



HAL
open science

Modèle bayésien pour l'analyse multimodale d'environnements dynamiques et encombrés : Application à l'assistance à la conduite en milieu urbain

Christophe Coue

► **To cite this version:**

Christophe Coue. Modèle bayésien pour l'analyse multimodale d'environnements dynamiques et encombrés : Application à l'assistance à la conduite en milieu urbain. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 2003. Français. NNT: . tel-00005527

HAL Id: tel-00005527

<https://theses.hal.science/tel-00005527v1>

Submitted on 5 Apr 2004

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.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

| / / / / / / / / / |

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : « Imagerie, Vision, Robotique »

préparée au laboratoire GRAVIR et l'INRIA Rhône-Alpes, dans le cadre de l'École
Doctorale « Mathématiques, Sciences et Technologies de l'Information,
Informatique »

présentée et soutenue publiquement par

Christophe COUÉ

le 03/12/2003

Titre :

**Modèle bayésien pour l'analyse multimodale
d'environnements dynamiques et encombrés :
Application à l'assistance à la conduite en milieu
urbain**

Directeur de Thèse :

Pierre BESSIÈRE

Composition du jury :

M.	Augustin Lux	Président
M.	Roland Siegwart	Rapporteur
M.	Philippe Martinet	Rapporteur
M.	Michel Parent	Examineur
M.	Pierre Bessière	Directeur de thèse
M.	Thierry Fraichard	Co-directeur de thèse

Remerciements

Je souhaite tout d'abord remercier Michel Parent pour avoir accepté de financer cette thèse, au sein du projet européen Carsense¹.

Je souhaite bien sûr remercier Pierre Bessière et Thierry Fraichard, qui ont co-encadré cette thèse. Leurs points de vue se sont montrés très complémentaires tout au long de ces trois années.

Je remercie également Roland Siewart et Philippe Martinet, mes rapporteurs, pour leur temps passé à travailler sur ce document. Je remercie également Augustin Lux pour m'avoir fait l'honneur de présider mon jury de thèse.

J'adresse également mes remerciements à Claude Puech et Roger Mohr, directeurs successifs du laboratoire GRAVIR au sein duquel j'ai effectué ce travail dans d'excellentes conditions. Je joins bien sûr à ces remerciements le personnel de GRAVIR et de L'INRIA Rhône-Alpes.

Je remercie également tous les membres du projet Sharp/Cybermove/eMotion, pour leurs compétences diverses qui font d'eux des adversaires redoutables tant pour les discussions scientifiques ou politiques qu'aux fléchettes.

Enfin un grand merci à Virginie, qui a eu seule à subir la face cachée de mon flegme apparent.

¹Projet IST-1999-12224 “*Sensing of Car Environment at Low Speed Driving*”

Résumé

La perception de l'environnement constitue un prérequis fondamental pour l'extension des systèmes d'assistance à la conduite actuels (comme l'ACC : *Advanced Cruise Control*) aux situations complexes de conduite induites par le milieu urbain.

La perception est classiquement basée sur des algorithmes de poursuite multicapteur et multipiste, dont le but est de fournir une estimation des caractéristiques dynamiques des différents acteurs (voitures, piétons, etc) de ces situations. Les difficultés posées par le milieu urbain (nombreuses occultations, apparitions et disparitions) rendent la poursuite multipiste peu fiable, et par conséquent le contrôle du véhicule impossible.

Selon nous, ce manque de fiabilité tient aux décisions prises au cours de la phase d'association pistes/observations de la poursuite, qui entraîne une perte d'information trop importante.

Nous proposons donc dans cette thèse une représentation alternative de l'environnement du véhicule, qui évite toute prise de décision prématurée en s'abstrayant de la notion de piste.

Notre méthode consiste dans le rapprochement de deux méthodes classiques. Les grilles d'occupation tout d'abord qui permettent de nous abstraire de la notion de piste. Dans notre cas, l'espace représenté par ces grilles est l'espace d'états des acteurs de la circulation (position et vitesse). Le filtre bayésien ensuite, qui permet de prendre en compte la dynamique de la scène. Nous appelons ce rapprochement le *filtre d'occupation bayésien*.

Le document illustre les qualités de notre représentation sur de nombreux exemples de manière incrémentale. Nous commençons par illustrer l'estimation statique de grille, lorsqu'un puis plusieurs capteurs fournissent une puis plusieurs observations. Par statique, nous entendons sans prise en compte de la dynamique temporelle de la scène. L'estimation dynamique, c'est à dire le filtre d'occupation bayésien en lui-même, est ensuite illustrée en suivant la même progression.

Enfin, l'intérêt de cette représentation dans le contexte automobile est montrée par l'implantation d'un algorithme d'évitement de collision sur le cycab. Le contrôle du cycab est alors effectué sans qu'aucune décision sur les objets n'ait été prise.

Le formalisme de *Programmation Bayésienne* est utilisé tout au long de cette thèse pour présenter les différents concepts, qu'ils soient classiques, comme les grilles d'occupation, ou nouveaux.

Mots-Clés : Assistance à la conduite automobile, grille d'occupation, filtres bayésiens, poursuite multi-cibles multi-capteurs.

Table des matières

Table des matières	i
Table des figures	v
Liste des tables	vii
Notations	xi
1 Introduction	1
1.1 Contexte : assistance à la conduite automobile	1
1.2 Les difficultés : perception et représentation de l'environnement	2
1.3 Contribution : le <i>filtre d'occupation bayésien</i>	3
1.4 Plan de lecture	4
2 Programmation bayésienne : concepts de base	7
2.1 Définitions fondamentales	8
2.1.1 Probabilité d'une proposition logique	8
2.1.2 Règles de calcul	8
2.1.3 Autres définitions et notations utiles	8
2.2 Méthode	10
2.2.1 Description (définition)	11
2.2.2 Spécification	11
2.2.3 Identification	12
2.2.4 Utilisation	12
2.2.5 Exemple : la fusion de capteurs	15
2.2.6 Conclusion sur la programmation bayésienne	18
3 Grille d'occupation : concepts de base	21
3.1 Grille d'occupation	21
3.1.1 Définition	21
3.1.2 Hypothèse d'indépendance des cellules	22
3.1.3 Utilisation des grilles d'occupation	22
3.2 Estimation d'une grille d'occupation à partir d'une observation capteur	23
3.2.1 Programme bayésien	24

3.2.2	Modélisation d'un capteur télémétrique	26
3.2.3	Illustration	31
3.3	Fusion de capteurs dans les grilles d'occupation	35
3.3.1	Utilisation de plusieurs capteurs : intérêts et difficultés	35
3.3.2	Programme bayésien	36
3.3.3	Illustration	39
3.3.4	Modélisation des capteurs : décalage temporel	43
3.4	Conclusion	45
4	Estimation des grilles d'occupation : cas multiobjet	47
4.1	Estimation d'une grille d'occupation à partir des observations d'un capteur	47
4.1.1	Programme bayésien	47
4.1.2	Illustration	51
4.2	Fusion de capteurs	52
4.2.1	Programme bayésien	52
4.2.2	Illustration	56
4.3	Conclusion sur l'estimation des grilles d'occupation	61
4.4	Limites de l'estimation statique des grilles	61
5	Le filtre d'occupation bayésien	65
5.1	Le filtre bayésien	65
5.1.1	Programme bayésien	67
5.1.2	Remarque : modèle dynamique incluant des actions du véhicule	71
5.1.3	Les implantations classiques du filtre bayésien	73
5.2	Grille d'occupation et filtres bayésien	77
5.2.1	Programme bayésien	77
5.2.2	Approximation de l'intégrale lors de la prédiction	80
5.2.3	Illustration	82
5.3	Fusion de capteurs	86
5.3.1	Programme bayésien	86
5.3.2	Illustration	89
5.4	Conclusion	90
6	Le filtre d'occupation bayésien : cas multiobjet	91
6.1	Poursuite multicible : problèmes spécifiques et solutions classiques	91
6.1.1	Algorithmes classiques d'association de données	92
6.1.2	Extension au cas multicapteur	97
6.2	Extension de notre méthode au cas multiobjet	99
6.2.1	Programme bayésien	99
6.2.2	Illustrations	102
6.3	Fusion de capteurs	108
6.4	Discussion et conclusion sur le filtre d'occupation bayésien	109

7	Application : évitement de collision	111
7.1	Plate-forme expérimentale : le cycab	111
7.2	L'application	112
7.2.1	Clustering de la grille d'occupation	113
7.2.2	Contrôle du cycab : programme bayésien	113
7.2.3	Résultats expérimentaux	117
7.3	Conclusion sur l'évitement de collision	123
8	Perspectives et conclusion	125
8.1	Conclusion	125
8.2	Perspectives	126
	Bibliographie	127
A	Closest Point of Approach	133

Table des figures

1.1	Éléments de base d'un algorithme classique de poursuite multipiste [BP00]. Les phases inscrites dans des cadres en pointillés sont spécifiques à la poursuite multipiste.	2
2.1	Exemple de programme bayésien : fusion de capteurs de luminosité.	15
2.2	Valeur moyenne de la réponse d'un capteur de luminosité en fonction de la distance D et de l'orientation θ de la source lumineuse.	16
2.3	Résultat de la fusion pour $\theta = 10$ (au centre). Chacun des huit diagrammes périphériques montre le résultat de l'estimation à partir de la mesure d'une unique capteur.	19
2.4	Structure générale d'un programme bayésien.	19
3.1	Exemple de grille d'occupation obtenue sur simulateur [Thr98]. La hauteur d'un point représente la probabilité d'occupation de la cellule. Dans cet exemple, l'environnement du robot mesure 105 mètres par 63.	22
3.2	Estimation de l'état d'une cellule : programme bayésien.	25
3.3	Modèle capteur : programme bayésien.	26
3.4	Exemple sur un cas à une dimension (capteur de distance uniquement) de forme paramétrique pour $P(Z [D_X = 0] [E_X = 1] [X = 10])$	29
3.5	Exemple d'estimation de grille d'occupation à une dimension à partir d'une mesure capteur.	30
3.6	Champ de vue du capteur (gris foncé), et zone couverte par la grille d'occupation (gris clair). Ces deux zones ne se superposent pas. Le capteur est matérialisé par le disque noir.	32
3.7	Estimation statique d'une grille d'occupation 2d : programme bayésien.	32
3.8	Grille d'occupation quand aucun objet n'est détecté. La couleur indique la valeur de la probabilité $P([E_{x,y} = 1] \Phi x y)$	34
3.9	Grilles d'occupations, un objet ayant été détecté, pour différentes précisions du capteur. La couleur indique la valeur de la probabilité $P([E_{x,y} = 1] \rho \theta x y)$	34
3.10	Estimation de l'état d'une cellule à partir des observations de plusieurs capteurs : programme bayésien.	37
3.11	Champ de vue du deuxième capteur (foncé), ajouté à la situation décrite dans la figure 3.6.	39

3.12	Illustration : programme bayésien.	39
3.13	Premier exemple : pas d'objet dans les champs de vue des capteurs.	40
3.14	La cible est détectée par les deux capteurs.	40
3.15	Le capteur 1 manque la cible. La probabilité de non détection du capteur 1 et la probabilité de fausse alarme du capteur 2 sont du même ordre de grandeur.	41
3.16	Le capteur 1 manque la cible. La probabilité de non détection du capteur 1 est très supérieure à la probabilité de fausse alarme du capteur 2.	42
3.17	Le capteur 1 manque la cible. La probabilité de non détection du capteur 1 est très inférieure à la probabilité de fausse alarme du capteur 2.	42
4.1	Estimation d'une grille d'occupation à partir des observations d'un capteur : programme bayésien.	50
4.2	Champ de vue du capteur (gris foncé), et zone couverte par la grille d'occupation (gris clair).	50
4.3	Exemple d'estimation d'une grille d'occupation 2d à partir des observations d'un capteur : programme bayésien.	51
4.4	Résultat de l'estimation d'une grille 2d à partir de plusieurs observations dans le cas mono-capteur.	51
4.5	Estimation d'une grille d'occupation à partir des observations de plusieurs capteurs : programme bayésien.	55
4.6	Champ de vue du deuxième capteur (foncé), ajouté à la situation décrite dans la figure 4.2.	55
4.7	Estimation d'une grille d'occupation à partir des observations de plusieurs capteurs dans le cas multiobjet : programme bayésien.	56
4.8	Premier exemple d'association et d'estimation.	57
4.9	Positions des objets dans l'expérience destinée à illustrer la sensibilité de l'association et de l'estimation à la précision des capteurs.	57
4.10	Résultats d'association et d'estimation pour différentes précisions angulaires des capteurs.	58
4.11	Position des cinq objets dans l'environnement du véhicule.	60
4.12	Résultats de l'association (à gauche) et de l'estimation de la grille d'occupation.	60
4.13	Exemple de scène dynamique. L'estimation statique de la grille ne permet pas de gérer les occultations.	62
5.1	Le filtre bayésien vu comme une boucle prédiction-estimation.	66
5.2	Filtre bayésien : programme.	71
5.3	Filtre Bayésien incluant des actions du véhicule portant le capteur : programme bayésien	72
5.4	Le filtre d'occupation bayésien monocapteur : programme bayésien	79
5.5	Algorithme d'approximation de l'intégrale dans le filtre d'occupation bayésien.	81
5.6	Premier exemple d'estimation de grille d'occupation incluant une phase de prédiction.	83

5.7	Autre exemple d'estimation, le cycab avançant vers une cible immobile. . .	84
5.8	Le filtre d'occupation bayésien multicapteur : programme bayésien	87
5.9	Exemple d'estimation dynamique de grille d'occupation dans le cas multicapteur. Le cycab avance vers une cible immobile.	88
6.1	Phases d'un algorithme classique de poursuite multipiste [BP00]. Les phases inscrites dans des cadres en pointillés sont spécifiques à la poursuite multiobjet.	92
6.2	Exemple de situation conflictuelle. P1 et P2 représentent les états prédits de deux objets poursuivis. Les cercles représentent les fenêtres de validations associées à ces prédictions. O1, O2 et O3 représentent les dernières observations capteurs.	94
6.3	Architecture de poursuite multicapteur « niveau capteur ». Chaque capteur fait sa propre poursuite multiobjet, les pistes des différents capteurs sont ensuite fusionnées.	97
6.4	Architecture de poursuite multicapteur « globale ». Les observations capteurs sont directement passées au niveau global.	98
6.5	Estimation dynamique des grilles d'occupation dans le cas monocapteur et multiobjet : programme bayésien.	101
6.6	Exemple de scène dynamique. L'estimation statique de la grille ne permet pas de gérer les occultations.	102
6.7	Exemple dynamique de grille d'occupation. Un premier objet cache temporairement un second objet au capteur.	103
6.8	Premier exemple d'apparition d'objet.	106
6.9	Deuxième exemple d'apparition d'objet.	106
6.10	Estimation dynamique des grilles d'occupation dans le cas multicapteur et multiobjet : programme bayésien.	108
7.1	Le cycab de l'INRIA Rhône-Alpes.	111
7.2	Contrôle du Cycab : programme bayésien.	116
7.3	Un piéton imprudent traverse devant le cycab!	117
7.4	Observations du capteurs lors de l'évitement de piéton à trois instants différents.	118
7.5	En haut, profil de la vitesse consigne du Cycab lors de l'évitement du piéton. En bas, distance entre le piéton et le Cycab au cours de cette même expérience.	118
7.6	En haut, profil de la vitesse consigne du Cycab lors d'un suivi de véhicule. En bas, distance entre le véhicule et le Cycab au cours de cette même expérience.	119
7.7	En haut, profil de la vitesse consigne du Cycab lors d'un Stop & Go. En bas, distance entre le véhicule et le Cycab au cours de cette même expérience.	120
7.8	Exemple de situation incluant une occultation : description du scénario. . .	121
7.9	Profils de vitesse du cycab lors de l'évitement de piéton incluant une occultation	122
A.1	CPA pour deux objets mobiles dans un plan à vitesse constante.	133

Liste des tableaux

6.1	Hypothèses d'associations possibles pour l'exemple présenté figure 6.2. N signifie que l'observation est considérée comme provenant d'une nouvelle piste.	95
6.2	Principaux avantages et inconvénients des architectures globales et niveau capteur pour la poursuite multicapteur.	98
8.1	Evolution du nombre de cellules dans la grille et du temps de calcul en fonction de l'espace représenté par la grille. Il suffit de double la taille de la grille selon chaque dimension pour que le nombre de cellules explose et que le temps de calcul devienne incompatible avec le contrôle d'un robot. . . .	126

Notations et conventions

Tout au long de ce manuscrit, les variables aléatoires sont notées par des lettres majuscules italiques. Leurs réalisations sont notées soit par des lettres minuscules, soit par une notation entre crochets, par exemple :

$$[Z = z]$$

signifie « la variable aléatoire Z prend pour valeur z ».

Les fonctions, et en particulier les distributions de probabilité sur des variables discrètes et les densités de probabilité sur des variables continues sont notées par des lettres droites.

La notation :

$$Z_{s,i}^k$$

désigne l'observation i du capteur s à l'instant k . Les différents éléments sont omis si le contexte le justifie. Par exemple, le numéro s du capteur n'est pas mentionné dans une partie traitant d'un cas monocapteur.

Afin d'alléger l'écriture des distributions de probabilité, nous définissons un ensemble de notation pour les conjonctions des variables représentant les observations des capteurs. Ainsi :

- Z_s^k représente l'ensemble des différentes observations du capteur s à l'instant k ;
- $Z_s^{l:k}$ représente l'ensemble des observations du capteur s depuis l'instant l jusqu'à l'instant k ;
- $Z_{1:S}^k$ représente l'ensemble des observations des différents capteurs à l'instant k ;
- $Z_{1:S}^{l:k}$ représente l'ensemble des observations des différents capteurs depuis l'instant l jusqu'à l'instant k .

Les différents éléments peuvent également être omis si le contexte le justifie. Le caractère calligraphique peut ne pas être utilisé quand on suppose que chaque capteur retourne une unique observation.

Chapitre 1

Introduction

1.1 Contexte : assistance à la conduite automobile

L'assistance à la conduite automobile est un thème à la mode. Les gouvernements y voient un moyen de réduire « l'insécurité routière », en palliant les déficiences et les erreurs des conducteurs. Pour les constructeurs automobiles, c'est un instrument de marketing non négligeable. Ainsi, toutes les voitures récentes sont équipées de systèmes électroniques tels le répartiteur électronique de freinage¹, l'amplificateur de freinage d'urgence², le correcteur électronique de trajectoire³, les systèmes d'aide à la navigation, etc.

Un autre système électronique plus orienté assistance est commercialisé depuis 1999. Il s'agit du « régulateur de vitesse adaptatif »⁴. Ce système est capable de détecter grâce à un radar ou un laser [WDH⁺00] le véhicule précédent, d'en estimer la vitesse et la distance, et de prendre en partie le contrôle du véhicule conduit (en bloquant l'accélérateur voire même en actionnant le frein) afin de maintenir la distance de sécurité. De tels systèmes sont aujourd'hui disponibles en option sur les véhicules haut-de-gamme de tous les constructeurs automobiles.

L'ACC ne peut être utilisé sans danger que sur les autoroutes, dans des conditions de trafic fluide [MMDB00]. Les situations de conduite (dépassement, insertion) sont simples et les acteurs (automobiles, camions) bien identifiés. Pour que ces systèmes soient réellement utilisables, il est nécessaire d'étendre leur champ d'opération aux conditions de trafic plus dense et à d'autres environnements, par exemple aux environnements urbains. Ces environnements présentent des situations beaucoup plus complexes (intersections, virages serrés, etc.) ainsi que des acteurs plus variés, plus vulnérables et au comportement imprévisible (piétons, vélos, etc.). Le développement d'un système d'assistance à la conduite pour le milieu urbain est donc un problème complexe.

¹EBV : *Electronic Brake force Distribution*

²BAS : *Brake ASsist*

³ESP : *Electronic Stability Program*

⁴ACC : *Adaptive Cruise Control*

1.2 Les difficultés : perception et représentation de l'environnement

Deux pré-requis majeurs pour le développement d'un système d'assistance à la conduite en milieu urbain sont :

- le développement d'un système de perception capable de fournir toutes les informations nécessaires à la tâche de conduite. En particulier, les caractéristiques dynamiques (position, vitesse) des acteurs de la circulation doivent être estimées de manière précise et robuste ;
- la représentation des informations fournies par le système de perception sous une forme exploitable et n'entraînant aucune perte d'information. Cette représentation est fortement liée au système de perception.

