5940396](https://doi.org/10.1109/IVS.2011.5940396).
- Morales, Yoichi, Alexander Carballo, Eijiro Takeuchi, Atsushi Aburadani, and Takashi Tsubouchi (2009). "Autonomous robot navigation in outdoor cluttered pedestrian walkways". In: *Journal of Field Robotics* 26.8, pp. 609–635. DOI: [10.1002/rob.20301](https://doi.org/10.1002/rob.20301).

- Moschovakis, Adonis K., Charles A. Scudder, and Stephen M. Highstein (1996). “The microscopic anatomy and physiology of the mammalian saccadic system”. In: *Progress in Neurobiology* 50.2–3, pp. 133–254. DOI: [10.1016/S0301-0082\(96\)00034-2](https://doi.org/10.1016/S0301-0082(96)00034-2).
- MRP (2013). *France Robots Initiatives – Plan national pour la robotique*. Tech. rep. Ministère du Redressement Productif.
- Nagatani, Keiji, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, and Shinji Kawatsuma (2013). “Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots”. In: *Journal of Field Robotics* 30.1, pp. 44–63. DOI: [10.1002/rob.21439](https://doi.org/10.1002/rob.21439).
- Naji, Jenny J. and Tom C. A. Freeman (2004). “Perceiving depth order during pursuit eye movement”. In: *Vision Research* 44.26, pp. 3025–3034. DOI: [10.1016/j.visres.2004.07.007](https://doi.org/10.1016/j.visres.2004.07.007).
- Neal, Mark, Tom Blanchard, Alun Hubbard, Nolwenn Chauché, Richard Bates, and John Woodward (2012). “A hardware proof of concept for a remote-controlled glacier-surveying boat”. In: *Journal of Field Robotics* 29.6, pp. 880–890. DOI: [10.1002/rob.21420](https://doi.org/10.1002/rob.21420).
- Nguyen, Q. V., F. Colas, E. Vincent, and F. Charpillet (2016). “Localizing an intermittent and moving sound source using a mobile robot”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1986–1991. DOI: [10.1109/IROS.2016.7759313](https://doi.org/10.1109/IROS.2016.7759313).
- Nüchter, Andreas, Hartmut Surmann, Kai Lingemann, Joachim Hertzberg, and Sebastian Thrun (2004). “6D SLAM with an application in autonomous mine mapping”. In: *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. Vol. 2, 1998–2003 Vol.2*. DOI: [10.1109/ROBOT.2004.1308117](https://doi.org/10.1109/ROBOT.2004.1308117).
- Okamura, Allison, Maja Mataric, and Henrik Christensen (2010). “Medical and Health-Care Robotics”. In: *IEEE Robotics & Automation Magazine* 17.3, pp. 26–37. DOI: [10.1109/MRA.2010.937861](https://doi.org/10.1109/MRA.2010.937861).
- Ottes, Fenno P., Jan A. M. Van Gisbergen, and Jos J. Eggermont (1986). “Visuomotor fields of the superior colliculus: A quantitative model”. In: *Vision Research* 26.6, pp. 857–873. DOI: [10.1016/0042-6989\(86\)90144-6](https://doi.org/10.1016/0042-6989(86)90144-6).
- Pan, Ye, Bo Dai, and Qicong Peng (2010). “Fast and robust 3D face matching approach”. In: *2010 International Conference on Image Analysis and Signal Processing (IASP)*, pp. 195–198. DOI: [10.1109/IASP.2010.5476132](https://doi.org/10.1109/IASP.2010.5476132).
- Pêtrès, Clément, Yan Pailhas, Pedro Patrón, Yvan Petillot, Jonathan Evans, and David Lane (2007). “Path Planning for Autonomous Underwater Vehicles”. In: *IEEE Transactions on Robotics* 23.2, pp. 331–341. DOI: [10.1109/TR0.2007.895057](https://doi.org/10.1109/TR0.2007.895057).
- Pfister, Sam T., Kristo L. Kriechbaum, Stergios I. Roumeliotis, and Joel W. Burdick (2002). “Weighted range sensor matching algorithms for mobile robot displacement estimation”. In: *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02. Vol. 2, 1667–1674 vol.2*. DOI: [10.1109/ROBOT.2002.1014782](https://doi.org/10.1109/ROBOT.2002.1014782).
- Philippsen, Roland and Roland Siegwart (2005). “An Interpolated Dynamic Navigation Function”. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, pp. 3782–3789. DOI: [10.1109/ROBOT.2005.1570697](https://doi.org/10.1109/ROBOT.2005.1570697).