L'estimation des caractéristiques dynamiques des acteurs de la circulation peut être vue comme un problème classique de poursuite multipiste. Afin d'augmenter la robustesse de la poursuite, il est envisageable d'utiliser plusieurs capteurs.

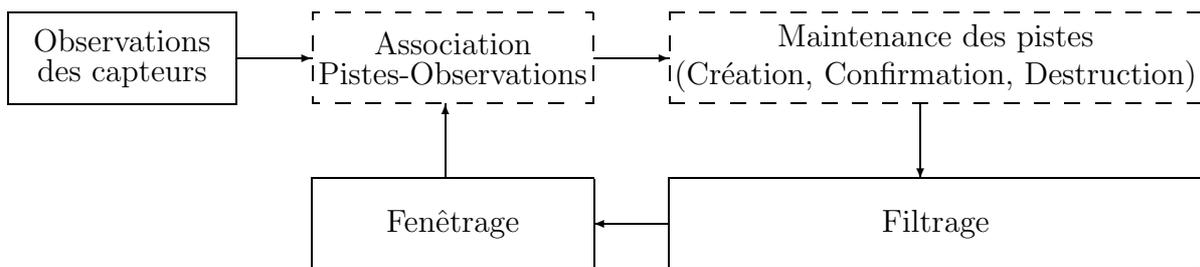


FIG. 1.1 – Éléments de base d'un algorithme classique de poursuite multipiste [BP00]. Les phases inscrites dans des cadres en pointillés sont spécifiques à la poursuite multipiste.

Le but de la poursuite multipiste est de maintenir une liste de *pistes*, c'est à dire une liste d'objets supposés présents dans l'environnement du véhicule. Pour former et maintenir ces pistes, le système de poursuite se base bien entendu sur des *observations* fournies par le ou les capteurs dont il dispose.

La figure 1.1 présente les éléments de base d'un algorithme classique de poursuite multipiste [BP00]. Chaque fois qu'une nouvelle série d'observations des capteurs est disponible, le système doit d'abord décider d'une association entre ces nouvelles mesures et les pistes existantes. C'est la phase d'association observations-pistes. Au vu de cette association, la phase de maintenance des pistes doit décider :

- de la confirmation de pistes existantes qui ont été associées à une nouvelle observation capteur ;
- de la création de nouvelles pistes, pour chaque observation capteur qui n'a été associée à aucune piste existante. Cette décision doit tenir compte du fait que les capteurs

peuvent générer des *fausses alarmes*, c'est à dire des observations ne provenant pas d'un objet existant mais d'une défaillance du capteur ;

- de la suppression de pistes existantes, pour les pistes qui ne sont associées à aucune nouvelle observation. Cette décision doit tenir compte du fait que les capteurs peuvent avoir manqué un objet existant, soit parce que le capteur est défaillant soit parce que l'objet en question est masqué par un autre objet.

La phase de filtrage n'est pas spécifique à la poursuite multipiste, puisqu'elle consiste pour chaque piste en la mise à jour et la prédiction de l'état de l'objet. Ceci est généralement fait à l'aide d'un filtre de Kalman.

Enfin la dernière phase est une phase de fenêtrage. Cette phase de fenêtrage a pour but de simplifier la phase d'association : une observation capteur pourra être considérée comme pouvant provenir de cette piste si sa distance à l'état prédit de l'objet est inférieure à un seuil. Déterminer ce seuil n'est pas trivial. La « fenêtrage de validation » doit être suffisamment grande pour qu'un impact provenant d'une piste s'y trouve et suffisamment petite pour pouvoir efficacement filtrer les impacts.

Les deux premières phases, association et maintenance des pistes, constituent la grosse difficulté de la poursuite multipiste. Ces deux phases sont en général résumées sous le terme *d'association de données*.

Si le problème de l'association de données est largement étudié [BP00][BSF88][BSL95], l'environnement urbain constitue un défi pour ce domaine. En effet, compte tenu de la grande dynamique des objets, le système de poursuite doit être capable de gérer de très nombreuses apparitions et disparitions d'objets, et donc un nombre d'objets très variable. De plus, les occultations entre objets sont très fréquentes. Les algorithmes de poursuite multipiste actuels sont donc peu fiables dans un contexte d'assistance à la conduite en milieu urbain.

À ce problème de poursuite s'ajoute selon nous un problème de représentation de l'environnement. En effet, la sortie - et donc les informations dont disposera le système d'assistance - d'un algorithme de poursuite multipiste est constituée d'une liste de pistes. Une quantité importante d'informations est donc perdue : par exemple les zones de l'environnement qui sont masquées.

1.3 Contribution : le *filtre d'occupation bayésien*

Nous avons choisi d'aborder le problème de la représentation de l'environnement de manière totalement transversale, en nous abstrayant totalement des notions d'objet et de piste. Cette abstraction a pour but d'éviter toute prise de décision prématurée, en particulier sur l'association et la maintenance des pistes.

Notre principale contribution porte donc sur la définition d'un modèle de représentation d'environnements dynamiques et encombrés, que nous appelons *filtre d'occupation bayésien*⁵.

Pour nous abstraire des notions de piste et d'objet, nous nous basons sur une utilisation originale des *grilles d'occupation* [Elf89]. Dans cette représentation, l'environnement

⁵en anglais : *Bayesian Occupancy Filter*

est découpé en cellules. La probabilité pour chaque cellule d'être occupée par un objet quelconque est estimée à partir des observations des capteurs. Dans notre cas, les grilles sont définies dans un repère égocentré (c'est à dire centré sur le véhicule) représentant l'espace d'état des objets présents dans l'environnement (position et vitesse relative). L'espace représenté par ces grilles constitue une originalité par rapport aux utilisations classiques des grilles d'occupation (position seulement).

Pour tenir compte de la dynamique de la scène, nous nous basons sur la méthode classique du *filtre bayésien*. Dans notre cas, le modèle dynamique de ce filtre n'est bien sûr pas associé à une piste, mais à une cellule de la grille.

Notre contribution essentielle consiste donc dans le rapprochement de deux méthodes classiques : la grille d'occupation et le filtre bayésien, la grille d'occupation étant utilisée sous une forme et dans un contexte inhabituel. Cette contribution inclut le développement d'un algorithme d'approximation spécifique pour répondre au problème du coût calculatoire de ces méthodes.

Pour illustrer la faisabilité calculatoire et la pertinence du filtre d'occupation bayésien dans le cadre de l'assistance à la conduite automobile, nous définissons également une implantation originale d'évitement de collision sur un robot de type voiture. La principale originalité de cette méthode est de ne prendre aucune décision prématurée sur la notion d'objet - leur nombre, leurs caractéristiques - autour du véhicule.

1.4 Plan de lecture

Le formalisme de *Programmation Bayésienne* est utilisé tout au long de cette thèse pour présenter nos modélisations et les calculs induits par ces modélisations. Le chapitre 2 introduit ce formalisme, ainsi que les notions mathématiques nécessaires.

Le chapitre 3 présente ensuite les grilles d'occupation. Nous commençons par définir cette notion, puis nous en présentons les utilisations classiques. Ensuite, la méthode d'estimation d'une grille statique est présentée lorsqu'un capteur fournit une unique observation. Par statique, nous entendons qu'aucune considération sur la dynamique de la scène n'est prise en compte. Enfin, nous présentons l'extension de cette estimation au cas multicapteur, chaque capteur fournissant une unique observation.

Le chapitre 4 présente l'extension de la méthode d'estimation statique de la grille d'occupation lorsque les capteurs fournissent plusieurs observations, ce qui est le cas lorsque plusieurs objets sont présents dans l'environnement du véhicule.

Du fait de la dynamique de l'environnement qui implique des occultations entre objets, des apparitions et des disparitions d'objets, l'estimation statique des grilles n'est pas satisfaisante. Le chapitre 5 présente notre méthode de filtrage des grilles d'occupation : le filtre d'occupation bayésien. Il commence par présenter la notion de filtre bayésien et ses implantations classiques. Il montre ensuite comment ce filtre peut être appliqué pour l'estimation de la grille d'occupation en deux phases : prédiction et estimation. Nous revenons dans ce chapitre au cas dans lequel un ou plusieurs capteurs fournissent à chaque instant une unique observation.

Le chapitre 6 présente le filtre d'occupation bayésien lorsque un ou plusieurs capteurs fournissent plusieurs observations. Ce chapitre commence par une présentation des méthodes classiques de poursuite multipiste.

Le chapitre 7 présente une application d'évitement de collision sur le véhicule de démonstration de l'INRIA Rhône-Alpes, le cycab. Cette application est bien sûr basée sur l'estimation dynamique de grille d'occupation.

Enfin le dernier chapitre présente nos conclusions sur ce travail et les pistes de recherche envisagées.

Chapitre 2

Programmation bayésienne : concepts de base

*“It’s - it’s a variable.” Kaplan was shaking, white-lipped and pale.
“Something from which no inference can be made. The man from
the past. The machines can’t deal with him. The variable man!”*

PHILIP K. DICK, *The Variable Man*

Nous présentons dans ce chapitre le minimum théorique et mathématique nécessaire pour la compréhension des expériences présentées dans les chapitres suivants. Notre objectif est de présenter brièvement les principes du calcul bayésien comme une méthode de modélisation et d’inférence permettant de prendre en compte les incertitudes. Pour une présentation détaillée, nous renvoyons aux travaux de Pierre Bessière [BDL⁺98a][BDL⁺98b] et d’Olivier Lebeltel [Leb99] [LBDM04].

Nous nous plaçons plus précisément dans le cadre d’une théorie du raisonnement probabiliste appelée *Probability as Logic* (PaL) proposée par le physicien E.T.Jaynes [Jay03]. Cette théorie considère le calcul des probabilités comme une généralisation de la logique formelle (booléenne). Elle permet d’étendre la logique à des propositions dont la vérité n’est pas connue avec certitude et de formaliser la notion de raisonnement plausible. Dans cette théorie, la notion de probabilité est utilisée non pas dans son sens *fréquentiste* où elle caractérise la *fréquence* d’une mesure physique donnée, mais plutôt dans son sens *subjectiviste* où elle exprime un *état de connaissance* représentant les *informations* dont on dispose sur le phénomène en question. Cette utilisation des probabilités découle directement de la formalisation de la notion intuitive de *plausibilité* proposée par Cox [Cox46][Cox61]. En se fixant un ensemble de desiderata définissant la notion de plausibilité pour une proposition logique, Cox montre que l’unique façon de manipuler cette notion (en restant fidèle aux desiderata de départ) est donnée par la théorie des probabilités. La notion de plausibilité est ainsi formalisée par la notion mathématique bien définie de *probabilité*.

2.1 Définitions fondamentales

2.1.1 Probabilité d'une proposition logique

Une proposition logique est un énoncé qui peut être soit vrai soit faux. Nous définissons la plausibilité (probabilité) $P(\mathcal{A})$ d'une proposition \mathcal{A} comme le degré de certitude accordé à sa véracité.

En réalité, toute probabilité d'une proposition \mathcal{A} ne peut être donnée qu'au vu d'un ensemble de *connaissances préalables* que l'on dénotera c . Il est alors plus convenable d'expliciter dans les notations les connaissances c qui ont permis d'assigner une valeur $P_{\mathcal{A}}$ à une proposition \mathcal{A} en écrivant $P(\mathcal{A}|c) = P_{\mathcal{A}}$. Toutefois, dans le but d'alléger les notations, nous nous dispensons d'écrire ces connaissances c dans les formules que nous manipulons dans le reste de ce document. En effet à un instant donné nous ne nous intéressons qu'à un seul ensemble de connaissances préalables c .

Nous utiliserons, dans ce qui suit, la notation $\mathcal{A}\mathcal{B}$ pour la conjonction de \mathcal{A} et \mathcal{B} , $\mathcal{A} + \mathcal{B}$ pour leur disjonction et $\neg\mathcal{A}$ et $\neg\mathcal{B}$ pour leurs négations respectives.

2.1.2 Règles de calcul

À partir des desiderata définissant le raisonnement plausible, Cox montre que ce raisonnement doit utiliser 2 règles fondamentales qui sont à la base de toute inférence probabiliste. Étant données les propositions logiques \mathcal{A} et \mathcal{B} , ces deux règles sont la règle du produit et la règle de normalisation.

2.1.2.1 Règle du produit

La règle du produit donne la probabilité d'une conjonction :

$$P(\mathcal{A} \mathcal{B}) = P(\mathcal{A})P(\mathcal{B} | \mathcal{A}) = P(\mathcal{B})P(\mathcal{A} | \mathcal{B}). \quad (2.1)$$

2.1.2.2 Règle de normalisation

La règle de normalisation exprime le fait que la somme des probabilités d'une proposition et de sa négation est égale à 1 :

$$P(\mathcal{A}) + P(\neg\mathcal{A}) = 1. \quad (2.2)$$

2.1.3 Autres définitions et notations utiles

2.1.3.1 Probabilité d'une disjonction

L'utilisation des règles (2.1) et (2.2) permet d'écrire la probabilité d'une disjonction comme suit :

$$P(\mathcal{A} + \mathcal{B}) = P(\mathcal{A}) + P(\mathcal{B}) - P(\mathcal{A} \mathcal{B}). \quad (2.3)$$

Cette dernière règle s'écrit dans le cas où A et B sont indépendantes :

$$P(\mathcal{A} + \mathcal{B}) = P(\mathcal{A}) + P(\mathcal{B}). \quad (2.4)$$

2.1.3.2 Variable discrète - distribution de probabilité discrète

Jusque ici, seule la probabilité d'une proposition logique a été définie. Supposons que l'on s'intéresse à assigner des probabilités aux différentes valeurs numériques d'une variable discrète X . Nous devons dans ce cas définir un ensemble de propositions logiques mutuellement exclusives $\mathcal{A}_n \equiv "X = n"$, où n prend toutes les valeurs possibles de X , et considérer les probabilités $P(\mathcal{A}_n) = f(n)$. La fonction f représente une *distribution de probabilité discrète* sur la plage de variation de la variable X .

Nous utilisons la notation $P(X)$ pour désigner une distribution de probabilité sur une variable discrète X .

Les deux règles (2.1) et (2.2) restent valides pour les cas des variables et s'écrivent respectivement pour deux variables X et Y comme suit :

$$P(X Y) = P(X)P(Y|X) = P(Y)P(X|Y)$$

$$\sum_X P(X) = 1.$$

2.1.3.3 Variable continue - densité de probabilité

Pour parler de probabilité pour une variable continue X en utilisant les résultats obtenus pour les propositions logiques, nous reprenons le raisonnement donné par Jaynes :

Commençons par définir les deux propositions suivantes :

$$\mathcal{F}_1 \equiv (X \leq q) \quad (2.5)$$

$$\mathcal{F}_2 \equiv (X > q) \quad (2.6)$$

Ces deux propositions sont exhaustives et mutuellement exclusives. La probabilité de \mathcal{F}_1 va dépendre de q en définissant une fonction G telle que :

$$P(\mathcal{F}_1) = G(q).$$

Cette fonction G est bien évidemment croissante.

Intéressons nous maintenant à la proposition $\mathcal{W} \equiv (a < X \leq b)$ (la valeur de X est dans l'intervalle $]a, b]$). Notons \mathcal{A} et \mathcal{B} les propositions suivantes :

$$\mathcal{A} \equiv (X \leq a); \quad (2.7)$$

$$\mathcal{B} \equiv (X \leq b). \quad (2.8)$$

Nous avons l'égalité $\mathcal{B} = \mathcal{A} + \mathcal{W}$. De plus, comme \mathcal{A} et \mathcal{W} sont mutuellement exclusives, l'application de la règle de la somme permet d'écrire :

$$P(\mathcal{B}) = P(\mathcal{A}) + P(\mathcal{W}),$$

ce qui donne finalement :

$$P(a < X \leq b) = p(\mathcal{W}) = G(b) - G(a).$$

Si G est différentiable, nous pourrions écrire :

$$P(a < X \leq b) = \int_a^b g(X)dX$$

avec $g(X) = \frac{dG(X)}{dX}$.

La fonction g est appelée *densité de probabilité* ou bien *distribution de probabilité* sur la variable X . Nous l'appellerons dans le reste de ce document *distribution de probabilité* et nous utiliserons la notation $P(X)$ pour dénoter la distribution sur une variable continue.

Les deux règles fondamentales (2.1) et (2.2) restent valides pour les cas des variables continues et s'écrivent respectivement pour deux variables X et Y comme suit :

$$P(X Y) = P(X)P(Y | X) = P(Y)P(X | Y);$$

$$\int P(X)dX = 1.$$

La distribution $P(X Y)$ est appelée la *distribution conjointe* de X et Y .

2.1.3.4 Marginalisation - Distribution marginale

Soit X et Y deux variables numériques. On appelle *distribution marginale* de X par rapport à Y la distribution de probabilité :

- $P(X) = \sum_Y P(X Y) = \sum_Y P(Y)P(X | Y)$ dans le cas où Y est discrète ;
- $P(X) = \int P(X Y)dY = \int P(Y)P(X | Y)dY$ dans le cas où Y est continue.

2.1.3.5 Formule de Bayes

Une transformation mathématique directe de la règle du produit permet d'obtenir la formule dite de *Bayes*

$$P(X | Y) = \frac{P(X)P(Y | X)}{P(Y)} = \frac{P(X)P(Y | X)}{\sum_X P(X)P(Y | X)}.$$

2.2 Méthode

Dans cette section, nous présentons brièvement la méthode appelée « Programmation Bayésienne », qui sera utilisée dans le reste du document. Cette méthode se base sur un objet formel, la *description* [Leb99][LBDM04]. La définition d'un « programme bayésien » à l'aide d'une description se décompose en trois phases :

- la spécification des connaissances préalables ;
- l'identification des valeurs des paramètres des distributions de probabilité ;
- l'utilisation de la description.

Nous allons maintenant définir le concept de description et décrire plus précisément chacune de ces trois phases.

2.2.1 Description (définition)

Une description est dénotée formellement par la distribution de probabilité conjointe $P(V_1 \dots V_n | d c)$ d'un ensemble de variables V_1, \dots, V_n , déterminée au vu des connaissances préalables c spécifiées par le programmeur et d'un ensemble de données expérimentales d . Ainsi, l'ensemble des connaissances devant être fournies se trouve circonscrit par les connaissances préalables c et le jeu de données d .

2.2.2 Spécification

La phase de spécification est la partie la plus délicate du travail du programmeur. Au cours de cette phase, il doit énoncer clairement et *explicitement* les connaissances dont il est à l'origine et celles qui résultent d'un processus adaptatif dépendant d'un jeu particulier de données expérimentales. Ces connaissances se subdivisent en trois : le choix des variables pertinentes, l'expression des dépendances entre les variables retenues sous la forme d'un produit de distributions élémentaires, et enfin la forme paramétrique associée à chacune de ces distributions.

2.2.2.1 Connaissances préalables structurelles : le choix des variables pertinentes

Nous appelons *connaissances préalables structurelles* les connaissances permettant de définir l'ensemble des variables V_1, \dots, V_n pour la description. Toutes les autres variables sont ainsi supposées non pertinentes pour le problème considéré.

- En robotique, nous pouvons classer ces variables naturellement en trois sous-ensembles :
- les variables sensorielles extéroceptives et proprioceptives ;
 - les variables motrices ;
 - et enfin les variables internes , qui permettent de coder les états internes du robot.

2.2.2.2 Connaissances préalables de dépendance : le choix d'une décomposition de la distribution conjointe

Comme énoncé précédemment, la description sur les variables V_1, \dots, V_n a pour but la définition de la distribution conjointe $P(V_1 \dots V_n | d c)$. Cette forme mathématique est une distribution de probabilité sur n dimensions, qui n'est souvent pas facile à spécifier. La règle du produit (2.1) nous permet de décomposer cette expression, en l'exprimant sous forme de produit de distributions.

Prenons en exemple une distribution conjointe $P(X Y Z)$ de 3 variables X, Y et Z . L'application de la règle du produit permet d'écrire :

$$P(X Y Z) = P(Z)P(Y | Z)P(X | Y Z).$$

Cette seconde étape de la spécification permet également d'exprimer les relations de dépendance, ou d'indépendance, entre les variables. Ces indépendances permettent de réduire fortement les dimensions des termes apparaissant dans la décomposition.