- Pivtoraiko, Mihail, Ross A. Knepper, and Alonzo Kelly (2009). “Differentially constrained mobile robot motion planning in state lattices”. In: *Journal of Field Robotics* 26.3, pp. 308–333. DOI: [10.1002/rob.20285](https://doi.org/10.1002/rob.20285).
- Pomerleau, François (2013). “Applied Registration for Robotics: Methodology and Tools for ICP-like Algorithms”. PhD thesis. Diss., Eidgenössische Technische Hochschule ETH.
- Pomerleau, François, Francis Colas, François Ferland, and François Michaud (2010). “Relative Motion Threshold for Rejection in ICP Registration”. In: *Field and Service Robotics*. Ed. by Andrew Howard, Karl Iagnemma, and Alonzo Kelly. Springer Tracts in Advanced Robotics 62. Springer Berlin Heidelberg, pp. 229–238.
- Pomerleau, François, Francis Colas, and Roland Siegwart (2015). “A Review of Point Cloud Registration Algorithms for Mobile Robotics”. In: *Foundations and Trends in Robotics* 4.1, pp. 1–104.
- Pomerleau, François, Francis Colas, Roland Siegwart, and Stéphane Magnenat (2013). “Comparing ICP variants on real-world data sets”. In: *Autonomous Robots* 34.3, pp. 133–148. DOI: [10.1007/s10514-013-9327-2](https://doi.org/10.1007/s10514-013-9327-2).
- Pomerleau, François, Ming Liu, Francis Colas, and Roland Siegwart (2012). “Challenging data sets for point cloud registration algorithms”. In: *The International Journal of Robotics Research* 31.14, pp. 1705–1711. DOI: [10.1177/0278364912458814](https://doi.org/10.1177/0278364912458814).
- Pomerleau, François, Stéphane Magnenat, Francis Colas, Ming Liu, and Roland Siegwart (2011). “Tracking a depth camera: Parameter exploration for fast ICP”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, pp. 3824–3829.
- Pulli, Kari (1999). “Multiview registration for large data sets”. In: *Second International Conference on 3-D Digital Imaging and Modeling, 1999. Proceedings*, pp. 160–168. DOI: [10.1109/IM.1999.805346](https://doi.org/10.1109/IM.1999.805346).
- Pylyshyn, Zenon W. and Vidal Annan (2006). “Dynamics of target selection in Multiple Object Tracking (MOT)”. In: *Spatial Vision* 19.6, pp. 485–504. DOI: [10.1163/156856806779194017](https://doi.org/10.1163/156856806779194017).
- Pylyshyn, Zenon W. and Ron W. Storm (1988). “Tracking multiple independent targets: Evidence for a parallel tracking mechanism*”. In: *Spatial Vision* 3.3, pp. 179–197. DOI: [10.1163/156856888X00122](https://doi.org/10.1163/156856888X00122).
- Remondino, Fabio and Clive Fraser (2006). “Digital Camera Calibration Methods: Considerations and Comparisons”. In: vol. 36, pp. 266–272.
- Richter, Silvia and Matthias Westphal (2010). “The LAMA Planner: Guiding Cost-based Anytime Planning with Landmarks”. In: *J. Artif. Int. Res.* 39.1, pp. 127–177.
- Rio, M., F. Colas, M. Andries, and F. Charpillet (2016). “Probabilistic sensor data processing for robot localization on load-sensing floors”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4544–4550. DOI: [10.1109/ICRA.2016.7487654](https://doi.org/10.1109/ICRA.2016.7487654).
- Robinson, D. A. (1972). “Eye movements evoked by collicular stimulation in the alert monkey”. In: *Vision Research* 12.11, pp. 1795–1808.
- Rogers, Brian J. and Maureen Graham (1979). “Motion parallax as an independent cue for depth perception”. In: *Perception* 8.2, pp. 125–134. DOI: [10.1068/p080125](https://doi.org/10.1068/p080125).
- Rogers, Sheena and Brian J. Rogers (1992). “Visual and nonvisual information disambiguate surfaces specified by motion parallax”. In: *Perception & Psychophysics* 52.4, pp. 446–452. DOI: [10.3758/BF03206704](https://doi.org/10.3758/BF03206704).

- Rusinkiewicz, Szymon and Marc Levoy (2001). “Efficient variants of the ICP algorithm”. In: *Third International Conference on 3-D Digital Imaging and Modeling, 2001. Proceedings*, pp. 145–152. DOI: [10.1109/IM.2001.924423](https://doi.org/10.1109/IM.2001.924423).
- Savage, Paul G. (1998). “Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms”. In: *Journal of Guidance, Control, and Dynamics* 21.2, pp. 208–221. DOI: [10.2514/2.4242](https://doi.org/10.2514/2.4242).
- Scaramuzza, Davide and Friedrich Fraundorfer (2011). “Visual Odometry [Tutorial]”. In: *IEEE Robotics Automation Magazine* 18.4, pp. 80–92. DOI: [10.1109/MRA.2011.943233](https://doi.org/10.1109/MRA.2011.943233).
- Scharstein, Daniel and Richard Szeliski (2002). “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms”. In: *International Journal of Computer Vision* 47.1-3, pp. 7–42. DOI: [10.1023/A:1014573219977](https://doi.org/10.1023/A:1014573219977).
- Schwartz, Eric L. (1980). “Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding”. In: *Vision Research* 20.8, pp. 645–669. DOI: [10.1016/0042-6989\(80\)90090-5](https://doi.org/10.1016/0042-6989(80)90090-5).
- Scudder, Charles A., Chris R. Kaneko, and Albert F. Fuchs (2002). “The brainstem burst generator for saccadic eye movements”. In: *Experimental Brain Research* 142.4, pp. 439–462. DOI: [10.1007/s00221-001-0912-9](https://doi.org/10.1007/s00221-001-0912-9).
- Segal, Aleksandr V., Dirk Haehnel, and Sebastian Thrun (2009). “Generalized-ICP”. In: *Robotics: Science and Systems V. Proceedings of the Conference on*.
- Siminoff, Robert, Horst Otto Schwassmann, and Lawrence Kruger (1966). “An electrophysiological study of the visual projection to the superior colliculus of the rat”. In: *The Journal of Comparative Neurology* 127.4, pp. 435–444. DOI: [10.1002/cne.901270402](https://doi.org/10.1002/cne.901270402).
- Sommer, Marc A. and Robert H. Wurtz (2000). “Composition and Topographic Organization of Signals Sent From the Frontal Eye Field to the Superior Colliculus”. In: *Journal of Neurophysiology* 83.4, pp. 1979–2001.
- SPARC (2014). *Strategic Research Agenda for Robotics in Europe 2014–2020*. Tech. rep. Eu-Robotics aisbl.
- Stewart, Charles V., Chia-Ling Tsai, and Badrinath Roysam (2003). “The dual-bootstrap iterative closest point algorithm with application to retinal image registration”. In: *IEEE Transactions on Medical Imaging* 22.11, pp. 1379–1394. DOI: [10.1109/TMI.2003.819276](https://doi.org/10.1109/TMI.2003.819276).
- Stumm, Elena, Andreas Breitenmoser, François Pomerleau, Cédric Pradalier, and Roland Siegwart (2012). “Tensor-voting-based navigation for robotic inspection of 3D surfaces using lidar point clouds”. In: *The International Journal of Robotics Research* 31.12, pp. 1465–1488. DOI: [10.1177/0278364912461537](https://doi.org/10.1177/0278364912461537).
- Subramanian, Anbumani, Xiaojin Gong, Jamie N. Riggins, Daniel J. Stilwell, and Christopher L. Wyatt (2006). “Shoreline Mapping using an Omni-directional Camera for Autonomous Surface Vehicle Applications”. In: *OCEANS 2006*, pp. 1–6. DOI: [10.1109/OCEANS.2006.306906](https://doi.org/10.1109/OCEANS.2006.306906).
- Tanner, Thomas, Luiz Henrique Canto-Pereira, and Heinrich Bühlhoff (2007). “Free vs constrained gaze in a multiple-object-tracking paradigm”. In:
- Thrun, Sebastian, Wolfram Burgard, and Dieter Fox (1998). “A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots”. In: *Autonomous Robots* 5.3-4, pp. 253–271. DOI: [10.1023/A:1008806205438](https://doi.org/10.1023/A:1008806205438).

- Tisdale, John, ZuWhan Kim, and J. Karl Hedrick (2009). “Autonomous UAV path planning and estimation”. In: *IEEE Robotics Automation Magazine* 16.2, pp. 35–42. DOI: [10.1109/MRA.2009.932529](https://doi.org/10.1109/MRA.2009.932529).
- Todd, James T. and Paola Bressan (1990). “The perception of 3-dimensional affine structure from minimal apparent motion sequences”. In: *Perception & Psychophysics* 48.5, pp. 419–430. DOI: [10.3758/BF03211585](https://doi.org/10.3758/BF03211585).
- Todd, James T. and J. Farley Norman (1991). “The visual perception of smoothly curved surfaces from minimal apparent motion sequences”. In: *Perception & Psychophysics* 50.6, pp. 509–523. DOI: [10.3758/BF03207535](https://doi.org/10.3758/BF03207535).
- Tompkins, Paul, Anthony Stentz, and David Wettergreen (2004). “Global path planning for Mars rover exploration”. In: *2004 IEEE Aerospace Conference, 2004. Proceedings*. Vol. 2, 801–815 Vol.2. DOI: [10.1109/AERO.2004.1367681](https://doi.org/10.1109/AERO.2004.1367681).
- Trawny, Nikolas and Stergios I. Roumeliotis (2005). *Indirect Kalman Filter for 3D Attitude Estimation*. Technical Report 2005-002. Mars Lab: University of Minnesota.
- Tsai, Chia-Ling, Chun-Yi Li, Gehua Yang, and Kai-Shung Lin (2010). “The Edge-Driven Dual-Bootstrap Iterative Closest Point Algorithm for Registration of Multimodal Fluorescein Angiogram Sequence”. In: *IEEE Transactions on Medical Imaging* 29.3, pp. 636–649. DOI: [10.1109/TMI.2009.2030324](https://doi.org/10.1109/TMI.2009.2030324).
- Ullman, Shimon (1979). *The interpretation of visual motion*. Oxford, England: Massachusetts Inst of Technology Pr.
- Walker, Michael W., Lejun Shao, and Richard A. Volz (1991). “Estimating 3-D location parameters using dual number quaternions”. In: *CVGIP: Image Understanding* 54.3, pp. 358–367. DOI: [10.1016/1049-9660\(91\)90036-0](https://doi.org/10.1016/1049-9660(91)90036-0).
- Wallach, Hans and D. N. O’Connell (1953). “The kinetic depth effect”. In: *Journal of Experimental Psychology* 45.4, pp. 205–217. DOI: [10.1037/h0056880](https://doi.org/10.1037/h0056880).
- Wallach, Hans, Linda Stanton, and Dean Becker (1974). “The compensation for movement-produced changes of object orientation”. In: *Perception & Psychophysics* 15.2, pp. 339–343. DOI: [10.3758/BF03213955](https://doi.org/10.3758/BF03213955).
- Weiss, Stephan, Markus W. Achtelik, Simon Lynen, Michael C. Achtelik, Laurent Kneip, Margarita Chli, and Roland Siegwart (2013). “Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium”. In: *Journal of Field Robotics* 30.5, pp. 803–831. DOI: [10.1002/rob.21466](https://doi.org/10.1002/rob.21466).
- Weiss, Yair, Eero P. Simoncelli, and Edward H. Adelson (2002). “Motion illusions as optimal percepts”. In: *Nature Neuroscience* 5.6, pp. 598–604. DOI: [10.1038/nn0602-858](https://doi.org/10.1038/nn0602-858).
- Wexler, Mark (2003). “Voluntary Head Movement and Allocentric Perception of Space”. In: *Psychological Science* 14.4, pp. 340–346. DOI: [10.1111/1467-9280.14491](https://doi.org/10.1111/1467-9280.14491).
- Wexler, Mark, Ivan Lamouret, and Jacques Droulez (2001). “The stationarity hypothesis: an allocentric criterion in visual perception”. In: *Vision Research* 41.23, pp. 3023–3037. DOI: [10.1016/S0042-6989\(01\)00190-0](https://doi.org/10.1016/S0042-6989(01)00190-0).
- Wexler, Mark, Francesco Panerai, Ivan Lamouret, and Jacques Droulez (2001). “Self-motion and the perception of stationary objects”. In: *Nature* 409.6816, pp. 85–88. DOI: [10.1038/35051081](https://doi.org/10.1038/35051081).
- Wurtz, Robert H., Marc A. Sommer, Martin Paré, and Stefano Ferraina (2001). “Signal transformations from cerebral cortex to superior colliculus for the generation of saccades”. In: *Vision Research* 41.25-26, pp. 3399–3412. DOI: [10.1016/S0042-6989\(01\)00066-9](https://doi.org/10.1016/S0042-6989(01)00066-9).