Si l'on suppose dans notre exemple que les variables X et Y sont indépendantes sachant la valeur de Z , on aura :

$$P(X \ Y \ Z) = P(Z)P(Y \mid Z)P(X \mid Z).$$

2.2.2.3 Connaissances préalables d'observation : le choix des formes paramétriques

Il faut maintenant associer à chacun des termes apparaissant dans la décomposition choisie à l'étape précédente une forme paramétrique. Par ces choix, nous allons fournir des *a priori* sur les valeurs des distributions de probabilité et la manière dont ces valeurs seront modifiées par l'expérience. Ce dernier point est composé d'un ensemble de valeurs initiales pour les paramètres, et d'un mécanisme de mise à jour au vu de données expérimentales (ce mécanisme peut éventuellement être vide si l'on veut figer la distribution lors de la présente phase de spécification). Une description dans laquelle tous les termes sont ainsi fixés est appelée *spécification (description) a priori*.

Les formes paramétriques sont en général des lois de probabilité classiques, par exemple des lois uniformes ou des lois normales. Une *question* à une autre description peut également être utilisée comme programme probabiliste. Ce point sera illustré dans le chapitre suivant.

2.2.3 Identification

Les formes paramétriques peuvent contenir des paramètres libres, comme les moyennes ou écarts-type de distributions Gaussiennes. Il est nécessaire de fixer les valeurs numériques de ces paramètres pour achever notre description.

Pour obtenir les données expérimentales nécessaires à cet apprentissage, plusieurs méthodes sont disponibles. Par exemple, dans le contexte de la robotique mobile, des apprentissages par télé-opération [Dia99] ou par imitation d'un robot professeur [Ras02] ont été considérés.

La spécification des paramètres peut également être faite *a priori* par le programmeur sans considérer de données expérimentales. C'est ce qui sera fait dans cette thèse.

2.2.4 Utilisation

Au terme des phases de spécification et d'identification, nous disposons d'une description complètement définie. La phase d'utilisation va consister à mettre en œuvre les descriptions par le biais de questions probabilistes.

2.2.4.1 Question (définition)

Poser une question consiste à chercher la distribution de probabilité d'un certain nombre de variables \mathcal{E}_q de la description, connaissant les valeurs d'autres variables \mathcal{E}_c , et éventuellement ignorant les valeurs d'un troisième groupe de variables \mathcal{E}_i . Une question probabiliste

est donc n'importe quelle expression de la forme :

$$P(V_k \dots V_l | v_m \dots v_n), \quad (2.9)$$

où $\mathcal{E}_q = \{V_k, \dots, V_l\} \neq \emptyset$, $\mathcal{E}_c = \{V_m, \dots, V_n\}$, et $\mathcal{E}_i = \{V_o, \dots, V_p\}$ est l'ensemble des variables n'apparaissant ni dans \mathcal{E}_q , ni dans \mathcal{E}_c . Ces trois ensembles doivent bien sûr former une partition de l'ensemble des variables considérées pour que la question ait un sens.

2.2.4.2 Inférence

La connaissance de la distribution conjointe $P(V_1 \dots V_n)$ et l'application des règles du produit (2.1) et de la marginalisation (2.2) nous permet de répondre à toute question de la forme (2.9). Appliquons d'abord la règle de Bayes :

$$P(V_k \dots V_l | v_m \dots v_n) = \frac{P(V_k \dots V_l v_m \dots v_n)}{P(v_m \dots v_n)}. \quad (2.10)$$

En appliquant maintenant la règle de marginalisation, nous pouvons exprimer le dénominateur et le numérateur de ce quotient en fonction de la distribution conjointe que l'on sait calculer :

$$P(V_k \dots V_l | v_m \dots v_n) = \frac{\sum_{V_o \dots V_p} P(V_k \dots V_l v_m \dots v_n V_o \dots V_p)}{\sum_{\substack{V_k \dots V_l \\ V_o \dots V_p}} P(V_k \dots V_l v_m \dots v_n V_o \dots V_p)}. \quad (2.11)$$

Du fait de la normalisation sur les variables $V_o \dots V_p$, cette méthode de résolution d'une question peut être coûteuse (l'inférence bayésienne en général est *NP-difficile* [Coo90]). Cependant différentes formes de simplification peuvent apparaître lors de l'inférence.

Ces simplifications peuvent être purement mathématiques. Nous en utiliserons deux en particulier dans ce document :

- si dans une somme portant sur une variable A , il existe des termes dans lesquels A n'apparaît pas, ce terme peut être sorti de la somme :

$$\sum_A (P(B)P(A)P(C | A)) = P(B) \times \sum_A (P(A)P(C | A)); \quad (2.12)$$

- si toutes les variables sur lesquelles porte une somme apparaissent en variable gauche dans un terme de la somme, cette somme est égale à 1 :

$$\sum_{A, B} P(A | C)P(B | AC) = 1. \quad (2.13)$$

Prenons par exemple le cas particulier qui survient lorsque la question probabiliste posée à une description est un terme de la décomposition. Par exemple, supposons qu'une description portant sur les variables X_1, X_2, X_3 choisisse la décomposition suivante :

$$P(X_1 X_2 X_3) = P(X_1)P(X_2 | X_1)P(X_3 | X_1).$$

Posons à cette description la question $P(X_2 | X_1)$. En appliquant le résultat de l'inférence donné par l'expression (2.11), nous obtenons :

$$P(X_2 | X_1) = \frac{\sum_{X_3} P(X_1)P(X_2 | X_1)P(X_3 | X_1)}{\sum_{X_2, X_3} P(X_1)P(X_2 | X_1)P(X_3 | X_1)}. \quad (2.14)$$

Grâce à la simplification (2.12), nous pouvons écrire :

$$P(X_2 | X_1) = \frac{P(X_1)P(X_2 | X_1)\sum_{X_3} P(X_3 | X_1)}{P(X_1)(\sum_{X_2} P(X_2 | X_1))(\sum_{X_3} P(X_3 | X_1))} \quad (2.15)$$

ce qui se simplifie grâce à (2.13) :

$$P(X_2 | X_1) = \frac{P(X_1)P(X_2 | X_1)}{P(X_1)} \quad (2.16)$$

$$= P(X_2 | X_1). \quad (2.17)$$

Ce résultat est rassurant : si la question est un terme qui apparaît dans la décomposition, donc un terme auquel une forme paramétrique connue est associée, l'inférence restitue ce terme. Le résultat de la question est donc donné par la forme paramétrique associée, sans avoir recours à aucun calcul.

Des simplifications symboliques plus complexes peuvent également apparaître dans l'inférence. Par exemple, dans le cas de sommes portant sur plusieurs variables, un ordonnancement correct des sommes peut permettre d'alléger les calculs [AM00].

Des simplifications peuvent également découler des connaissances préalables fournies lors de la phase de spécification, dans le cas d'utilisation de distributions uniformes ou de distributions dirac par exemple. Enfin l'utilisation du résultat de la question peut également permettre de simplifier l'inférence. Par exemple, si on ne s'intéresse qu'au maximum de la distribution de probabilité, il n'est pas nécessaire de calculer le dénominateur de l'expression (2.11).

Il a été développé dans l'équipe un moteur d'inférence probabiliste qui permet de répondre à toute question probabiliste. Ce moteur a pour ambition d'être capable d'automatiser les raisonnements probabilistes. Il peut être en cela comparé aux nombreux moteurs d'inférence qui permettent d'automatiser le raisonnement logique (tels Prolog, LogLisp ou Lolita). Pour une question donnée, l'expression correspondante est tout d'abord simplifiée symboliquement par le moteur, en utilisant les différentes formes de simplifications que nous venons de présenter. La valeur numérique de la question est ensuite estimée. Si on ne dispose pas du temps suffisant pour mener entièrement le calcul, des méthodes d'approximation des sommes (type Monte-Carlo [Nea93][Gew96]) peuvent être utilisées pour accélérer le calcul. Ceci se fait bien sûr au détriment de l'exactitude.

2.2.4.3 Décision

Le résultat de l'inférence fournit une distribution de probabilité sur les variables recherchées. Cette distribution de probabilité résume les connaissances préalables du programmeur sur le problème et les informations apportées par des observations capteurs par exemple.

Dans le cadre de la robotique, cette distribution de probabilité porte typiquement sur les variables motrices. Afin de contrôler le robot, il convient de choisir une valeur pour ces variables. Il s'agit donc d'un problème de décision. De nombreuses stratégies sont ici imaginables, mais les plus simples restent de choisir la valeur correspondant au maximum de probabilité, ou encore de tirer les valeurs aléatoirement selon la distribution obtenue.

2.2.5 Exemple : la fusion de capteurs

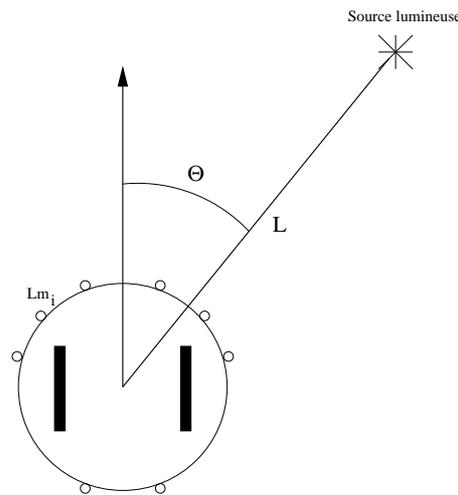


FIG. 2.1 – Exemple de programme bayésien : fusion de capteurs de luminosité.

Cet exemple reprend succinctement une expérience de la thèse d'Olivier Lebeltel ([Leb99]) sur le mini robot mobile Khépéra. Ce robot dispose de huit capteurs de luminosité. Le but de cette expérience est de déterminer la direction d'une source lumineuse, en fonction des réponses de ces huit capteurs (voir figure 2.1).

2.2.5.1 Spécification

Variables Les variables retenues sont :

- les huit variables correspondant aux huit capteurs de luminosité : Lm_0 à Lm_7 . Les valeurs de ces variables varient entre 0 (luminosité forte) et 511 (luminosité faible) ;
- la variable caractérisant la direction de la source lumineuse par rapport au robot : θ_L . Les valeurs de cette variable varient entre -180° et 180° , le pas de discrétisation étant de 10° ;
- la variable caractérisant la distance de la source lumineuse par rapport au robot : D . Les valeurs de cette variable varient entre 0 cm et 25 cm, le pas de discrétisation étant de 1 cm.

Ces variables définissent la distribution de probabilité conjointe :

$$P(Lm_0 \dots Lm_7 \theta D). \quad (2.18)$$

Décomposition La décomposition de la distribution de probabilité conjointe est à la charge du programmeur, est n'est en général pas facile. Dans l'exemple actuel, plusieurs milliers de décompositions de la distribution (2.18) sont possibles.

Le programmeur choisit donc une décomposition en fonction de certaines connaissances. Supposons dans notre exemple que nous sachions modéliser la réponse des capteurs de luminosité en fonction de la position de la source lumineuses, soit les distributions $P(Lm_i | \theta D)$. Le but de la décomposition va être de faire apparaître ces différentes distributions de probabilité

L'application successive de la règle du produit permet d'écrire la décomposition :

$$P(Lm_0 \dots Lm_7 | \theta D) = P(\theta D)P(Lm_0 | \theta D) \dots P(Lm_7 | Lm_6 \dots Lm_0 | \theta D) \quad (2.19)$$

On fait ici une hypothèse forte d'indépendance conditionnelle. Il est clair que les valeurs lues sur deux capteurs ne sont pas indépendantes. On suppose cependant que la dépendance entre deux capteurs est résumée dans la connaissance de la source, donc de θ et D . Ceci nous permet d'écrire :

$$P(Lm_i | Lm_{i-1} \dots Lm_0 | \theta D) = P(Lm_i | \theta D) \quad (2.20)$$

La décomposition finale est donc :

$$P(Lm_0 \dots Lm_7 | \theta D) = P(\theta D) \prod_{i=0}^7 P(Lm_i | \theta D). \quad (2.21)$$

Formes paramétriques On ne souhaite exprimer aucune connaissance *a priori* sur la position de la source lumineuse. La distribution $P(\theta D)$ est donc représentée par une loi uniforme.

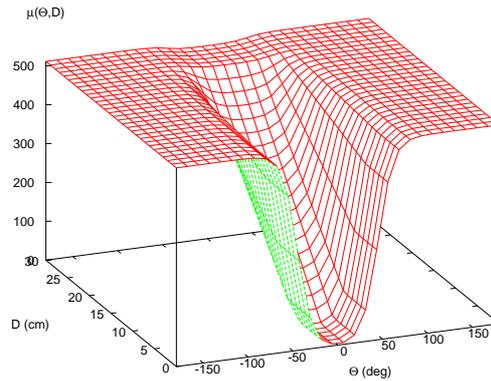


FIG. 2.2 – Valeur moyenne de la réponse d'un capteur de luminosité en fonction de la distance D et de l'orientation θ de la source lumineuse.

Les différentes $P(Lm_i | \theta D)$ représentent la réponse d'un capteur de luminosité en fonction de la position de la source lumineuse. Une famille de lois normales paramétrées

par la position de la source lumineuse et par la position du capteur sur le robot est choisie pour les représenter. La figure 2.2 présente la fonction permettant de calculer la moyenne de ces lois normales en fonction de la position de la source lumineuse. Sur cette figure, la position du capteur sur le robot n'est pas prise en compte. Cette fonction est choisie d'après la documentation des capteurs de luminosité. L'écart-type est quant à lui fixé à une valeur constante (20 unités de luminosité) indépendamment de la position de la source lumineuse.

Identification Il n'y a pas de paramètre libre, puisque le modèle capteur est connu, donc pas de phase d'identification.

Utilisation Pour estimer la direction de la source lumineuse en tenant compte des observations des huit capteurs de luminosité, nous posons la question :

$$P(\Theta \mid Lm_0 \dots Lm_7). \quad (2.22)$$

Pour cette question, le résultat de l'inférence donne :

$$P(\Theta \mid Lm_0 \dots Lm_7) = \frac{\sum_D P(\Theta \mid D \mid Lm_0 \dots Lm_7)}{\sum_{\Theta, D} P(\Theta \mid D \mid Lm_0 \dots Lm_7)} \quad (2.23)$$

$$= \frac{\sum_D \left(P(\Theta \mid D) \prod_{i=0}^7 P(Lm_i \mid \Theta \mid D) \right)}{\sum_{\Theta, D} \left(P(\Theta \mid D) \prod_{i=0}^7 P(Lm_i \mid \Theta \mid D) \right)} \quad (2.24)$$

Puisque nous avons choisi de représenter la distribution $P(\Theta \mid D)$ par une loi uniforme, nous pouvons simplifier ce terme. Le résultat de l'inférence devient donc :

$$P(\Theta \mid Lm_0 \dots Lm_7) = \frac{\sum_D \prod_{i=0}^7 P(Lm_i \mid \Theta \mid D)}{\sum_{\Theta, D} \prod_{i=0}^7 P(Lm_i \mid \Theta \mid D)} \quad (2.25)$$

$$= \frac{1}{\alpha} \sum_D \prod_{i=0}^7 P(Lm_i \mid \Theta \mid D), \quad (2.26)$$

où α est une constante de normalisation.

Afin de comparer les résultats de la fusion avec les résultats issus de la mesure d'un unique capteur, nous posons également à notre description les huit questions :

$$P(\Theta \mid Lm_i), \quad i = 0, \dots, 7. \quad (2.27)$$

Pour chacune de ces questions, le résultats de l'inférence donne (nous prenons ici l'exemple pour $i=0$) :

$$P(\theta | Lm_0) = \frac{\sum_{D, Lm_1, \dots, Lm_7} P(\theta D Lm_0 \dots Lm_7)}{\sum_{D, Lm_0, \dots, Lm_7} P(\theta D Lm_0 \dots Lm_7)} \quad (2.28)$$

$$= \frac{\sum_{D, Lm_1, \dots, Lm_7} P(\theta D) \prod_{i=0}^7 P(Lm_i | \theta D)}{\sum_{D, Lm_0, \dots, Lm_7} P(\theta D) \prod_{i=0}^7 P(Lm_i | \theta D)} \quad (2.29)$$

Puisque nous avons choisi de représenter la distribution $P(\theta D)$ par une loi uniforme, nous pouvons simplifier ce terme. En réorganisant les sommes, nous pouvons écrire :

$$P(\theta | Lm_0) = \frac{\sum_D \left(P(Lm_0 | \theta D) \sum_{Lm_1, \dots, Lm_7} \prod_{i=1}^7 P(Lm_i | \theta D) \right)}{\sum_{D, Lm_0} \left(P(Lm_0 | \theta D) \sum_{Lm_1, \dots, Lm_7} \prod_{i=1}^7 P(Lm_i | \theta D) \right)} \quad (2.30)$$

Et nous voyons apparaître des termes égaux à 1 (sommes dans lesquelles toutes les variables apparaissent à gauche). Le résultat de l'inférence donne donc au final :

$$P(\theta | Lm_0) = \frac{\sum_D P(Lm_0 | \theta D)}{\sum_{D, Lm_0} P(Lm_0 | \theta D)} \quad (2.31)$$

$$= \frac{1}{\alpha_0} \sum_D P(Lm_0 | \theta D), \quad (2.32)$$

où α_0 est une constante de normalisation.

2.2.5.2 Résultats et discussion

La figure 2.3 présente les résultats pour une source lumineuse placée à $\theta = 10^\circ$. Les huit diagrammes périphériques présentent le résultat de l'estimation en utilisant la mesure d'un unique capteur, soit les réponses aux questions (2.27). Le diagramme central présente le résultat de la fusion, soit la distribution $P(\theta | Lm_0 \dots Lm_7)$.

L'information fournie par chacun des huit capteurs pris isolément est très médiocre. Le résultat de la fusion probabiliste de ces huit capteurs est saisissant ! il s'agit d'un Dirac sur la position 10° .

2.2.6 Conclusion sur la programmation bayésienne

La figure 2.4 résume les différentes étapes d'un programme bayésien. Nous utiliserons cette représentation pour résumer chacun des programmes que nous définirons dans le reste du document.

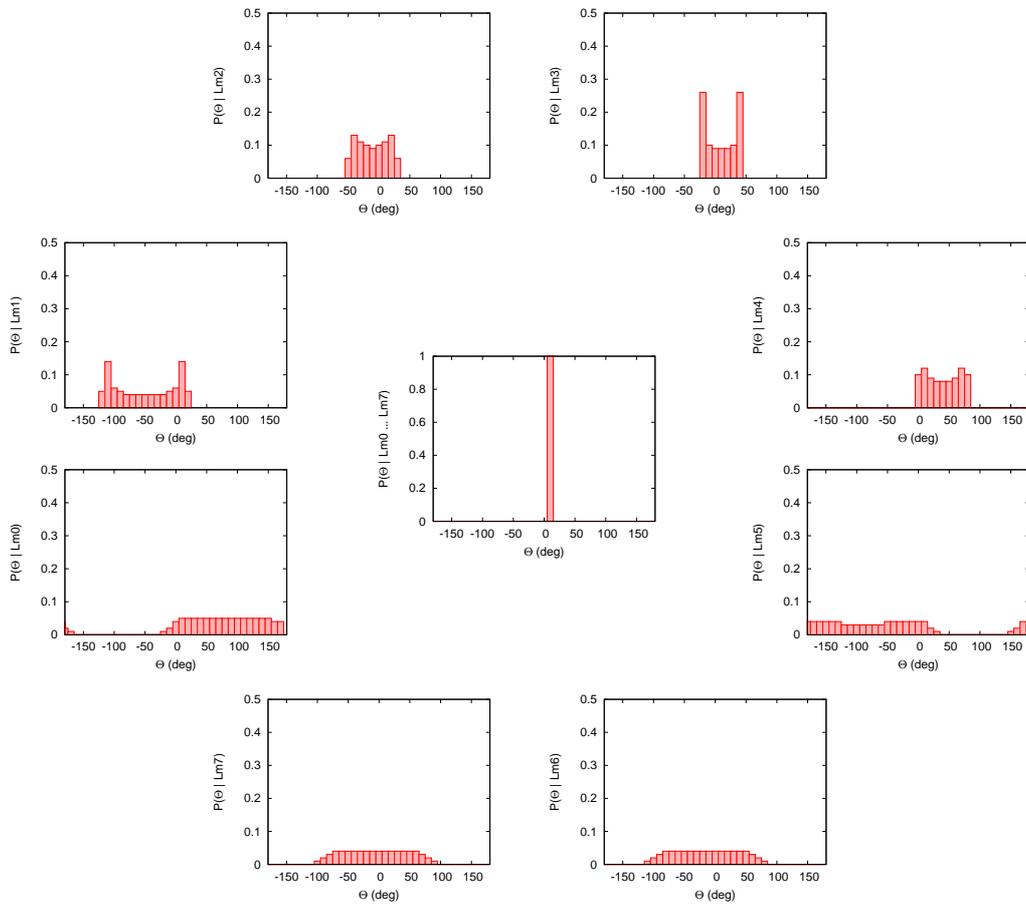


FIG. 2.3 – Résultat de la fusion pour $\theta = 10$ (au centre). Chacun des huit diagrammes périphériques montre le résultat de l'estimation à partir de la mesure d'une unique capteur.