- Yoshitaka, Hara, Kawata Hirohiko, Ohya Akihisa, and Yuta Shin'ichi (2006). "Mobile Robot Localization and Mapping by Scan Matching using Laser Reflection Intensity of the SOKUIKI Sensor". In: *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pp. 3018–3023. DOI: [10.1109/IECON.2006.347701](https://doi.org/10.1109/IECON.2006.347701).
- Zhang, Zhengyou (1994). "Iterative point matching for registration of free-form curves and surfaces". In: *International Journal of Computer Vision* 13.2, pp. 119–152. DOI: [10.1007/BF01427149](https://doi.org/10.1007/BF01427149).

Curriculum Vitæ

1 Formation

- 2002–2006 : thèse de *doctorat en informatique* de l'Institut National Polytechnique de Grenoble (INPG) soutenue le 27 janvier 2006 :
 - *Perception des objets en mouvement. Composition bayésienne du flux optique et du mouvement de l'observateur*,
 - sous la direction de Pierre Bessière et Jacques Droulez ;
- 2001–2002 : *Diplôme d'Études Approfondies* de l'INPG, spécialité « Informatique, Systèmes et Communication » :
 - *Modélisation bayésienne des cellules de direction de la tête*,
 - sous la direction de Pierre Bessière ;
- 1999–2002 : *Ingénieur* de l'École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble (ENSIMAG), spécialité « Logiciels et langages ».

2 Expérience professionnelle

- Depuis 10.2014 : *Chargé de recherche* de première classe chez Inria dans l'équipe LARSEN du centre Inria Nancy – Grand Est commune avec le LORIA (UMR 7503) ;
- 2009–09.2014 : *Chercheur* à l'AUTONOMOUS SYSTEMS LAB du Pr. Roland Siegwart à l'École Polytechnique Fédérale de Zurich (ETHZ) ;
- 2007–2009 : *Séjour post-doctoral* au LABORATOIRE DE PHYSIOLOGIE DE LA PERCEPTION ET DE L'ACTION du Pr. Berthoz au Collège de France :
 - modèles bayésiens de cartes rétinotopiques pour la sélection de mouvements oculaires,
 - en collaboration avec Dr. Benoît Girard ;
- 2006 : *Séjour post-doctoral* dans l'équipe E-MOTION, Inria Grenoble – Rhône-Alpes :
 - détection et suivi de véhicule pour la conduite autonome,
 - transfert technologique,
 - sous la direction de Dr. Christian Laugier.

3 Enseignement et encadrement

3.1 Cours

Niveau Master

- CentraleSupélec majeure « Systèmes Interactifs et Robotiques » en troisième année :

- cours de « Robotique Autonome » 12 heures de cours magistral depuis 2016 ;
- Mines de Nancy :
 - Introduction à ROS : cours et TP ;
- Master à l'ETH Zurich :
 - Participation aux travaux pratiques du cours *Autonomous Mobile Robots* en 2010 et 2011 ;
 - cours *Information Processing for Robotics* en 2009, 2010 et 2011 conçu avec deux autres personnes.

3.2 Encadrement individuel

Co-direction de thèse

- François Pomerleau, en co-direction avec Pr. Roland Siegwart (ETH Zurich) :
 - *Applied Registration for Robotics – Methodology and Tools for ICP-like Algorithms*,
 - soutenue le 3 mai 2013 ;
- Ming Liu, en co-direction avec Pr. Roland Siegwart (ETH Zurich) :
 - *Topological Scene Recognition and Visual Navigation for Mobile Robots using Omnidirectional Cameras*,
 - soutenue le 21 juin 2013 ;
- Van Quan Nguyen en co-direction avec Emmanuel Vincent (Inria – Loria) :
 - *Mapping of a sound environment with a mobile robot*,
 - commencée en novembre 2014.

Projets de master

Titre	Étudiant	Dates	Université
<i>If plants could walk : robust control</i>	Janine Stocker	2010– 2011	ETH Zurich
<i>3D dataset construction with ground truth</i>	Benoît Lescot	2011	ETH Zurich
<i>Trajectory control of an adaptive robot</i>	Srivatsa Mahesh	2012	ETH Zurich
<i>Safe navigation for Search and Rescue robots</i>	Lorenz Oswald	2013	ETH Zurich
<i>Localisation pour la robotique mobile</i>	Corentin Fagot	2015	Université de Lorraine

Stages de première année de master

Titre	Étudiant	Dates	Université
<i>Generation of a set of feasible vehicle motion primitives</i>	Florian Volken	2009	ETH Zurich
<i>Incremental topological segmentation for robot mapping</i>	Luc Oth	2011	ETH Zurich
<i>Kinematic model of tracked or 4/6-parallel-wheeled vehicles</i>	Rocco Bottani	2011	ETH Zurich

<i>Robust localization for Search and Rescue robots</i>	Lorenz Oswald	2012	ETH Zurich
<i>Multi-target tracking on pressure tiles</i>	Alexis Grall	2016	ENSEIRB-MATMECA
<i>Data collection for deep learning</i>	Mohammad-Rami Koujan	2016	University of Girona

Projets de licence

Titre	Étudiant(s)	Dates	Université
<i>Obstacle management for a ballbot</i>	Michael Neunert	2010	ETH Zurich
<i>Arrester between a ball and a body</i>	Lukas Limacher	2010	ETH Zurich
<i>Signal processing for a ballbot</i>	Jonathan Hüsey	2010	ETH Zurich
<i>Control of a ballbot</i>	Péter Fankhauser et Corsin Gwerder	2010	ETH Zurich
<i>Mechatrical realization of a design concept for a ballbot</i>	Jérôme Käser	2010	ETH Zurich
<i>Evaluation and implementation of a control device for a ballbot</i>	Thomas Kammermann	2010	ETH Zurich
<i>Time-optimal trajectories for a ballbot</i>	Simon Dössegger	2010	ETH Zurich
<i>If plants could walk : Intelligence</i>	Michael Marti et Bryan Anastasiades	2010	ETH Zurich
<i>Studies on Mechatronics : Review on Iterative Closest Point</i>	Lorenz Oswald et Michael Wild	2010–2011	ETH Zurich
<i>Snake Joystick</i>	Dominique Hunziker	2011	ETH Zurich

4 Intégration à la communauté de recherche

4.1 Activités de review et d'organisation

- Reviewer chaque année des conférences IROS et ICRA ;
- reviewer occasionnel des conférences FSR, RSS et SAB ;
- reviewer pour les journaux *Autonomous Robot*, *Journal of Field Robotics*, *IEEE Robotics and Automation Magazine* ;
- 2010 : membre du *Programme Committee* de SAB2010 : *The 11th International Conference on the Simulation of Adaptive Behavior 2010* ;
- 2013 : co-organisateur du *Rescue Robotics Camp* (école d'automne) durant SSRR 2013 : *11th IEEE International Symposium on Safety, Security, and Rescue Robotics* ;
- 2014 : membre du *Programme Committee* de SAB2014 : *The 13th International Conference on the Simulation of Adaptive Behavior 2014* ;
- 2016 : membre du *Technical Programme Committee* de SCiFI2016 : *EAI International Conference on Smart Cities and the Future Internet 2016*.

4.2 Responsabilités collectives

- Responsable permanent de l'équipe Larsen ;

- membre de la commission développement technologique (CDT) ;
- membre de la commission des utilisateurs des moyens informatiques (CUMI) ;
- 2016 : membre du comité de sélection pour le poste de maître de conférence à l'ESS-TIN ;
- 2016 : membre du jury pour la soutenance à mi-parcours de Jean-Nicola Russo : « évaluation des risques routiers à l'aide de réseaux bayésiens » à l'Université de Haute Alsace.

4.3 Autres

- 2012 : intervenant pour le GDR robotique le 11 juillet 2012 : « Retour d'expérience sur ROS » ;
- 2013 : enseignant pour le GDR robotique le 10 juillet 2013 : « Journée Technique sur ROS » à Montpellier ;
- 2013 : participation en tant qu'expert à un panel sur la coordination des projets de sauvetage et recherche en Europe à SSRR 2013.

5 Contrats et projets

5.1 Investigateur principal

- Rédaction et gestion pour l'ETH Zurich du projet européen NIFTi (FP7-ICT-247870) d'un budget total de 6,6 M€ dont 0,98 M€ pour l'ETH Zurich (2010–2013) ;
- rédaction et gestion pour l'ETH Zurich du projet européen TRADR (FP7-ICT-609763) d'un budget total de 8,7 M€ dont 1,1 M€ pour l'ETH Zurich (nov. 2013–2017).