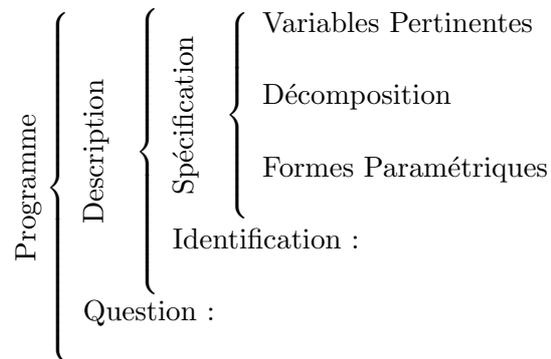


FIG. 2.4: Structure générale d'un programme bayésien.

Cette méthode de programmation a initialement été appliquée à la programmation de tâches « réflexes » simples sur des robots mobiles [Leb99][CB01], et à la CAO-Robotique [Mek99][MMB01]. Depuis, son utilisation s'est étendue à d'autres domaines de la robotique, comme la programmation des bras manipulateurs [GR03], et la localisation [Dia03].

La programmation bayésienne et le raisonnement probabiliste en général prennent une nouvelle dimension avec le projet européen BIBA¹. Un des buts majeurs de ce projet est de montrer la plausibilité biologique de l'inférence probabiliste tant au niveau microscopique [Col02] que macroscopique.

Dans cette thèse, la programmation bayésienne sera utilisée comme un outil. Tous nos raisonnements seront exprimés dans ce formalisme. Nous nous efforcerons également de présenter des méthodes classiques de raisonnement probabiliste (grille d'occupation, filtre bayésien) sous forme de programmes bayésiens.

¹IST-2001-3215 « *Bayesian Inspired Brain and Artefacts : Using probabilistic logic to understand brain function and implement life-like behavioural co-ordination* ». <http://www-biba.imag.fr>

Chapitre 3

Grille d'occupation : concepts de base

Dans ce chapitre, nous présentons l'outil de base de notre modélisation de l'environnement du véhicule : la grille d'occupation.

Nous commençons par une présentation générale de cet outil et des utilisations classiques en robotique.

Nous présentons ensuite de manière plus technique comment une grille d'occupation est estimée à partir d'une observation d'un capteur télémétrique. Cette estimation requiert la définition d'un modèle capteur adéquat.

Enfin, nous présentons comment les observations provenant de plusieurs capteurs peuvent être fusionnées pour estimer une grille d'occupation.

Pour faciliter la lecture et la compréhension des résultats, nous nous limitons pour illustrer le cas statique à des grilles comportant une ou deux dimensions, représentant la position par rapport au véhicule.

3.1 Grille d'occupation

Depuis une quinzaine d'année, la *grille d'occupation*¹ présentée par A. Elfes [Elf89] est un outil très utilisé pour la construction de carte en robotique mobile. Elle est particulièrement adaptée à la représentation d'environnements statiques et structurés, typiquement les couloirs d'un laboratoire ou d'un musée [BBC⁺95]. Elle fournit une carte métrique et exocentrée de l'environnement du robot.

3.1.1 Définition

La grille d'occupation est un découpage de l'espace dans lequel évolue le robot mobile (typiquement 2-D ou 3-D) en un ensemble de *cellules*. On cherche à estimer la probabilité qu'un *objet* quelconque (mur, humain ...) occupe chacune de ces cellules, à partir de mesures fournies par un capteur. Cette estimation se fait de manière incrémentale, en considérant une-à-une les différentes observations.

¹ *Occupancy Grid* en anglais

3.1.2 Hypothèse d'indépendance des cellules

La principale hypothèse des grilles d'occupation est que l'état de chaque cellule est supposé indépendant de l'état des autres cellules. Ainsi la complexité de l'estimation de la grille complète est « cassée ». La probabilité d'occupation de chaque cellule est estimée indépendamment de celle des cellules voisines. Estimer la grille complète revient à appliquer N fois l'estimation d'une seule cellule, N étant le nombre de cellules composant l'environnement complet du robot.

Si cette hypothèse est nécessaire pour assurer une estimation de la grille d'occupation en un temps raisonnable, elle peut entraîner un manque de précision des cartes obtenues, particulièrement lorsque celles-ci sont construites à partir des mesures d'un sonar [Thr01].

L'hypothèse d'indépendance des cellules est tout de même largement utilisée, et les grilles obtenues se révèlent satisfaisantes. Nous ferons donc cette hypothèse dans toute la suite de ce document.

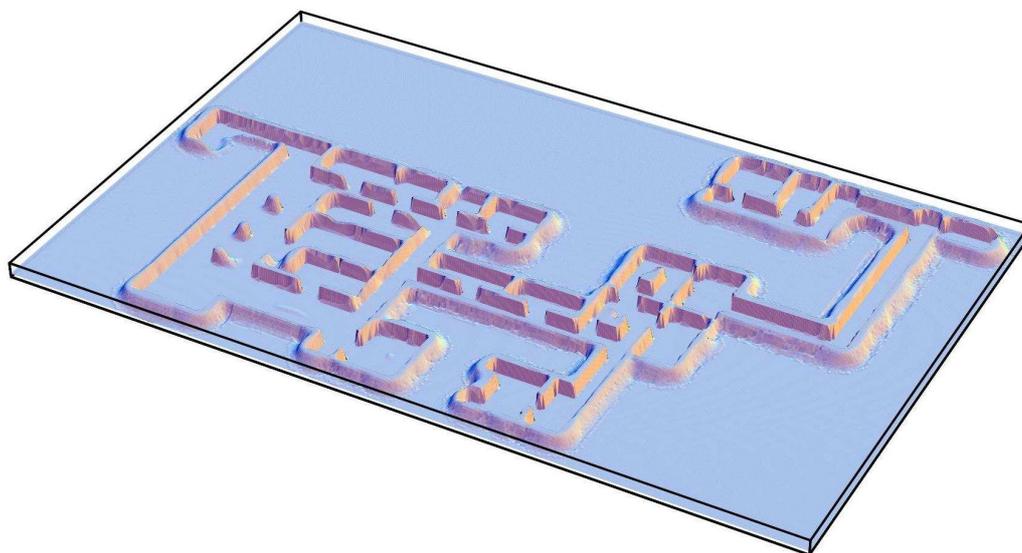


FIG. 3.1: Exemple de grille d'occupation obtenue sur simulateur [Thr98]. La hauteur d'un point représente la probabilité d'occupation de la cellule. Dans cet exemple, l'environnement du robot mesure 105 mètres par 63.

3.1.3 Utilisation des grilles d'occupation

La figure 3.1 montre un exemple typique d'utilisation des grilles d'occupation : la modélisation en deux dimensions d'environnements statiques et structurés [Thr98]. Cette modélisation est exo-centrée, puisqu'elle se fait dans un repère extérieur au robot. Sur cette figure, obtenue sur simulateur, la hauteur d'un point correspond à la probabilité d'occupation de la cellule correspondante. Pour construire l'ensemble de la carte, le robot

a parcouru son environnement pendant 45 minutes, la grille étant mise à jour à chaque observation des capteurs. Pour obtenir une telle précision, le simulateur considérait une odométrie, et donc une localisation du robot, parfaite. Les imprécisions d'un capteur de type sonar étaient quant à elles prises en compte. Les résultats obtenus avec un vrai robot sont bien entendu bien moins précis, compte tenu des erreurs de localisation.

À partir de cette représentation, la localisation du robot dans son environnement, la planification et l'exécution de trajectoire peuvent être réalisées. Les grilles d'occupation ont ainsi été à la base de visites guidées de musée, un robot jouant le rôle du guide [BBC⁺95][TBB⁺00]

Une des grandes qualités des grilles d'occupation réside dans son estimation incrémentale. Il est donc facile de tenir compte d'une succession d'observations capteur pour construire la grille. Cette qualité peut être utilisée pour filtrer les objets dynamiques d'une série d'observations capteur [YL97]. Dans cet exemple, le but de l'application était la localisation dans un environnement comportant des objets mobiles. La carte de l'environnement statique avait été préalablement construite.

Une extension des grilles d'occupation pour des environnements contenant des objets mobiles a été proposée par Arbuckle [AHM02]. En plus de contenir des informations sur la probabilité d'occupation, chaque cellule contient des informations sur le temps d'occupation de ces cellules. Ainsi on peut classer les cellules suivant leurs temps d'occupation. Par exemple celles qui sont occupées en permanence correspondent à des objets immobiles (murs, mobilier). On peut également en déduire les cellules correspondant à des lieux de passages fréquents.

Enfin les grilles d'occupation ont également été utilisées pour de la poursuite multipiste. Prassler reconstruit les caractéristiques dynamiques des objets entourant un fauteuil roulant à partir de différences de grilles d'occupation correspondant à des instants successifs. Sa méthode a la particularité d'utiliser des grilles d'occupation centrées sur le fauteuil roulant, mais qui restent en deux dimensions. Les cellules de la grille ne comportent donc pas d'information de vitesse. Mittal [MD03] construit une grille d'occupation en deux dimensions d'une pièce à partir d'informations fournies par plusieurs systèmes stéréoscopiques. Les différents objets sont extraits de cette grille par seuillage. Ces observations sont ensuite utilisées dans un algorithme classique de poursuite multipiste.

3.2 Estimation d'une grille d'occupation à partir d'une observation capteur

Nous présentons tout d'abord dans cette section le mécanisme qui permet d'estimer une grille d'occupation à partir d'une unique observation capteur. Cette présentation est faite sous la forme d'un programme bayésien.

L'utilisation de l'observation du capteur pour l'estimation de la grille requiert la définition d'un modèle capteur. Nous présentons donc la modélisation que nous avons adoptée.

Enfin nous illustrons ce mécanisme d'estimation de la grille d'occupation sur des données

simulées.

3.2.1 Programme bayésien

L'hypothèse d'indépendance des cellules permet de traiter de manière indépendante chaque cellule de la grille. Notre programme doit donc permettre d'évaluer la probabilité d'occupation d'une cellule, sachant l'observation du capteur.

3.2.1.1 Variables

- Nous cherchons à représenter par des variables probabilistes les différentes informations :
- la région de l'espace à laquelle correspond notre cellule. Nous avons donc besoin d'une variable que nous notons X . La notation X est ici une notation générale. Suivant l'espace représenté par la grille, cette variable peut comporter une ou plusieurs dimensions ;
 - l'état de cette cellule : vide ou occupée? Nous avons donc besoin d'une variable booléenne, que nous notons E_X . $[E_X = 1]$ signifie « la cellule X est occupée par un objet », $[E_X = 0]$ signifie bien sûr « la cellule X est inoccupée ». Du fait de l'hypothèse d'indépendance des cellules, nous n'avons pas à considérer l'état d'occupation des cellules voisines. E_X est donc la seule variable dont nous avons besoin ;
 - la mesure capteur. Nous avons donc besoin d'une variable que nous noterons Z . Comme pour la variable X , cette variable Z est une notation générale, elle peut comporter plusieurs dimensions suivant l'espace observé par le capteur.

Ces variables définissent la distribution de probabilité conjointe :

$$P(X \ E_X \ Z). \quad (3.1)$$

3.2.1.2 Décomposition

Le but de la décomposition dans ce cas est de déterminer comment est utilisée la mesure capteur dans l'estimation de l'état de notre cellule. En appliquant deux fois la règle du produit, on choisit la décomposition :

$$P(X \ E_X \ Z) = P(X)P(E_X \mid X)P(Z \mid E_X \ X). \quad (3.2)$$

Nous devons maintenant choisir une forme paramétrique pour chacun des termes apparaissant dans cette décomposition.

3.2.1.3 Formes paramétriques

$P(X)$ représente l'information dont nous disposons sur l'environnement de travail du robot. Elle est donc choisie uniforme.

$P(E_X \mid X)$ représente l'information *a priori* sur l'occupation de notre cellule X . Elle peut par exemple provenir d'une précédente estimation. Si rien n'est connu sur l'environnement, la distribution uniforme est choisie.

$P(Z | E_X X)$ est couramment nommée le « modèle capteur », et représente la réponse attendue du capteur sachant si la cellule X est occupée ou non. Comme ce modèle est primordial, nous consacrerons la sous-section suivante à l'étudier en détail sur notre problème. Nous illustrerons également son influence sur le résultat de l'estimation.

Notre description est maintenant complète, nous pouvons maintenant l'utiliser, en spécifiant des questions.

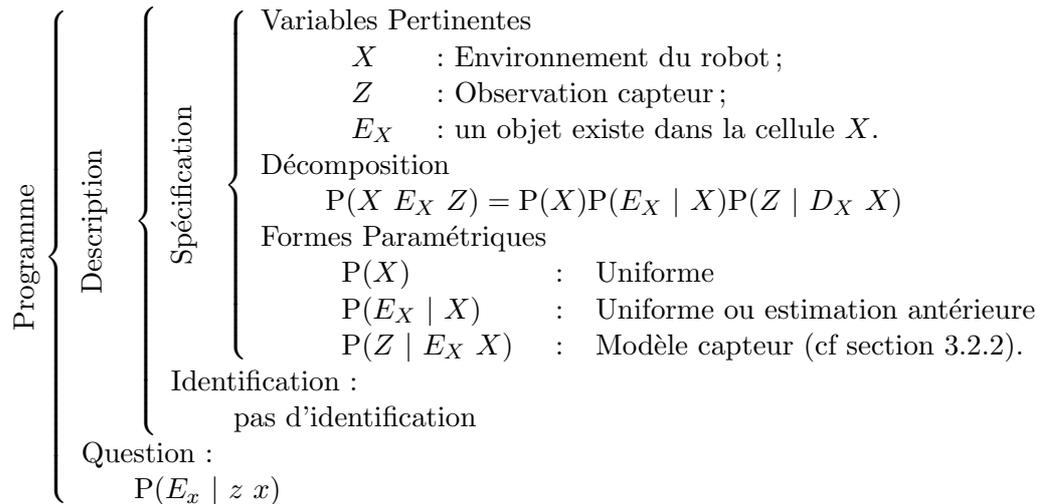


FIG. 3.2: Estimation de l'état d'une cellule : programme bayésien.

3.2.1.4 Question

Le but de la grille d'occupation est d'estimer la probabilité d'occupation d'une cellule x connaissant la mesure capteur z . Nous posons donc à notre description la question suivante :

$$P(E_x | z x). \quad (3.3)$$

Le résultat de l'inférence pour cette question donne :

$$P(E_x | z x) = \frac{P(x E_x z)}{P(z x)}; \quad (3.4)$$

$$= \frac{P(x)P(E_x | x)P(z | E_x x)}{P(x)P(z | x)}; \quad (3.5)$$

$$= \frac{P(E_x | x)P(z | E_x x)}{\sum_{E_x} P(E_x | x)P(z | E_x x)}. \quad (3.6)$$

On remarque donc que le choix de la forme paramétrique pour la distribution $P(X)$ n'a pas d'importance pour cette question, la connaissance *a priori* sur X disparaissant au cours de l'inférence.

En posant cette question pour chacune des cellules x de notre environnement, on obtient la grille d'occupation complète de l'environnement du robot.

Le programme que nous venons de détailler est résumé par la figure 3.2.

3.2.2 Modélisation d'un capteur télémétrique

Pour percevoir son environnement, dans notre cas la distance et la vitesse relative des véhicules automobiles, piétons et autres objets pouvant se trouver dans un environnement automobile, un système robotique utilise des capteurs.

L'information récupérée des capteurs est limitée en précision. En effet, de nombreux facteurs peuvent jouer sur la précision de la mesure : température, éclairage, couleur de l'objet... Pour exploiter les mesures d'un capteur, il est donc important de tenir compte de ce manque de précision.

Nous nous attachons ici à la description d'un modèle probabiliste de la réponse d'un capteur de type télémétrique, basée sur une primitive qui est le point. Ce modèle est similaire à celui décrit par Elfes dans sa présentation des grilles d'occupation [Elf92].

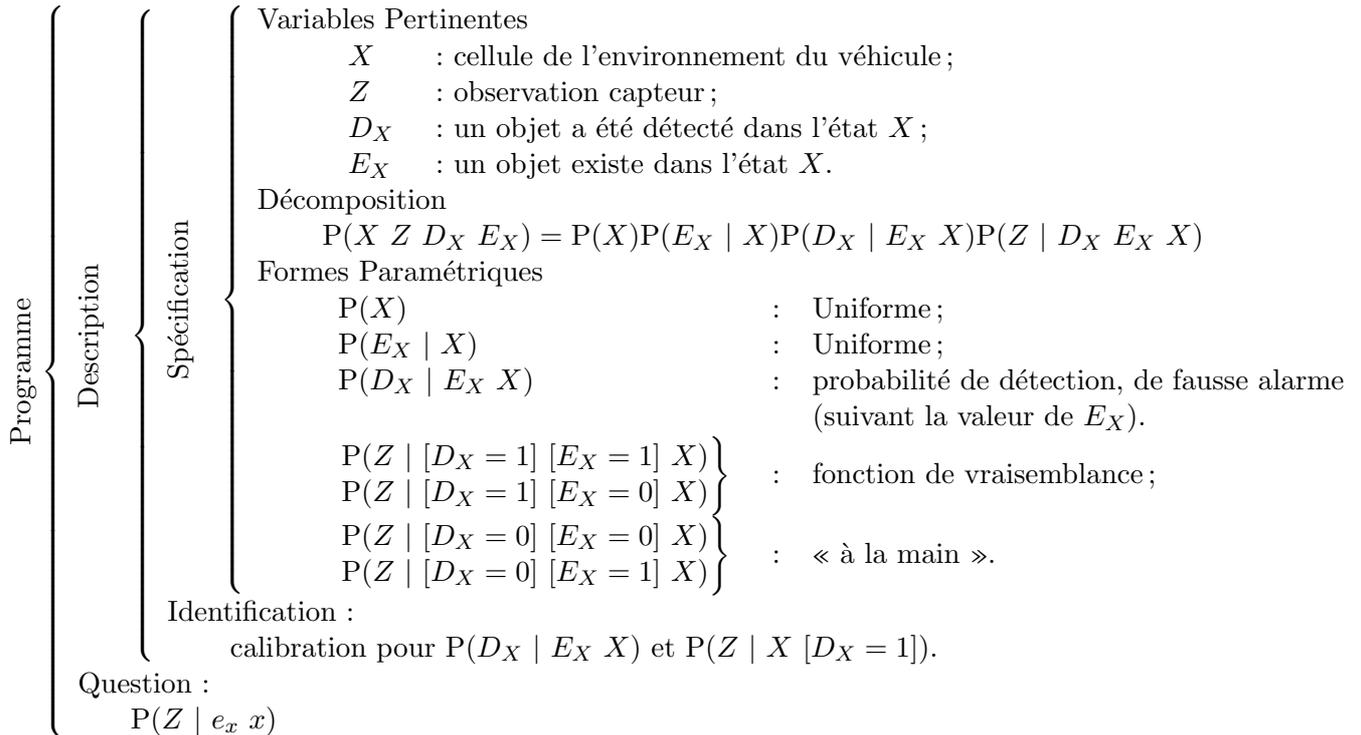


FIG. 3.3: Modèle capteur : programme bayésien.