5.2 Participant

- 2002–2005 : participation au projet européen BIBA (FP5-IST-2001-32115) ;
- 2007–2008 : participation au projet européen BACS (FP6-IST-027140) ;
- 2009 : participation et gestion du projet européen BACS (FP6-IST-027140).

6 Prix

- 2011 : deuxième place dans la catégorie *Most Useful* du concours *ROS 3D Contest* pour la soumission *Automatic Calibration of Extrinsic Parameters* ;
- 2013 : prix Robocup du meilleur papier à IROS 2013 : *3D path planning and execution for search and rescue ground robots*.

7 Liste des publications

7.1 Chapitre de livre

- [1] Colas, Francis, Pierre Bessière, Jacques Droulez, and Mark Wexler (2008). “Bayesian modelling of perception of structure from motion”. In: *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*. Springer Berlin/Heidelberg, pp. 301–328.

7.2 Journaux

- [1] Bessière, Pierre, Julien Diard, and Francis Colas (2016). “Modèles probabilistes formels pour problèmes cognitifs usuels”. In: *Intellectica* 65.1, pp. 111–141.
- [2] Hitz, Gregory, François Pomerleau, Francis Colas, and Roland Siegwart (2015). “Relaxing the planar assumption: 3D state estimation for an autonomous surface vessel”. In: *The International Journal of Robotics Research* 34.13, pp. 1604–1621.
- [3] Kruijff-Korbayová, Ivana, Francis Colas, Mario Gianni, Fiora Pirri, Joachim de Greeff, Koen Hindriks, Mark Neerincx, Petter Ögren, Tomáš Svoboda, and Rainer Worst (2015). “TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response”. In: *Künstliche Intelligenz* 29.2, pp. 193–201.
- [4] Kubelka, Vladimír, Lorenz Oswald, François Pomerleau, Francis Colas, Tomáš Svoboda, and Michal Reinstein (2015). “Robust data fusion of multi-modal sensory information for mobile robots”. In: *Journal of Field Robotics* 32.4, pp. 447–473.
- [5] Liu, Ming, Francis Colas, Luc Oth, and Roland Siegwart (2015). “Incremental topological segmentation for semi-structured environments using discretized GVG”. In: *Autonomous Robots* 38.2, pp. 143–160.
- [6] Pomerleau, François, Francis Colas, and Roland Siegwart (2015). “A Review of Point Cloud Registration Algorithms for Mobile Robotics”. In: *Foundations and Trends in Robotics* 4.1, pp. 1–104.
- [7] Kruijff, Geert-Jan M, Ivana Kruijff-Korbayová, Shanker Keshavdas, Benoit Larochelle, Miroslav Janíček, Francis Colas, Ming Liu, François Pomerleau, Roland Siegwart, Mark Neerincx, Rosemarijn Looije, Nanja Smets, Tina Mioch, Jurriaan van Diggelen, Fiora Pirri, Mario Gianni, Federico Ferri, Matteo Menna, Rainer Worst, Thorsten Linder, Varoslav Tretyakov, Hartmut Surmann, Tomáš Svoboda, Michal Reinstein, Karel Zimmermann, Tomáš Petříček, and Vašek Hlaváč (2014). “Designing, developing, and deploying systems to support human-robot teams in disaster response”. In: *Advanced Robotics* 28.23, pp. 1547–1570.
- [8] Pomerleau, François, Francis Colas, Roland Siegwart, and Stéphane Magnenat (2013). “Comparing ICP variants on real-world data sets”. In: *Autonomous Robots* 34.3, pp. 133–148.
- [9] Pomerleau, François, Ming Liu, Francis Colas, and Roland Siegwart (2012). “Challenging data sets for point cloud registration algorithms”. In: *The International Journal of Robotics Research* 31.14, pp. 1705–1711.
- [10] Colas, Francis, Julien Diard, and Pierre Bessière (2010). “Common Bayesian models for common cognitive issues”. In: *Acta Biotheoretica* 58.2-3, pp. 191–216.

- [11] Perrin, Xavier, Ricardo Chavarriaga, Francis Colas, Roland Siegwart, and José del R Millán (2010). “Brain-coupled interaction for semi-autonomous navigation of an assistive robot”. In: *Robotics and Autonomous Systems* 58.12, pp. 1246–1255.
- [12] Colas, Francis, Fabien Flacher, Thomas Tanner, Pierre Bessière, and Benoît Girard (2009). “Bayesian models of eye movement selection with retinotopic maps”. In: *Biological Cybernetics* 100.3, pp. 203–214.
- [13] Colas, Francis, Jacques Droulez, Mark Wexler, and Pierre Bessière (2007). “A unified probabilistic model of the perception of three-dimensional structure from optic flow”. In: *Biological Cybernetics* 97.5, pp. 461–477.