Le but de ce modèle capteur est de fournir une forme paramétrique à la distribution de probabilité $P(Z | e_x x)$ apparaissant dans la décomposition (3.2). Le formalisme de programmation bayésienne permettant l'utilisation de sous-programme pour la spécification de

formes paramétriques, le modèle capteur est lui-même spécifié sous forme d'un programme bayésien. La figure 3.3 résume le programme que nous allons maintenant détailler.

3.2.2.1 Variables

Les variables qui nous intéressent ici sont :

- la mesure capteur Z . Il faut noter ici que la variable Z inclut une valeur spéciale qui signifie qu'aucune mesure n'a été renvoyée par le capteur. Cette information est en effet importante, comme le montrera l'illustration de la section suivante ;
- la variable X qui définit la cellule de l'environnement à laquelle nous nous intéressons. Il faut noter ici que Z et X ne sont pas nécessairement exprimées dans le même système de coordonnées. Par exemple, si une cellule de la grille représente une position, on peut avoir choisi une découpe de l'environnement dans le repère cartésien. La variable X est donc définie dans ce repère. Le capteur peut quant à lui fournir une observation dans le repère polaire, c'est à dire une distance et un gisement ;
- la variable E_X , variable booléenne signifiant « un objet existe dans la cellule X » ;
- et enfin une variable D_X , booléenne elle aussi, qui signifie « un objet a été détecté dans la cellule X ».

La conjonction des variables E_X et D_X permet d'explicitier les quatre situations possibles pour la réponse d'un capteur. Deux correspondent à un fonctionnement correct du capteur :

- $[E_X = 1] \wedge [D_X = 1]$: un objet occupe la cellule X , il est détecté par le capteur ;
- $[E_X = 0] \wedge [D_X = 0]$: la cellule est vide, le capteur n'y détecte rien.

Les deux autres possibilités correspondent à une défaillance du capteur :

- $[E_X = 1] \wedge [D_X = 0]$: un objet occupe la cellule X , il n'est pas détecté par le capteur. Ceci peut être dû à plusieurs raisons : la cellule X est en dehors du champ de vue du capteur, elle est masquée par un autre objet, ou le capteur est défaillant ;
- $[E_X = 0] \wedge [D_X = 1]$: la cellule est vide et pourtant le capteur y détecte un objet. On parle alors communément de « fausse alarme ».

Ces variables définissent la distribution de probabilité conjointe :

$$P(Z \ X \ E_X \ D_X). \quad (3.7)$$

3.2.2.2 Décomposition

Nous choisissons la décomposition :

$$P(Z \ X \ E_X \ D_X) = P(X)P(E_X \mid X)P(D_X \mid E_X \ X)P(Z \mid D_X \ E_X \ X). \quad (3.8)$$

3.2.2.3 Formes paramétriques - identification

Pour que la description soit complète, il reste à associer à chacune des distributions apparaissant dans (3.8) une forme paramétrique.

$P(X)$ et $P(E_X | X)$ représentent des *a priori* sur l'environnement du véhicule. Afin de ne privilégier aucune situation sensorielle, ces distributions sont choisies uniformes.

Les distributions de probabilité sur la variable D_X représentent suivant la valeur de la variable E_X la capacité du capteur à détecter ou non une cible, ou bien à générer des fausses alarmes :

- $P([D_X = 1] | [E_X = 1] X)$ représente la probabilité de détection du capteur. Elle est souvent notée $P_D(X)$. $P([D_X = 0] | [E_X = 1] X) = 1 - P_D(X)$ représente bien sûr la probabilité que le capteur manque une cible existante. Nous appelons cette distribution probabilité de non-détection, et nous la notons $P_{ND}(X)$.
- Au contraire, $P([D_X = 1] | [E_X = 0] X)$ représente la probabilité de fausse alarme du capteur, notée $P_{FA}(X)$. $P([D_X = 0] | [E_X = 0] X)$ est alors égale à $1 - P_{FA}(X)$.

Ces distributions de probabilités dépendent bien entendu des capteurs et des conditions dans lesquelles on les utilise. $P_D(X)$ et $P_{FA}(X)$ peuvent être estimées à partir d'une modélisation des caractéristiques physiques des capteurs et des objets détectés [Swe54][Mar47]. Elles peuvent également être apprises expérimentalement dans les conditions particulières d'utilisation du capteur.

Il nous reste à définir les différentes formes paramétriques associées à la famille de distribution $P(Z | D_X E_X X)$, suivant les valeurs de D_X et E_X . Ces distributions de probabilité nous permettent d'exprimer des connaissances supplémentaires sur le fonctionnement de notre capteur.

- $P(Z | [D_X = 1] [E_X = 0] X)$ et $P(Z | [D_X = 1] [E_X = 1] X)$: dans les deux cas, une détection a eu lieu pour la cellule X . Ces distributions représentent donc la probabilité sur la réponse du capteur, sachant qu'une détection a eu lieu en X . Lors de l'inférence, *i.e.* lors de l'estimation de l'état du système à partir de l'observation du capteur, cette fonction est utilisée avec Z instanciée pour les différentes valeurs de la variable X . C'est pourquoi elle est souvent vue comme une fonction $L_z(X) = P([Z=z] | [D_X=1] X)$, et non comme un ensemble de distributions de probabilité sur Z . Elle est alors connue sous le nom de *fonction de vraisemblance*² [SBC99]. Le plus souvent une famille de distributions normales est utilisée pour représenter $P(Z | [D_X = 1] [E_X = 0] X)$ et $P(Z | [D_X = 1] [E_X = 1] X)$:

$$P(Z | [D_X = 1] E_X X) = \mathcal{N}(Z, \mu(X), \Sigma(X)). \quad (3.9)$$

Pour une valeur de X donnée, la moyenne $\mu(X)$ de la distribution normale représente la réponse attendue du capteur. Si la variable Z n'est pas définie dans le même repère que la variable X , cette fonction $\mu(X)$ peut intégrer un changement de repère. La matrice de covariance $\Sigma(X)$ de cette distribution permet de représenter les variations possibles de la réponse du capteur autour de la réponse attendue. En ce sens, elle représente la *précision* du capteur. Plus les valeurs propres de cette matrice sont grandes, plus les variations possibles sont importantes, donc moins le capteur est précis. Cette matrice de covariance est également une fonction de X , la précision d'un capteur pouvant varier à l'intérieur de son champ de vue.

²“Likelihood function” en anglais

Les paramètres du modèle, comme la moyenne et la matrice de covariance dans le cas Gaussien, peuvent être acquis par la calibration du capteur [Cro89][MR87].

- $P(Z | [D_X = 0] [E_X = 0] X)$: nous savons que la cellule X est vide, et qu'aucune détection n'a eu lieu dans cette cellule. En considérant uniquement ces informations, il est difficile de dire quelle peut être la valeur de l'observation Z . En effet, n'importe quelle valeur de Z est possible, cette valeur étant due à une détection d'objet dans une autre cellule. On choisit donc une distribution uniforme.

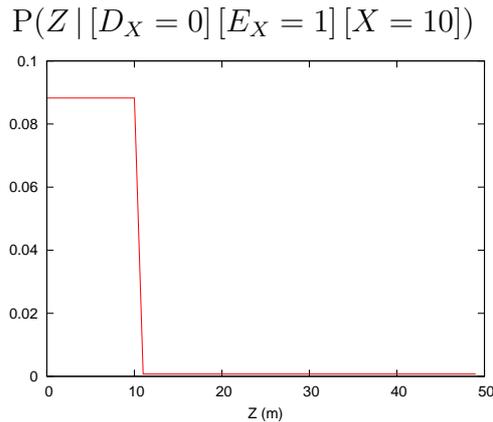


FIG. 3.4: Exemple sur un cas à une dimension (capteur de distance uniquement) de forme paramétrique pour $P(Z | [D_X = 0] [E_X = 1] [X = 10])$.

- $P(Z | [D_X = 0] [E_X = 1] X)$: pour spécifier cette famille de formes paramétriques, nous pouvons exprimer des connaissances relatives aux occultations d'objet. Nous savons que la cellule X est occupée par un objet, et qu'aucune détection n'a eu lieu dans cette cellule. Deux explications sont possibles : soit un objet masque la cellule X et a été détecté par le capteur, soit l'objet présent dans la cellule X n'est pas masqué, et n'a pas été détecté. La forme paramétrique choisie pour la distribution $P(Z | [D_X = 0] [E_X = 1] X)$ traduit ces explications : la probabilité pour la mesure d'être dans une zone masquée par la cellule X est différente de celle des zones non masquées.

La figure 3.4 illustre une telle distribution, dans un cas à une seule dimension. Z et X traduisent la distance en mètres d'un objet au capteur. Ces deux variables peuvent prendre des valeurs comprises entre 0 et 50 mètres, le pas de discrétisation étant d'un mètre. La valeur en chaque point de la distribution uniforme sur Z est donc environ égale à 0,02. Dans le cas illustré sur la figure 3.4, la cellule $[X = 10]$ est occupée mais aucune détection n'y a eu lieu. Pour traduire le fait que l'observation a plus de chance de provenir d'une zone non masquée par la cellule $[X = 10]$, et donc d'être inférieure à 10, nous attribuons une probabilité supérieure à l'uniforme pour les valeurs de Z inférieures à 10. Cette valeur dépend de $P_{FA}(X)$ et de $P_D(x)$. Les valeurs de la distribution pour $Z > 10$ sont fixées de manière à assurer la sommation à 1. Dans ce cas elles sont très proches de 0.

Une fois les paramètres du modèle donnés, la description est complète. On peut donc l'utiliser en posant une question.

3.2.2.4 Question

Le but du modèle capteur, et donc de ce programme bayésien, est de fournir une forme paramétrique à la distribution de probabilité $P(Z | E_X X)$, apparaissant dans la décomposition (3.2). C'est donc cette question que nous posons à notre description :

$$P(Z | e_x x). \quad (3.10)$$

Pour cette question, le résultat de l'inférence donne :

$$P(Z | e_x x) = \frac{\sum_{D_X} P(Z | x | e_x | D_X)}{P(e_x | x)} \quad (3.11)$$

$$= \frac{P(x)P(e_x | x) \sum_{D_X} P(D_X | e_x | x)P(Z | D_X | e_x | x)}{P(x)P(e_x | x) \underbrace{\sum_{Z, D_X} P(D_X | e_x | x)P(Z | D_X | e_x | x)}_{=1}} \quad (3.12)$$

$$= \sum_{D_X} P(D_X | e_x | x)P(Z | D_X | e_x | x). \quad (3.13)$$

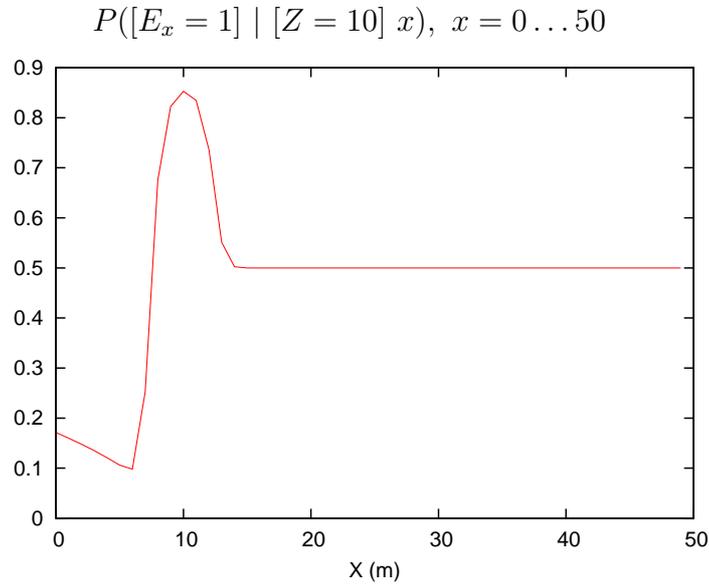


FIG. 3.5: Exemple d'estimation de grille d'occupation à une dimension à partir d'une mesure capteur.

3.2.3 Illustration

Le but de cette sous-section est d'illustrer l'estimation d'une grille d'occupation à partir d'une unique observation capteur. Nous commençons par une grille comportant une seule dimension avant d'étendre à une grille à deux dimensions.

3.2.3.1 Grille à une dimension

La figure 3.5 montre le résultat de l'estimation d'une grille d'occupation dans un cas très simple : la grille comporte une seule dimension, qui correspond à la distance d'un objet au capteur. La résolution de la grille est de un mètre, pour des valeurs de X comprises entre 0 et 50.

Le modèle capteur utilisé pour cette estimation est bien entendu celui décrit dans la sous-section précédente. En particulier, nous utilisons la distribution de probabilité décrite par la figure 3.4. La probabilité de détection du capteur est supposée constante selon les valeurs de X , elle est fixée à 0,8. La probabilité de fausse alarme est également supposée constante, sa valeur est fixée à 0,1. Enfin, les distributions $P(Z \mid [D = 1] E_X X)$ sont représentées par des lois normales. Puisque les variables Z et X sont exprimées dans le même repère, la fonction $\mu(X)$ représentant la moyenne des lois normales est la fonction identité. Enfin la variance des lois normales est fixée à 1 mètre quelle que soit la valeur de X .

Dans cet exemple, le capteur détecte un objet dans la cellule 10 de l'environnement. La figure décrit donc les distributions de probabilité $P(E_x \mid [Z = 10] x)$, pour les différentes cellules de l'environnement.

Comme on pouvait s'y attendre, on remarque une très forte probabilité d'occupation autour de l'observation du capteur. Ce qui est plus intéressant se situe dans les cellules situées « derrière » l'observation capteur. La probabilité d'occupation y est en effet égale à 0,5, ce qui signifie que l'on ne peut prendre de décision sur l'occupation de ces cellules. Intuitivement, cela correspond à dire que le capteur ayant détecté un objet à 10 mètres devant le véhicule n'a logiquement pu détecter un éventuel deuxième objet situé derrière le premier. On ne peut donc avoir en considérant uniquement ce capteur si un objet est présent ou non à une distance supérieure à 10 mètres devant le véhicule.

3.2.3.2 Grille à deux dimensions

Pour cette illustration, nous voulons estimer la grille d'occupation de l'environnement situé devant le véhicule, sur une distance de 15 mètres, et sur une largeur de 5 mètres de chaque côté du capteur. Cet espace est découpé en cellules carrées de 25 cm de côté.

Le capteur simulé, situé à l'origine du repère polaire attaché au véhicule, est capable de détecter un objet situé à une distance inférieure à 10 mètres, dans un cône de 50 degrés de demi-angle d'ouverture. La figure 3.6 montre le champ de vue de ce capteur, et la zone pour laquelle nous souhaitons estimer la probabilité d'occupation. On remarque que le champ de vue de notre capteur ne couvre pas entièrement cette zone.

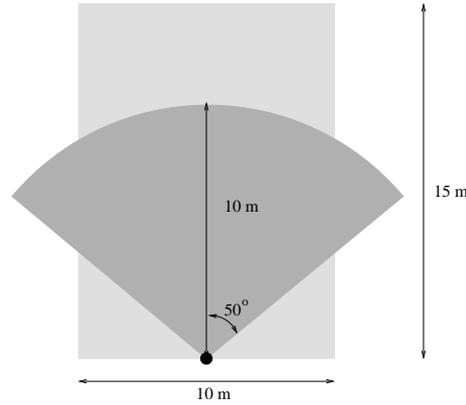


FIG. 3.6: Champ de vue du capteur (gris foncé), et zone couverte par la grille d'occupation (gris clair). Ces deux zones ne se superposent pas. Le capteur est matérialisé par le disque noir.

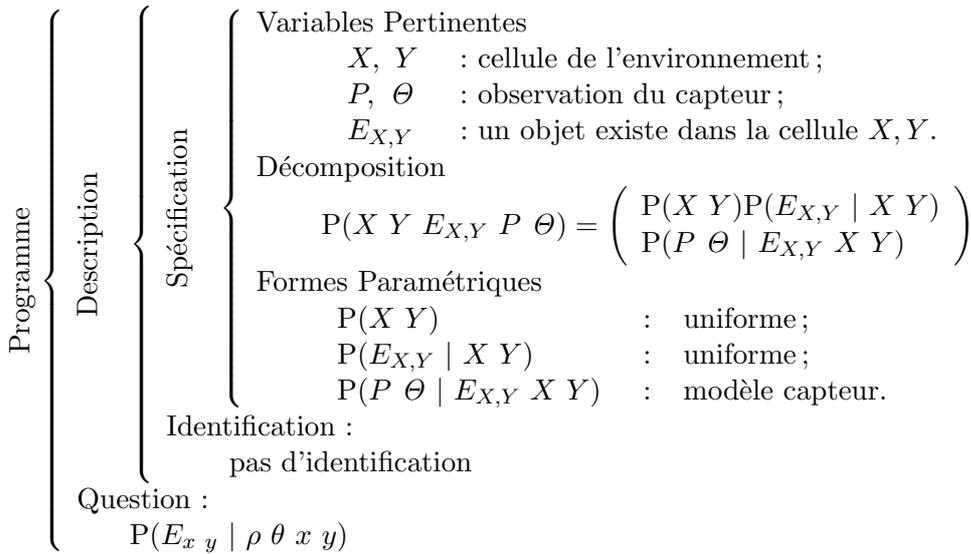


FIG. 3.7: Estimation statique d'une grille d'occupation 2d : programme bayésien.

Programme bayésien Le programme bayésien pour cette estimation est directement adapté du programme donné dans la section 3.2.1. Dans le cas présent, une cellule est représentée par la conjonction de deux variables : X et Y (distances longitudinale et latérale). Une instance de cette conjonction est notée x, y . L'observation du capteur est elle exprimée dans le repère polaire. Elle est constituée de la conjonction de deux variables, que nous notons³ P et θ . Une instance de cette conjonction est notée ρ, θ . L'absence de mesure est notée Φ . Enfin la variable booléenne E_X est notée $E_{X,Y}$.

³en grec, le ρ majuscule se note P

La figure 3.7 résume l'adaptation à notre cas de ce programme bayésien.

Modèle capteur Le modèle capteur utilisé comme forme paramétrique pour la distribution de probabilité $P(P \Theta | E_{X,Y} X Y)$ est bien entendu constitué par un sous-programme bayésien, correspondant au programme décrit dans la section précédente, et adapté à la situation présente.

Pour représenter la fonction de vraisemblance de ce modèle capteur, *i.e.* les formes paramétriques associées aux familles de distributions $P(P \Theta | [E_{X,Y} = 1] [D_{X,Y} = 1] X Y)$ et $P(P \Theta | [E_{X,Y} = 0] [D_{X,Y} = 1] X Y)$, nous choisissons une famille de lois normales à deux dimensions. Puisque l'observation et la position de la cellule ne sont pas exprimées dans le même repère, la fonction donnant la moyenne $(\bar{\rho}, \bar{\theta})$ de ces lois intègre un changement de repère :

$$\begin{aligned}\bar{\rho}(x, y) &= \sqrt{x^2 + y^2} \\ \bar{\theta}(x, y) &= \arctan \frac{x}{y}.\end{aligned}$$

La précision du capteur en distance est supposée être indépendante de la précision en gisement. La matrice de covariance des lois normales est donc de la forme :

$$\Sigma(x, y) = \begin{pmatrix} \sigma_{\rho}^2(x, y) & 0 \\ 0 & \sigma_{\theta}^2(x, y) \end{pmatrix}.$$

Dans nos simulations, nous avons considéré les précisions constantes sur le champ de vue du capteur. Nous ferons varier les valeurs de ces constantes, et donc la précision du capteur, pour illustrer les conséquences de cette précision sur l'estimation des grilles.

La probabilité de détection $P_D(X, Y)$ est choisie constante sur le champ de vue du capteur. Nous ferons également varier la valeur de cette constante dans nos différentes illustrations. En dehors du champ de vue du capteur, cette probabilité est nulle puisque le capteur ne peut rien observer en dehors de son champ de vue.

Enfin la probabilité de fausse alarme $P_{FA}(X, Y)$ est également supposée constante sur le champ de vue du capteur. Nous ferons varier la valeur de cette constante dans nos différentes illustrations. En dehors du champ de vue du capteur, cette probabilité est nulle, puisque le capteur ne peut générer de détection en dehors de son champ de vue. Il ne peut donc pas générer de fausses alarmes.

3.2.3.3 Résultats

La figure 3.8 montre un premier exemple de grille d'occupation quand aucun objet n'a été détecté. La couleur indique la valeur de la probabilité que la cellule correspondante soit occupée par un objet, c'est à dire la valeur de $P([E_{x,y} = 1] | \Phi x y)$.

Cette exemple nous permet d'illustrer deux propriétés importantes des grilles d'occupation. Premièrement, en dehors du champ de vue du capteur, on ne peut rien dire. La probabilité d'occupation des cellules correspondantes est donc de 0,5. Deuxièmement, le

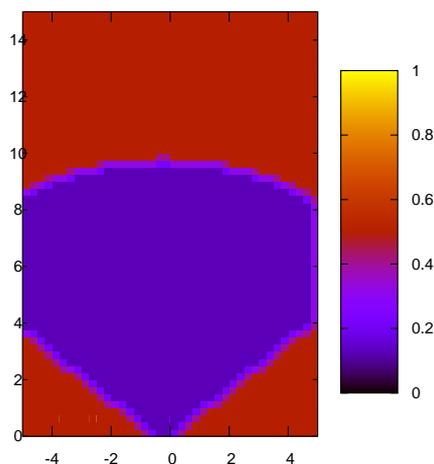


FIG. 3.8: Grille d'occupation quand aucun objet n'est détecté. La couleur indique la valeur de la probabilité $P([E_{x,y}=1] | \Phi x y)$.

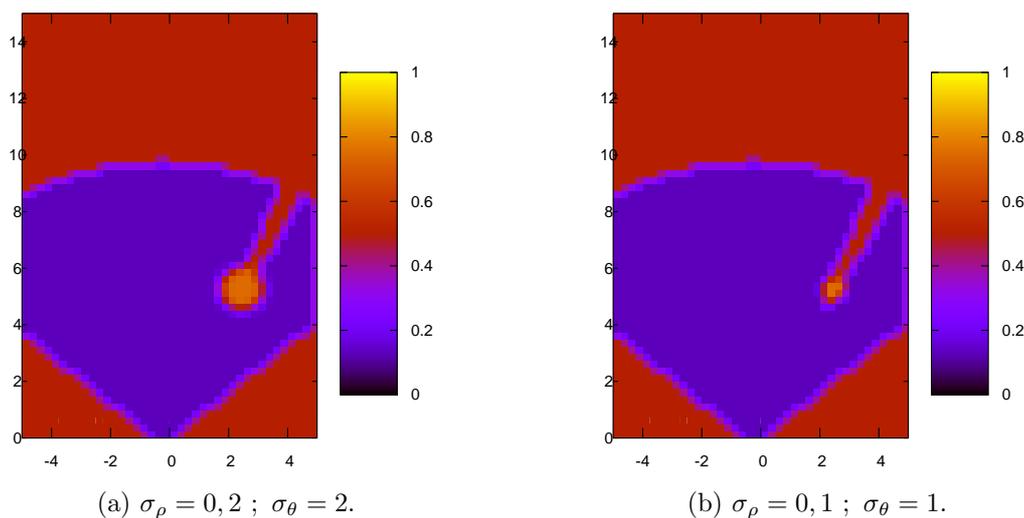


FIG. 3.9: Grilles d'occupations, un objet ayant été détecté, pour différentes précisions du capteur. La couleur indique la valeur de la probabilité $P([E_{x,y}=1] | \rho \theta x y)$.

fait que le capteur n'ait détecté aucune cible doit être pris en compte. Ceci se traduit par le fait que la probabilité d'occupation des cellules situées dans le champ de vue du capteur est inférieure à 0,5. Cette valeur dépend des probabilités de détection et de fausse alarme dans ces mêmes cellules.

La figure 3.9(a) montre la même grille d'occupation alors qu'un objet a été détecté par le capteur. En dehors du champ de vue du capteur, la probabilité d'occupation reste égale à 0,5, on ne peut toujours rien dire sur cette partie de l'espace. Dans le champ de vue du capteur, on voit apparaître une zone de forte probabilité d'occupation, qui correspond à l'objet détecté par le capteur. La taille de cette tache dépend de la précision du capteur,

et donc ici de la valeur de l'écart-type donnée pour la fonction de vraisemblance lors de la modélisation. La figure 3.9(b) montre la même grille d'occupation obtenue avec la même observation capteur, le capteur ayant été modélisé comme plus précis. La surface de la zone supposée occupée est moins importante.

Sur ces deux figures, la probabilité d'occupation de la zone située derrière l'observation du capteur est égale à 0,5. Ceci traduit le fait que le capteur ne peut logiquement fournir une observation dans une zone cachée par la première, ceci même si cette observation correspond à une fausse alarme.

3.3 Fusion de capteurs dans les grilles d'occupation

La section précédente présentait la méthode d'estimation de la grille d'occupation pour le cas où un unique capteur fournit une observation de l'environnement. Nous allons maintenant présenter le cas dans lequel plusieurs capteurs fournissent des observations de cet environnement.

Nous commençons par une discussion sur l'intérêt d'utiliser plusieurs capteurs et les difficultés pratiques de la fusion.

Nous présentons ensuite la méthode d'estimation de la grille d'occupation basée sur les observations de plusieurs capteurs, sous forme d'un programme bayésien. Des résultats issus de simulations viennent ensuite illustrer cette méthode.

3.3.1 Utilisation de plusieurs capteurs : intérêts et difficultés

En principe, l'utilisation de plusieurs capteurs permet d'augmenter la robustesse de l'observation et de la caractérisation d'un phénomène physique.

Ainsi les humains et les animaux utilisent naturellement plusieurs systèmes de perception afin d'augmenter leur chance de survie. Par exemple, un prédateur caché derrière un arbre ne peut être détecté grâce à la vue, mais peut l'être grâce à l'odorat ou à l'ouïe. L'utilisation de plusieurs espaces de perception suivant la nature du capteur est un premier avantage de la fusion multi-capteur.

L'utilisation de différents types de capteurs permet également de s'affranchir des limitations propres à chaque capteur. Par exemple, un laser est sensible aux conditions atmosphériques, particulièrement à la pluie, à la neige et au brouillard. Un radar est insensible à ces conditions.

Ensuite, l'utilisation de plusieurs capteurs permet une meilleure robustesse aux défaillances de fonctionnement d'un capteur, que ce soit pour les défauts d'observations ou au contraire pour les fausses alarmes.

Enfin, l'utilisation d'informations redondantes permet d'augmenter la précision de l'estimation d'un phénomène physique, en utilisant par exemple des techniques statistiques. Donc l'utilisation de plusieurs capteurs du même type peut également être intéressante. Cet aspect de la fusion sera illustré dans la sous-section 3.3.3.

La fusion de données multicateur est donc naturellement très étudiée depuis quelques années, et ceci dans de nombreux cadres d'application, civiles ou militaires [HL97]. Il n'existe pas de formalisme général pour décrire un processus de fusion de données. Le choix de la méthode de fusion dépend en effet fortement du type de données disponibles et de l'application visée [AG92].

Cependant, les différentes approches de la fusion de données, les applications nombreuses et portant sur des domaines très différents les uns des autres amènent au constat qu'il n'existe pas de formalisme général pour décrire un processus de fusion de données.

En contrepartie des différents avantages que nous venons de citer, l'utilisation de plusieurs capteurs pose des problèmes pratiques.

La principale difficulté vient de la désynchronisation des capteurs. En général, les observations de différents capteurs ne correspondent pas au même instant. De plus, l'observation de l'environnement par un capteur à un instant t n'est pas disponible immédiatement, du fait de la latence du capteur. L'estimation de l'environnement, pour nous l'estimation de la grille d'occupation, à un instant t se fait donc avec des observations plus anciennes. Cette différence doit éventuellement être prise en compte. Nous présentons dans la section 3.3.4 des considérations sur la prise en compte de ce décalage temporel directement dans le modèle capteur lorsqu'une datation des données est disponible.

Une seconde difficulté vient du fait que les observations de différents capteurs ne sont pas forcément données dans le même espace de représentation de l'environnement. Afin d'être fusionnées, les observations doivent être exprimées dans le même repère, ce qui n'est pas forcément immédiat. Pour nous, cette difficulté est prise en compte dans le modèle capteur, qui relie le repère dans lequel est donnée l'information au repère dans lequel elle est traitée.

Enfin, il est nécessaire de définir une logique particulière pour prendre en compte simultanément les observations des différents capteurs, une fusion naïve de ces observations pouvant entraîner une perte de précision [BBS96].

3.3.2 Programme bayésien

Le programme bayésien que nous détaillons ici présente le cas général de l'estimation de la probabilité d'occupation d'une cellule de la grille, lorsque S capteurs fournissent des observations de l'environnement.

La figure 3.10 résume le programme bayésien que nous détaillons ici.

3.3.2.1 Variables

Les variables qui nous intéressent ici sont :

- l'observation de chaque capteurs : Z_1, \dots, Z_S ;
- la variable X qui définit la cellule de l'environnement du véhicule à laquelle nous nous intéressons ;
- la variable E_X , qui précise si la cellule X est occupée par un objet ou non.

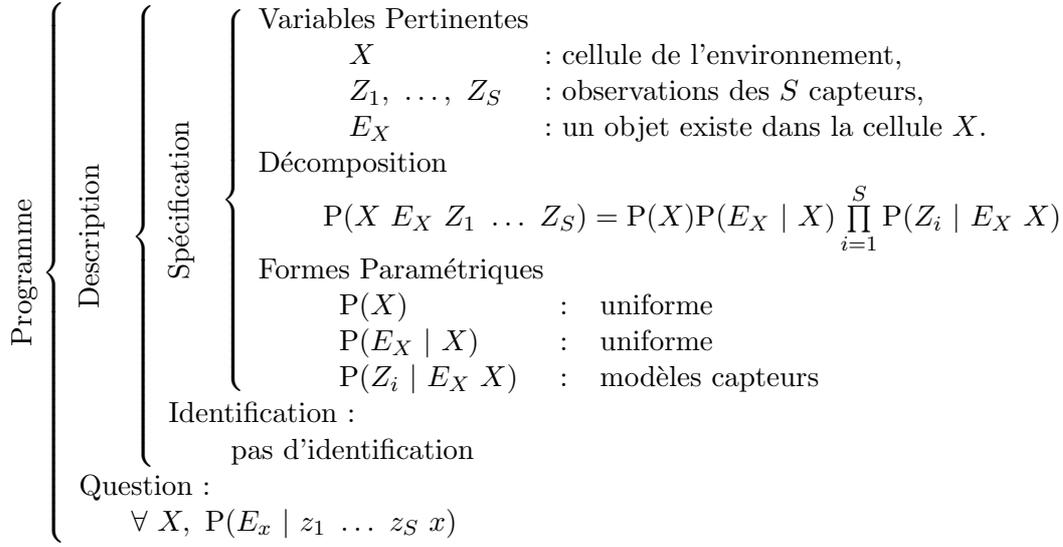


FIG. 3.10: Estimation de l'état d'une cellule à partir des observations de plusieurs capteurs : programme bayésien.

Ces variables définissent la distribution de probabilité conjointe :

$$P(Z_1 \dots Z_S X E_X) \quad (3.14)$$

3.3.2.2 Décomposition

En appliquant successivement la règle du produit, nous pouvons décomposer la distribution de probabilité conjointe (3.14) sous la forme :

$$P(X E_X Z_1 \dots Z_S) = P(X)P(E_X | X) \prod_{s=1}^S P(Z_s | Z_1 \dots Z_{s-1} E_X X). \quad (3.15)$$

Afin de simplifier cette décomposition, nous faisons une importante hypothèse d'indépendance conditionnelle. Nous faisons l'hypothèse que connaissant l'état de la cellule, les observations des différents capteurs sont indépendantes, ce qui se traduit :

$$\forall s \neq j, P(Z_s | Z_j E_X X) = P(Z_s | E_X X). \quad (3.16)$$

Ceci signifie que connaissant la position de l'objet détecté, la connaissance de l'observation du capteur j ne m'apporte aucune connaissance supplémentaire sur la réponse du capteur s .

L'hypothèse d'indépendance conditionnelle ne signifie pas que l'on considère les mesures des différents capteurs indépendantes, ce qui s'écrirait :

$$P(Z_s | Z_j) = P(Z_s). \quad (3.17)$$

L'hypothèse d'indépendance des capteurs est clairement fautive. Les mesures de deux capteurs observant le même phénomène étant forcément corrélées.

L'indépendance conditionnelle entre variable est très fréquemment utilisée par les outils de modélisation et d'inférence probabiliste (par exemple dans le système *MAIS* développé à l'École des Mines [Ste01]). Elle permet en effet de réduire la complexité d'une décomposition.

Grâce à l'hypothèse d'indépendance conditionnelle (3.16), la décomposition de la distribution de probabilité conjointe devient :

$$P(Z_1 \dots Z_S X E_X) = P(X)P(E_X | X) \prod_{s=1}^S P(Z_s | E_X X). \quad (3.18)$$

Une fois la décomposition choisie, il reste à donner une forme paramétrique à chacun des termes y apparaissant pour finir la description.

3.3.2.3 Formes paramétriques - identification

Comme nous ne voulons exprimer aucune connaissance sur l'environnement du véhicule, $P(X)$ est choisie uniforme.

$P(E_X | X)$ représente la connaissance *a priori* dont nous disposons quant à l'existence d'une cible dans la cellule X . Nous verrons dans le chapitre suivant que cette connaissance peut être donnée par une phase de prédiction, qui tient compte de la dernière estimation de notre grille d'existence et d'un modèle dynamique des cibles. Si cette phase de prédiction n'a pas eu lieu, cette distribution est choisie uniforme.

$P(Z_s | E_X X)$ est donnée par le modèle capteur que nous avons décrit dans la section 3.2.2. Bien entendu, un modèle différent est spécifié pour chaque capteur. Nous utilisons donc ici S sous-programmes bayésiens, un par capteur.

Pour ces formes paramétriques, aucune phase d'identification n'est nécessaire, puisqu'il ne reste pas de paramètres libres.

3.3.2.4 Question

Nous cherchons à estimer la probabilité d'occupation de la cellule X , connaissant les observations des différents capteurs. Nous posons donc la question :

$$P(E_x | z_1 \dots z_S x). \quad (3.19)$$

Pour cette question, l'inférence donne :

$$P(E_x | z_1 \dots z_S x) = \frac{P(z_1 \dots z_S x E_x)}{P(z_1 \dots z_S x)} \quad (3.20)$$

$$= \frac{P(x)P(E_x | x) \prod_{s=1}^S P(z_s | E_x x)}{P(x) \sum_{E_x} (P(E_x | x) \prod_{s=1}^S P(z_s | E_x x))} \quad (3.21)$$

$$= \frac{P(E_x | x) \prod_{s=1}^S P(z_s | E_x x)}{\sum_{E_x} (P(E_x | x) \prod_{s=1}^S P(z_s | E_x x))} \quad (3.22)$$

Ici encore, le choix de la forme paramétrique associée à $P(x)$ n'a pas d'importance.

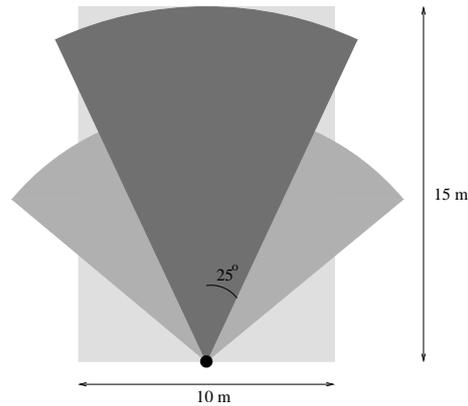


FIG. 3.11: Champ de vue du deuxième capteur (foncé), ajouté à la situation décrite dans la figure 3.6.

3.3.3 Illustration

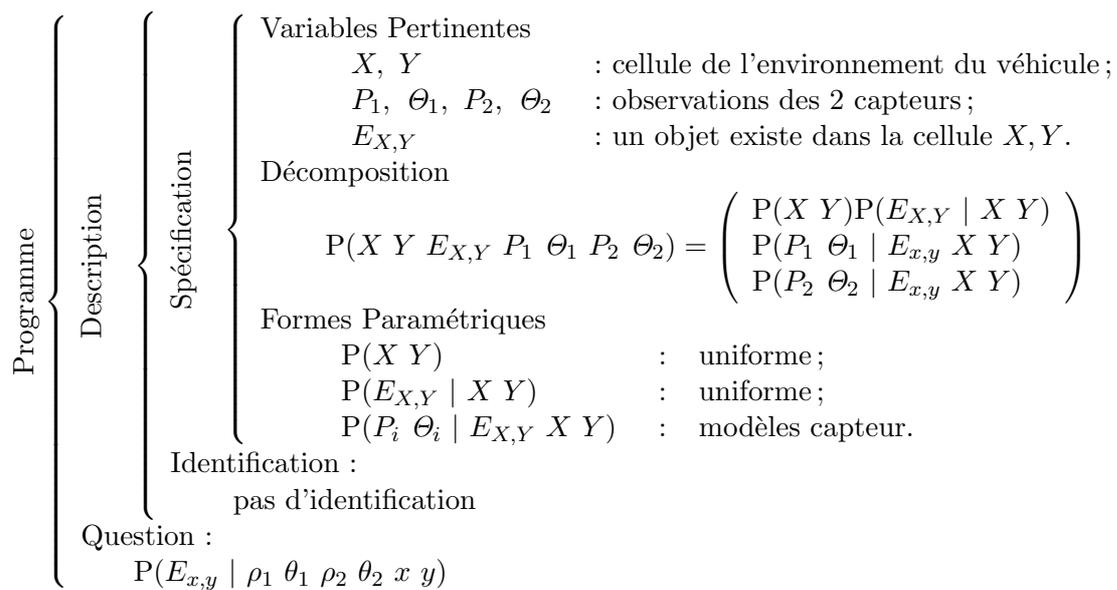


FIG. 3.12: Illustration : programme bayésien.

Pour cette illustration, nous nous plaçons dans la même situation que dans la section 3.2.3.2, décrite par la figure 3.6. Nous ajoutons un second capteur, d'une portée de 15 mètres, et de demi-ouverture 25 degrés.

La figure 3.11 décrit la nouvelle situation. On remarque que les deux champs de vue des deux capteurs ne sont pas identiques, et que la zone pour laquelle nous souhaitons estimer la grille d'occupation est mieux couverte par la réunion des deux champs de vue que par

un seul des deux. Une zone est cependant commune aux champs de vue des deux capteurs.

L'adaptation du programme bayésien pour l'estimation de la grille d'occupation en tenant compte de ces deux capteurs est présentée par la figure 3.12. La cellule d'intérêt est désignée par la conjonction de variables $X Y$, les mesures des deux capteurs par les conjonction de variables P_1, θ_1 et P_2, θ_2 .

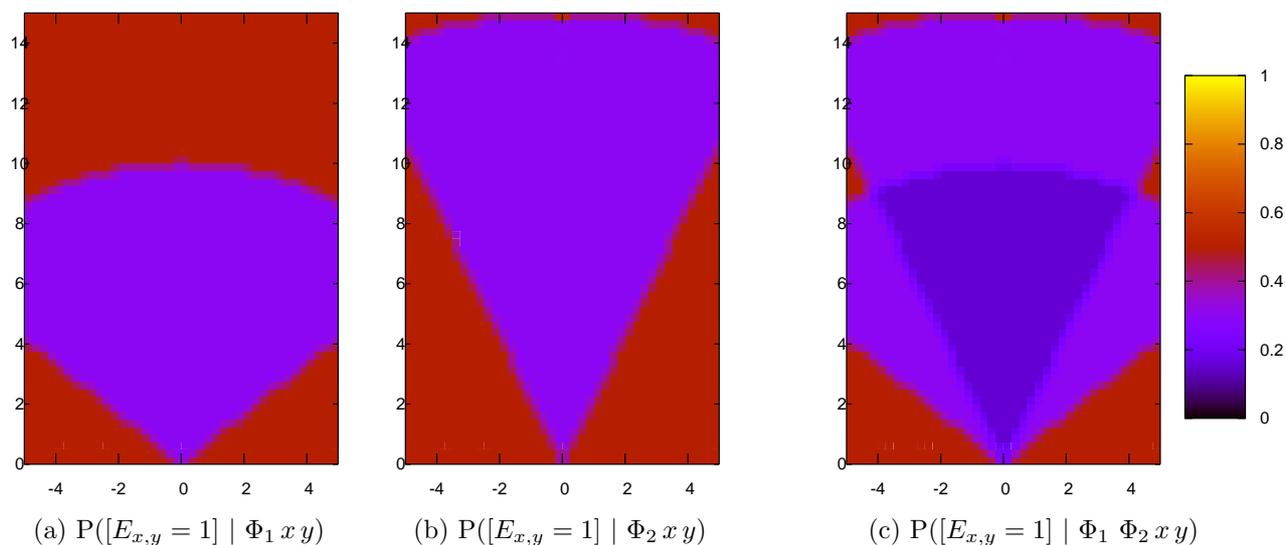


FIG. 3.13: Premier exemple : pas d'objet dans les champs de vue des capteurs.