7.3 Conférences internationales

- [1] Nguyen, Quan V., Francis Colas, Emmanuel Vincent, and François Charpillet (2017). “Long-term robot motion planning for active sound source localization with Monte-Carlo tree search”. In: *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2017 Fifth Joint Workshop on*. IEEE, pp. 1–5.
- [2] — (2016). “Localizing an intermittent and moving sound source using a mobile robot”. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, pp. 1–6.
- [3] Rio, Maxime, Francis Colas, Mihai Andries, and François Charpillet (2016). “Probabilistic sensor data processing for robot localization on load-sensing floors”. In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE.
- [4] Wang, Shiling, Francis Colas, Ming Liu, Francesco Mondada, and Stéphane Magnenat (2016). “Localization of inexpensive robots with low-bandwidth sensors”. In: *Distributed Autonomous Robotic Systems (DARS), 2016 International Symposium on*, pp. 1–14.
- [5] Hitz, Gregory, François Pomerleau, Francis Colas, and Roland Siegwart (2014). “State estimation for shore monitoring using an autonomous surface vessel”. In: *International Symposium on Experimental Robotics (ISER)*.
- [6] Pomerleau, François, Philipp Krüsi, Francis Colas, Paul Furgale, and Roland Siegwart (2014). “Long-term 3D map maintenance in dynamic environments”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, pp. 3712–3719.
- [7] Colas, Francis, Srivatsa Mahesh, François Pomerleau, Ming Liu, and Roland Siegwart (2013). “3D path planning and execution for search and rescue ground robots”. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, pp. 722–727.
- [8] Keshavdas, Shanker, Hendrik Zender, Geert-Jan M Kruijff, Ming Liu, and Francis Colas (2012). “Functional mapping: spatial inferencing to aid human-robot rescue efforts in unstructured disaster environments”. In: *Proceedings of the 2012 AAI Spring Symposium on Designing Intelligent Robots*.
- [9] Kruijff, Geert-Jan M, Francis Colas, Tomáš Svoboda, Jurriaan van Diggelen, Patrick Balmer, Fiora Pirri, and Rainer Worst (2012). “Designing intelligent robots for human-robot teaming in urban search & rescue”. In: *Proceedings of the AAI 2012 Spring Symposium on Designing Intelligent Robots*.

- [10] Kruijff, Geert-Jan M, Miroslav Janíček, Shanker Keshavdas, Benoit Larochelle, Hendrik Zender, Nanja Smets, Tina Mioch, Mark Neerincx, Jurriaan van Diggelen, Francis Colas, Ming Liu, François Pomerleau, Roland Siegwart, Vašek Hlaváč, Tomáš Svoboda, Tomáš Petříček, Michal Reinstein, Karel Zimmermann, Fiora Pirri, Mario Gianni, Panagiotis Papadakis, Arnab Sinha, Patrick Balmer, Nicola Tomatis, Rainer Worst, Thorsten Linder, Hartmut Surmann, Varoslav Tretyakov, Salvatore Corrao, Sylvia Pratzler-Wanczura, and Markus Sulk (2012). “Experience in system design for human-robot teaming in urban search & rescue”. In: *Proc. of 8th Int. Conf. on Field and Service Robotics (FSR 2012), Matsushima, Japan, Jul.*
- [11] Liu, Ming, Francis Colas, François Pomerleau, and Roland Siegwart (2012). “A Markov semi-supervised clustering approach and its application in topological map extraction”. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on.* IEEE, pp. 4743–4748.
- [12] Liu, Ming, François Pomerleau, Francis Colas, and Roland Siegwart (2012). “Normal estimation for pointcloud using GPU based sparse tensor voting”. In: *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on.* IEEE, pp. 91–96.
- [13] Magnenat, Stéphane, Cédric Pradalier, and Francis Colas (2012). “Towards non-parametric Bayesian learning of robot behaviors from demonstration”. In: *NIPS Workshop on “Bayesian Nonparametric Models (BNPM) For Reliable Planning And Decision-Making Under Uncertainty”.*
- [14] Pomerleau, François, Andreas Breitenmoser, Ming Liu, Francis Colas, and Roland Siegwart (2012). “Noise characterization of depth sensors for surface inspections”. In: *Applied Robotics for the Power Industry (CARPI), 2012 2nd International Conference on.* IEEE, pp. 16–21.
- [15] Gianni, Mario, Panagiotis Papadakis, Fiora Pirri, Ming Liu, Francois Pomerleau, Francis Colas, Karel Zimmermann, Tomáš Svoboda, Tomáš Petříček, Geert-Jan M Kruijff, Harmish Khambhaita, and Hendrik Zender (2011). “A unified framework for planning and execution-monitoring of mobile robots”. In: *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence.*
- [16] Liu, Ming, Francis Colas, and Roland Siegwart (2011). “Regional topological segmentation based on mutual information graphs”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE, pp. 3269–3274.
- [17] Perrin, Xavier, Francis Colas, Ricardo Chavarriaga, Cédric Pradalier, José del R Millán, and Roland Siegwart (2011). “Learning user habits for semi-autonomous navigation using low throughput interfaces”. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC).* IEEE, pp. 1–6.
- [18] Pomerleau, François, Benoit Lescot, Francis Colas, Ming Liu, and Roland Siegwart (2011). “Dataset acquisitions for USAR environments”. In: *2011 AAAI Fall Symposium Series.*
- [19] Pomerleau, François, Stéphane Magnenat, Francis Colas, Ming Liu, and Roland Siegwart (2011). “Tracking a depth camera: parameter exploration for fast ICP”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on.* IEEE, pp. 3824–3829.
- [20] Stocker, Janine, Aline Veillat, Stéphane Magnenat, Francis Colas, and Roland Siegwart (2011). “Towards adaptive robotic green plants”. In: *Towards Autonomous Robotic Systems.* Springer Berlin/Heidelberg, pp. 422–423.