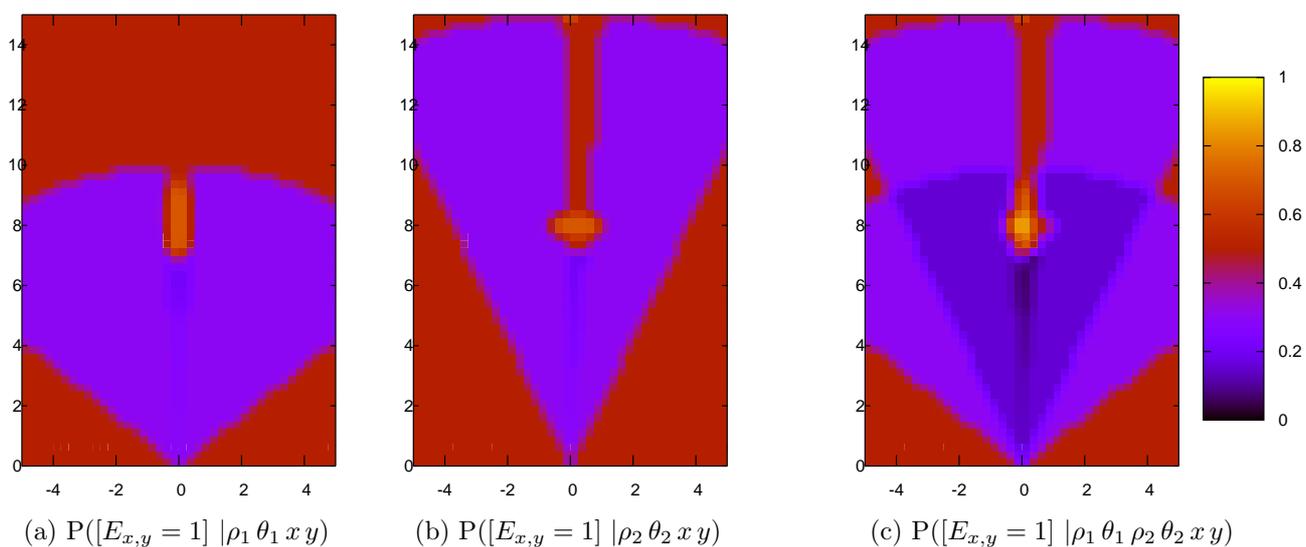


FIG. 3.14: La cible est détectée par les deux capteurs.

La figure 3.13 montre un premier exemple de grille d'occupation lorsqu'aucune détection n'a eu lieu. Sur cette figure, on voit à gauche l'estimation de la grille d'occupation en

utilisant uniquement les informations fournies par le premier capteur. On retrouve une grille d'occupation similaire à celle de la figure 3.8. Au centre, on voit l'estimation de la grille d'occupation en utilisant les informations fournies par le second capteur. Enfin à droite, on voit l'estimation de la grille d'occupation en fusionnant les informations fournies par les deux capteurs. Sur cette estimation, on retrouve le fait qu'en dehors des champs de vue des capteurs, on ne peut rien dire (probabilité d'occupation à 0.5). On remarque également que dans le champ de vue commun aux deux capteurs, la certitude que les cellules soient inoccupées est renforcée par rapport à l'utilisation d'un seul capteur (couleur plus sombre).

La figure 3.14 montre un exemple de grille d'occupation, un objet ayant cette fois-ci été détecté dans le champ de vue commun aux deux capteurs. La disposition est la même que pour la figure précédente : à gauche, estimation à partir des seules données du premier capteur, au centre, estimation à partir des seules données du second capteur et à droite, estimation en fusionnant les données de deux capteurs.

Dans cet exemple, le premier capteur est supposé être précis en gisement, mais pas en distance. Au contraire, le second capteur est supposé être précis en distance, mais pas en gisement. Ceci est traduit dans les fonctions de vraisemblance des modèles de chaque capteur ($\sigma_\rho = 0,1$; $\sigma_\theta = 0,5$ pour le premier capteur et $\sigma_\rho = 0,01$; $\sigma_\theta = 2$ pour le second).

Comme on pouvait s'y attendre, la fusion des données issues des deux capteurs permet d'obtenir une bonne précision à la fois en distance et en gisement. On remarque également une plus forte probabilité d'occupation pour la zone pour laquelle les deux capteurs sont d'accord pour dire qu'elle est occupée.

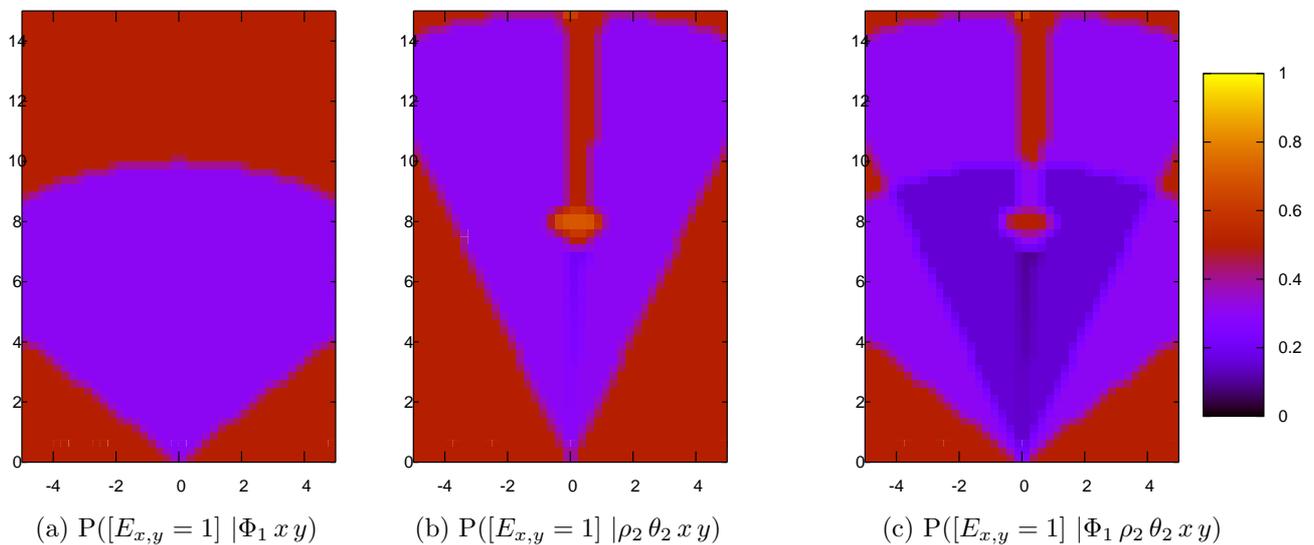


FIG. 3.15: Le capteur 1 manque la cible. La probabilité de non détection du capteur 1 et la probabilité de fausse alarme du capteur 2 sont du même ordre de grandeur.

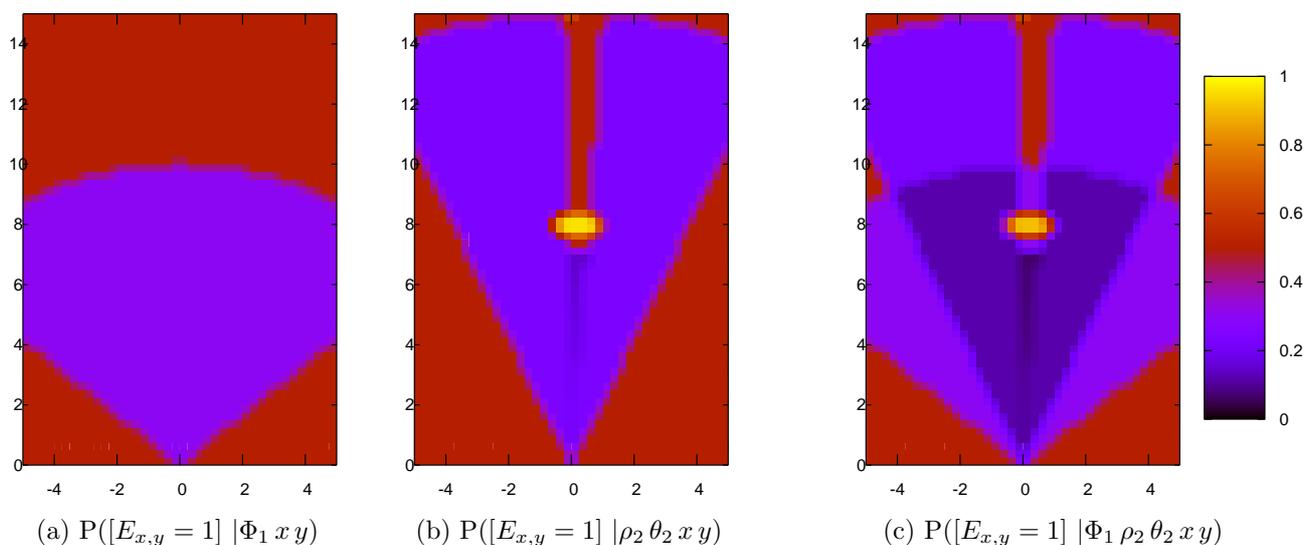


FIG. 3.16: Le capteur 1 manque la cible. La probabilité de non détection du capteur 1 est très supérieure à la probabilité de fausse alarme du capteur 2.

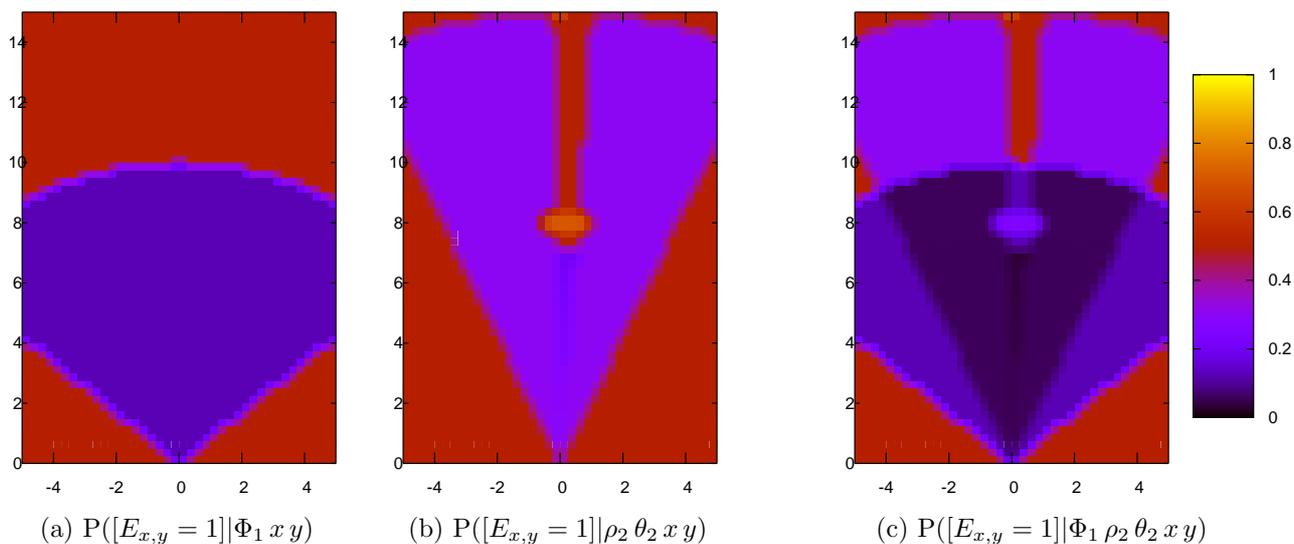


FIG. 3.17: Le capteur 1 manque la cible. La probabilité de non détection du capteur 1 est très inférieure à la probabilité de fausse alarme du capteur 2.

Les figures 3.15, 3.16 et 3.17 présentent trois fois le même cas dans lequel les deux capteurs ne sont pas d'accord : le premier capteur n'a rien détecté, le second capteur a détecté un objet, situé dans le champ de vue commun aux deux capteurs. Bien entendu, nous ne savons pas si un objet est réellement présent. Ces figures ont pour but d'illustrer l'importance de la prise en compte des probabilités de détection et de fausse alarme dans

le modèle capteur.

Dans le cas de la figure 3.15, la probabilité de fausse alarme $P_{FA}(\rho, \theta)$ du capteur 2 et la probabilité de non détection $P_{ND}(\rho, \theta)$ du capteur 1 sont du même ordre de grandeur. Si on ne considère que le premier capteur (à gauche), on retrouve bien une probabilité de présence faible dans tout le champ de vue de ce capteur. Si on ne considère que le second capteur, on obtient une tache de probabilité de présence supérieure à 0,5, correspondant à la détection d'un objet de ce capteur. En fusionnant les informations des deux capteurs, on obtient toujours une tache de probabilité d'occupation plus élevée dans le champ de vue commun aux deux capteurs, mais cette probabilité est très proche de 0,5. Il sera donc difficile de décider si un objet est présent dans cette zone ou non. Ceci correspond au fait qu'il est difficile de décider lequel des deux capteurs a raison : le premier capteur a-t-il manqué une cible ou le second capteur voit-il un fantôme ?

Dans le cas de la figure 3.16, la probabilité de fausse alarme $P_{FA}(\rho, \theta)$ du capteur 2 dans son champ de vue est très inférieure à la probabilité de non détection $P_{ND}(\rho, \theta)$ du capteur 1. Le plus probable est donc que le capteur 1 ait manqué la cible. La fusion se range donc du côté du capteur 2. Toutefois, on remarque que la probabilité d'occupation de la zone supposée occupée est légèrement plus faible que dans le cas où on ignore les informations du premier capteur (probabilité d'occupation égale à 0,90 au lieu de 0,96).

Enfin dans le cas de la figure 3.17, c'est la probabilité de fausse alarme du capteur 2 qui est très supérieure à la probabilité de non détection du capteur 1. Une tache est toujours présente dans la grille d'occupation estimée en fusionnant les informations des deux capteurs, mais la probabilité d'occupation dans cette zone reste très faible. Cette fois la fusion s'est donc rangée du côté du capteur 1.

3.3.4 Modélisation des capteurs : décalage temporel

Dans toutes nos modélisations, nous avons pour l'instant supposé que l'état d'une cellule E_X et l'observation d'un capteur Z correspondaient au même instant t . En pratique, cette hypothèse n'est pas vérifiée. L'observation d'un capteur n'est pas disponible immédiatement, du fait d'une part de la latence même du capteur et d'autre part du temps de transmission existant entre le capteur et l'unité de traitement. Il existe donc un décalage temporel entre une observation capteur et l'estimation de l'état du système.

Si ce décalage est trop important, il est nécessaire d'en tenir compte lors de l'estimation de notre grille d'occupation. Ceci est particulièrement vrai lorsque l'on fusionne des informations provenant de différents capteurs. En effet, les observations de chaque capteur correspondent à des instants différents lorsque les capteurs fonctionnent de manière asynchrone.

Nous présentons ici une adaptation du modèle capteur défini dans la section 3.2.2 lorsque le capteur fournit une observation correspondant à un instant t_s et l'estimation de l'état d'une cellule de la grille correspond à un instant t_g . Nous avons bien sûr $t_s < t_g$. Cette adaptation nécessite bien entendu de disposer d'une datation fiable des observations.

Nous notons alors Z^{t_s} la variable représentant l'observation du capteur, et X^{t_g} , $E_X^{t_g}$ les variables représentant la cellule de la grille et son état à l'instant t_g .

En notant X^{t_s} , $E_X^{t_s}$ les variables représentant l'état de la grille à l'instant correspondant à la mesure capteur, le modèle capteur défini dans la sous-section 3.2.2 fournit une forme paramétrique à la distribution de probabilité :

$$P(Z^{t_s} | E_X^{t_s} X^{t_s}).$$

Notre but ici est de nous appuyer sur ce modèle pour fournir une forme paramétrique à la distribution de probabilité :

$$P(Z^{t_s} | E_X^{t_g} X^{t_g}).$$

Nous allons donc avoir à considérer ici les variables Z^{t_s} , X^{t_s} , $E_X^{t_s}$, $E_X^{t_g}$ et X^{t_g} , et nous décomposons la distribution de probabilité conjointe sous la forme :

$$P(Z^{t_s} E_X^{t_s} X^{t_s} E_X^{t_g} X^{t_g}) = P(E_X^{t_g} X^{t_g})P(E_X^{t_s} X^{t_s} | E_X^{t_g} X^{t_g})P(Z^{t_s} | E_X^{t_s} X^{t_s}). \quad (3.23)$$

Nous associons aux termes apparaissant dans cette décomposition les formes paramétriques :

- $P(E_X^{t_g} X^{t_g})$ représente les connaissances *a priori* sur la cellule considérée et son état à l'instant t_g . Nous choisissons une distribution uniforme pour la représenter ;
- la forme paramétrique associée à $P(Z^{t_s} | E_X^{t_s} X^{t_s})$ est donnée par le modèle capteur défini dans la sous-section 3.2.2 ;
- enfin le terme $P(E_X^{t_s} X^{t_s} | E_X^{t_g} X^{t_g})$ représente nos connaissances sur l'état d'une cellule à l'instant t_s sachant l'état d'une autre cellule à l'instant t_g . Ceci correspond donc à un modèle dynamique du système remontant le temps.

Comme nous l'avons mentionné en introduction, notre grille est définie dans l'espace des états d'un acteur de la circulation. Dans nos illustrations précédentes, nous avons choisi que cet espace d'états ne représentait que la position relative de l'objet par rapport au véhicule, nos grilles d'occupation comportant par conséquent une ou deux dimensions. Dans le cas de scènes dynamiques, l'état d'un objet est donné par sa position et sa vitesse relative par rapport au véhicule. Par conséquent, notre grille d'occupation comporte quatre dimensions : deux pour la position et deux pour la vitesse. Nous donnerons des exemples d'estimation de telles grilles quand nous présenterons la filtre bayésien d'occupation.

Dans ce cas, une cellule X représente une position et une vitesse relative. Il est donc possible d'exprimer un lien entre l'état d'occupation d'une cellule à l'instant t_s et l'état d'occupation d'une autre cellule à l'instant t_g . Par exemple, si nous considérons que les objets circulent à vitesse constante, et si la cellule $(x^{t_g}, y^{t_g}, \dot{x}, \dot{y})$ est occupée à l'instant t_g , nous pouvons conclure que la cellule $(x^{t_g} - \dot{x}\Delta_t, y^{t_g} - \dot{y}\Delta_t, \dot{x}, \dot{y})$ était occupée à l'instant t_s . (ici nous avons bien sûr $\Delta_t = t_g - t_s$).

Notre description est maintenant complète, et nous lui posons la question :

$$P(Z^{t_s} | E_x^{t_g} x^{t_g}). \quad (3.24)$$

Pour cette question, le résultat de l'inférence donne :

$$P(Z^{t_s} | E_x^{t_g} x^{t_g}) = \frac{\sum_{\substack{E_X^{t_s} \\ X^{t_s}}} P(E_x^{t_g} x^{t_g}) P(E_X^{t_s} X^{t_s} | E_x^{t_g} x^{t_g}) P(Z^{t_s} | E_X^{t_s} X^{t_s})}{P(E_x^{t_g} x^{t_g})} \quad (3.25)$$

$$= \sum_{\substack{E_X^{t_s} \\ X^{t_s}}} P(E_X^{t_s} X^{t_s} | E_x^{t_g} x^{t_g}) P(Z^{t_s} | E_X^{t_s} X^{t_s}). \quad (3.26)$$

La forme paramétrique associée à la distribution $P(E_x^{t_g} x^{t_g})$ avait finalement peu d'importance, puisqu'elle disparaît lors de l'inférence.