- [21] Sturm, Jürgen, Stéphane Magnenat, Nikolas Engelhard, François Pomerleau, Francis Colas, Daniel Cremers, Roland Siegwart, and Wolfram Burgard (2011). “Towards a benchmark for RGB-D SLAM evaluation”. In: *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf.(RSS), Los Angeles, USA*. Vol. 2.
- [22] Colas, Francis, Pierre Bessière, and Benoît Girard (2010). “Maximum entropy perception-action space: a Bayesian model of eye movement selection”. In: *30th Conf. on Bayesian Methods and Maximum Entropy in Science and Engineering (MaxEnt2010)*. Vol. 1305. American Institute of Physics, pp. 83–90.
- [23] Perrin, Xavier, Francis Colas, Cédric Pradalier, and Roland Siegwart (2009). “Learning to identify users and predict their destination in a robotic guidance application”. In: *Field and Service Robotics*. Springer Berlin/Heidelberg, pp. 377–387.
- [24] Pomerleau, François, Francis Colas, François Ferland, and François Michaud (2009). “Relative motion threshold for rejection in ICP registration”. In: *Field and Service Robotics*. Springer Berlin/Heidelberg, pp. 229–238.
- [25] Bessière, Pierre and Francis Colas (2007). “Bayesian Programming: life science modeling and robotics applications”. In: *International Symposium of Robotics Research*.
- [26] Pradalier, Cédric, Francis Colas, and Pierre Bessière (2003a). “Expressing Bayesian fusion as a product of distributions: application to randomized Hough transform”. In: *Bayesian Methods and Maximum Entropy in Science and Engineering (MaxEnt)*. AIP.
- [27] — (2003b). “Expressing Bayesian fusion as a product of distributions: applications in robotics”. In: *Intelligent Robots and Systems (IROS), 2003 IEEE/RSJ International Conference on*. Vol. 2. IEEE, pp. 1851–1856.

7.4 Autres

- [1] Colas, Francis, Fabien Flacher, Pierre Bessière, and Benoît Girard (2008). “Explicit uncertainty for eye movement selection”. In: *Deuxième conférence française de neurosciences computationnelles (Neurocomp08)*.
- [2] Colas, Francis (2006). “Perception des objets en mouvement. Composition bayésienne du flux optique et du mouvement de l’observateur”. PhD thesis. Institut National Polytechnique de Grenoble.

Résumé

Ce document décrit mes travaux de recherche autour de la modélisation bayésienne et de la robotique. Mon travail a commencé par la modélisation de processus biologiques avant, dans un deuxième temps, d'évoluer vers la robotique. Dans les deux cas, je me suis intéressé à la fois à la perception et à l'action. J'ai donc proposé un modèle de la perception humaine de plans par le flux optique qui réunit deux hypothèses de la littérature dans un cadre unique. J'ai aussi proposé et comparé différents modèles de la sélection de mouvement oculaire dans une tâche de suivi multi-cibles, et montré que le modèle prenant en compte explicitement l'incertitude proposait des mouvements plus proches de ceux des sujets.

Du côté robotique, j'ai travaillé sur l'estimation d'état de plusieurs robots avec des techniques classiques de filtrage mais en incluant la fusion de plusieurs sources d'informations de nature et caractéristiques différentes. Je discute aussi de l'algorithme d'*Iterative Closest Point* pour proposer une méthode plus rigoureuse d'évaluation des différentes variantes. Le dernier travail que je présente concerne la planification en ligne et l'exécution de chemin pour un robot à chenille avec des capacités de franchissement importantes.

Je conclus ce document par des perspectives de travail sur ce que j'appelle la robotique située, c'est-à-dire des robots non plus isolés mais plongés dans un environnement équipé de capteurs et partagé avec des humains.

Mots clés : Modélisation bayésienne, robotique, perception, estimation d'état, planification de chemin.

Abstract

This document describes my research around Bayesian modeling and robotics. My work started with the modeling of biological processes before evolving towards robotics. In both cases, I was interested in both perception and action. I first proposed a model of human perception of planar surfaces with optic flow which fuses in a single framework two concurrent hypotheses of the literature. I also proposed and compared several models of eye movement selection in a Multiple Object Tracking task. I was able to show that the model with explicit uncertainty was the closest to the subjects eye movements.

In robotics, I worked on the state estimation of several robots with classical filtering techniques but also including fusion of multiple sources of information of various nature and characteristics. I also discuss the Iterative Closest Point algorithm for which we proposed a more rigorous method for evaluating the different variants. The last piece of work I present deals with online three-dimensional path planning and execution of a tracked robot with significant climbing capabilities.

I conclude this document with perspectives on what I call situated robotics, that is robots not taken in isolation but embedded in a sensorized environment shared with humans.

Keywords: Bayesian modeling, robotics, perception, state estimation, path planning.