Ainsi si le décalage temporel le justifie, le résultat de cette question doit être utilisé en lieu et place du modèle capteur classique lors de l'estimation de l'état d'occupation d'une cellule, que ce soit dans le cas monocapteur (voir figure 3.2) ou multicapteur (voir figure 3.10).

Afin de ne pas surcharger les notations, nous ne ferons plus de considérations ni sur le décalage temporel entre observation et estimation, ni sur les problèmes de synchronisation entre les différents capteurs.

3.4 Conclusion

Dans ce chapitre, nous avons introduit l'outil de représentation de l'environnement sur lequel est basée toute la suite de ce document : la grille d'occupation.

Nous avons d'abord présenté la méthode d'estimation de cette grille à partir d'une observation d'un unique capteur. L'extension au cas dans lequel plusieurs capteurs fournissent une observation a ensuite été présentée. Dans les deux cas, des résultats expérimentaux ont illustré un aspect fondamental de l'estimation d'une grille d'occupation : la modélisation des capteurs.

Dans ce chapitre, nous nous sommes pour l'instant restreint à l'estimation de la grille d'occupation à partir d'une observation unique de chaque capteur. La prise en compte simultanée de plusieurs observations, nécessaire lorsque plusieurs objets sont présents dans l'environnement, demande une légère adaptation, qui fait l'objet du chapitre suivant.

Chapitre 4

Estimation des grilles d'occupation : cas multiobjet

Dans le chapitre précédent, nous avons illustré l'estimation des grilles d'occupation sous l'hypothèse que chaque capteur fournissait au plus une observation. Dans le cas où plusieurs objets sont présents dans l'environnement modélisé par la grille, cette hypothèse ne peut être faite.

Nous présentons dans ce chapitre l'estimation de grilles d'occupation lorsque chaque capteur fournit un nombre quelconque d'observations.

Comme dans le chapitre précédent, nous commençons par exposer le cas mono-capteur avant de présenter le cas multicapteur.

4.1 Estimation d'une grille d'occupation à partir des observations d'un capteur

Nous présentons tout d'abord le programme bayésien pour estimer la probabilité d'occupation d'une cellule, avant de l'illustrer sur des grilles en deux dimensions.

4.1.1 Programme bayésien

4.1.1.1 Variables

Comme dans le cas mono-observation, nous avons à considérer les variables suivantes :

- la variable X indiquant la cellule de l'environnement pour laquelle nous cherchons à estimer la probabilité d'occupation ;
- la variable E_X , qui précise si la cellule X est occupée par un objet ou non.

Au lieu d'une observation capteur, nous considérons l'ensemble des O observations fournies par le capteur. Une variable Z_i est nécessaire pour chaque observation (nous rappelons que la conjonction de ces variables est notée \mathcal{Z}).

Enfin, afin de pouvoir considérer plusieurs observations du capteur pour mettre à jour la probabilité d'occupation d'une cellule, nous introduisons une nouvelle variable, notée M . Cette variable comporte $O + 1$ cas. Nous appelons cette variable « variable d'association ». Si $[M = 0]$, cela signifie qu'aucune observation capteur n'est considérée pour la mise à jour de la cellule. Si $[M = i]$, cela veut dire que la i -ème observation du capteur est utilisée.

Ces variables définissent la distribution de probabilité conjointe :

$$P(X E_X M \mathcal{Z}). \quad (4.1)$$

4.1.1.2 Décomposition

Des applications successives de la règle du produit permettent d'écrire :

$$P(X E_X M \mathcal{Z}) = P(X)P(E_X | X)P(M | E_X X)P(\mathcal{Z} | M E_X X). \quad (4.2)$$

Nous choisissons de n'exprimer aucune connaissance *a priori* sur les observations utilisées pour l'estimation de la probabilité d'occupation de la cellule X . Notre variable M est donc indépendante des variables X et E_X . Nous pouvons donc écrire :

$$P(M | E_X X) = P(M). \quad (4.3)$$

La distribution $P(\mathcal{Z} | M E_X X)$ est elle-même redécomposée en appliquant successivement la règle du produit :

$$P(\mathcal{Z} | M E_X X) = \prod_{i=1}^O P(Z_i | Z_1 \dots Z_{i-1} M E_X X). \quad (4.4)$$

Nous faisons à nouveau une importante hypothèse d'indépendance conditionnelle. Nous supposons que connaissant l'état de la cellule et la mesure avec laquelle nous travaillons (donnée par la valeur de M), aucune connaissance sur l'observation Z_i n'est apportée par les observations Z_1, \dots, Z_{i-1} . Nous pouvons donc écrire :

$$P(\mathcal{Z} | M E_X X) = \prod_{i=1}^O P(Z_i | M E_X X). \quad (4.5)$$

Finalement, en replongeant (4.3) et (4.5) dans (4.2), on obtient la décomposition suivante de la distribution de probabilité conjointe :

$$P(X E_X M \mathcal{Z}) = P(X)P(E_X | X)P(M) \prod_{i=1}^O P(Z_i | E_X X M) \quad (4.6)$$

Nous devons maintenant associer une forme paramétrique à chacune des distributions apparaissant dans cette décomposition.

4.1.1.3 Formes paramétriques

Pour $P(X)$ et $P(E_X | X)$, nous gardons les mêmes formes paramétriques que dans le cas mono-observation (voir la section 3.2.1), c'est à dire des distributions uniformes.

Nous choisissons de n'exprimer aucune connaissance sur le choix des observations à considérer pour estimer l'état de la cellule X . $P(M)$ est donc choisie uniforme.

Enfin la forme des distributions $P(Z_i | M E_X X)$ dépend de la valeur de la variable M :

- si $[M = i]$, cela signifie que nous considérons l'observation Z_i pour estimer l'état de la cellule X . La forme de $P(Z_i | [M = i] E_X X)$ est donc donnée par le modèle capteur ;
- si $[M \neq i]$, l'observation Z_i n'est pas considérée pour l'estimation de l'état de la cellule X . L'état de la cellule X ne nous donne donc aucune information sur l'observation Z_i . $P(Z_i | [M \neq i] E_X X)$ est alors représentée par une distribution uniforme.

La description est maintenant complète, nous pouvons l'utiliser au travers de questions.

4.1.1.4 Question

Le but de ce programme est d'estimer la probabilité d'occupation d'une cellule de la grille connaissant la liste des observations d'un capteur. Nous posons donc la question :

$$P(E_x | z_1 \dots z_O x). \quad (4.7)$$

Pour cette question, le résultat de l'inférence donne :

$$P(E_x | z_1 \dots z_O x) = \frac{\sum_{m=0}^O P(E_x | x z_1 \dots z_O [M = m])}{\sum_{E_x} \sum_{m=0}^O P(E_x | X z_1 \dots z_O [M = m])} \quad (4.8)$$

$$= \frac{1}{\alpha} \sum_{m=0}^O \left(P(x) P(E_x | x) P([M = m]) \prod_{i=1}^O P(z_i | [M = m] E_x x), \right) \quad (4.9)$$

où α est une constante de normalisation.

Pour simplifier cette expression, nous pouvons utiliser les connaissances fournies par les formes paramétriques. En particulier, les distributions $P(z_i | [M \neq i] E_x x)$ ont été choisies uniformes, ce qui permet d'écrire (en notant U_Z la distribution uniforme sur l'espace des observations Z) :

$$\prod_{i=1}^O P(z_i | [M = m] E_x x) = U_Z^{O-1} \times P(z_m | [M = m] E_x x), \text{ pour } m \neq 0. \quad (4.10)$$

et

$$\prod_{i=1}^O P(z_i | [M = 0] E_x x) = U_Z^O, \text{ pour } m = 0. \quad (4.11)$$

La valeur U_Z^{O-1} apparaissant alors dans chaque terme de la somme peut être sorti de cette somme. Nous l'intégrons dans la constante de normalisation α , ce qui permet d'écrire :

$$P(E_x | z_1 \dots z_O x) = \frac{1}{\alpha} P(E_x | x) \left(U_Z + \sum_{m=1}^O P(z_m | [M = m] E_x x) \right). \quad (4.12)$$

Le résultat de cette inférence permet de comprendre le rôle de la variable M . Elle permet de prendre en compte les différentes observations pour estimer l'état de la cellule. L'influence de chaque observation pour l'estimation de l'état de la cellule dépend de la distance entre l'observation et cette cellule.

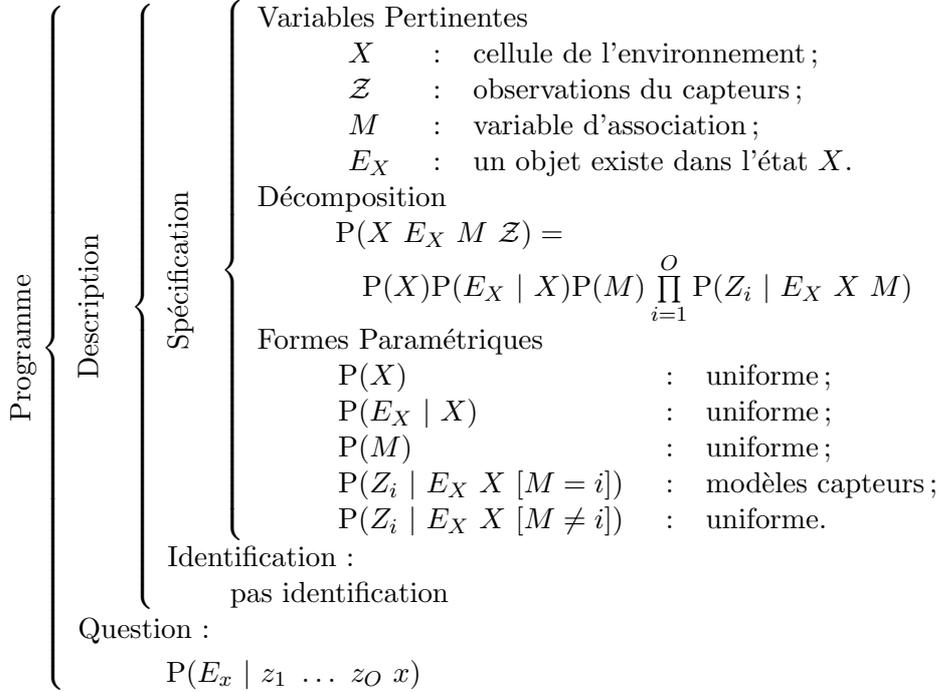


FIG. 4.1: Estimation d'une grille d'occupation à partir des observations d'un capteur : programme bayésien.

La figure 4.1 résume le programme bayésien que nous venons de détailler.

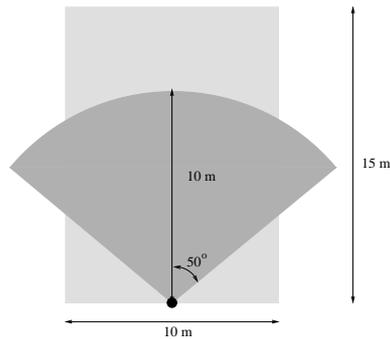


FIG. 4.2: Champ de vue du capteur (gris foncé), et zone couverte par la grille d'occupation (gris clair).

4.1.2 Illustration

Pour cette illustration, nous nous plaçons dans la même situation que dans la sous-section 3.2.3.2, que nous reprenons dans la figure 4.2. Les observations du capteur sont exprimées dans le repaire polaire, les cellules de la grille dans le repère cartésien.

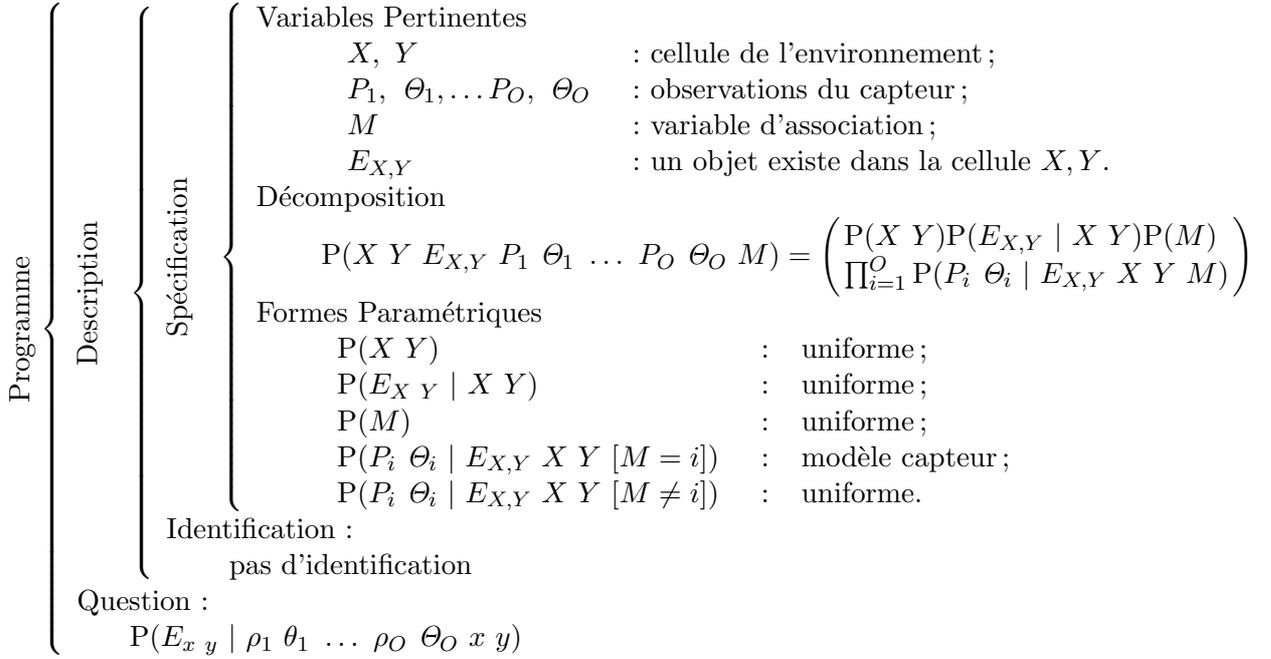


FIG. 4.3: Exemple d'estimation d'une grille d'occupation 2d à partir des observations d'un capteur : programme bayésien.

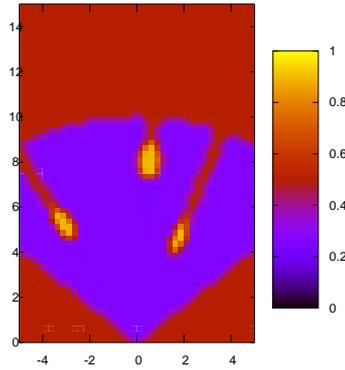


FIG. 4.4: Résultat de l'estimation d'une grille 2d à partir de plusieurs observations dans le cas mono-capteur.

La figure 4.3 présente l'adaptation du programme bayésien décrit dans la section précédente au cas dans lequel la cellule est représentée dans le repère cartésien X, Y et les observations du capteur sont représentées dans le repaire polaire P, θ .

La figure 4.4 présente le résultat de l'estimation d'une grille d'occupation à partir de trois observations du capteur. On retrouve sur cette figure les mêmes caractéristiques que dans le cas où le capteur retournait une unique observation : en dehors du champ de vue du capteur, la probabilité d'occupation des cellules est égale à 0,5 ; loin des observations (et dans le champ de vue du capteur), la probabilité d'occupation des cellules est inférieure à 0,5. La valeur de cette probabilité dépend des valeurs de probabilité de détection et de fausse alarme du capteur. Enfin on trouve une zone de forte probabilité d'occupation correspondant à chaque observation du capteur. La valeur de la probabilité d'occupation est égale dans chacune des trois zones. Comme dans le cas mono-observation, la probabilité d'occupation dans les zones masquées par chacune des observations est égale à 0,5.

4.2 Fusion de capteurs

Nous présentons tout d'abord le programme bayésien pour estimer la probabilité d'occupation d'une cellule à partir des observations de plusieurs capteurs, avant de l'illustrer sur des grilles en deux dimensions.

4.2.1 Programme bayésien

Nous présentons ici le programme dans un cadre très général, le nombre de capteurs fournissant des observations étant quelconque. Nous notons S ce nombre de capteurs. Chaque capteur retourne un nombre O_s d'observations, s étant le numéro du capteur. Il est donc compris entre 1 et S .

4.2.1.1 Variables

Nous devons ici considérer les observations des différents capteurs. Une observation particulière du capteur s est représentée par la variable Z_{s,i_s} , $i_s = 1 \dots O^s$. Nous avons donc un total de $\sum_{s=1}^S O^s$ variables pour les différentes mesures capteurs. Nous rappelons que la conjonction de ces variables est notée $Z_{1:S}$.

Comme dans le cas mono-capteur, nous ajoutons une variable d'association par capteur. Ces variables seront notées M_s , s désignant le capteur.

Enfin les variables X et E_X sont toujours pertinentes dans le cas multicapteur.

Ces variables définissent la distribution de probabilité conjointe :

$$P(X \ E_X \ Z_{1:S} \ M_1 \ \dots \ M_S). \quad (4.13)$$

4.2.1.2 Décomposition

Pour décomposer la distribution de probabilité conjointe, nous faisons l'hypothèse que les variables d'association des différents capteurs sont indépendantes.

Nous faisons également deux hypothèses d'indépendance conditionnelle :

- les observations de capteurs différents sont indépendantes conditionnellement à la connaissance de l'état de la cellule X (nous avons discuté de cette hypothèse dans la section 3.3);
- les observations d'un même capteur sont indépendantes conditionnellement à la connaissance de l'état de la cellule et de la variable d'association M (nous avons discuté de cette hypothèse dans la section précédente).

Grâce à ces différentes hypothèses, nous décomposons la distribution de probabilité conjointe (4.13) sous la forme :

$$P(X, E_X, M_1, \dots, M_S, Z_{1:S}) = P(X)P(E_X | X) \prod_{s=1}^S \left(P(M_s) \prod_{i_s=1}^{O_s} P(Z_{s,i_s} | E_X, X, M_s) \right) \quad (4.14)$$

4.2.1.3 Formes paramétriques

Les formes paramétriques choisies sont les mêmes que pour le cas mono-capteur. En particulier la forme de $P(Z_{s,i_s} | E_X, X, M_s)$ dépend de la valeur de M_s :

- si $[M_s = i_s]$, cela signifie que nous considérons l'observation Z_{s,i_s} pour estimer l'état de la cellule X . La forme de $P(Z_{s,i_s} | [M_s = i_s], E_X, X)$ est donc donnée par le modèle capteur associé au capteur s ;
- si $[M_s \neq i_s]$, l'observation Z_{s,i_s} n'est pas considérée pour l'estimation de l'état de la cellule X . L'état d'occupation de la cellule X ne nous donne donc aucune information sur l'observation Z_{s,i_s} . $P(Z_{s,i_s} | [M_s \neq i_s], E_X, X)$ est alors représentée par une distribution uniforme.

Chaque distribution $P(M_s)$ est choisie uniforme.

La description est maintenant complète, nous pouvons l'utiliser au travers de questions.

4.2.1.4 Question

Le but de ce programme est d'estimer la probabilité d'occupation d'une cellule de la grille connaissant la liste des observations des différents capteurs. Nous posons donc la question :

$$P(E_x | z_{1,1}, \dots, z_{S,O_S}, x). \quad (4.15)$$

Pour cette question, le résultat de l'inférence donne :

$$P(E_x | z_{1,1}, \dots, z_{S,O_S}, x) = \frac{1}{\alpha} \sum_{m_1, \dots, m_S} \left(\frac{P(x)P(E_x | x)}{\prod_{s=1}^S \left(P([M_s = m_s]) \prod_{i_s=1}^{O_s} P(z_{s,i_s} | E_x, x, [M_s = m_s]) \right)} \right) \quad (4.16)$$

Pour simplifier cette inférence, nous utilisons là encore les formes paramétriques des termes de la décomposition :

- $P(x)$ a été choisie uniforme, sa valeur peut donc être sortie de la somme et intégrée à la constante de normalisation α ;

- les distributions $P(M_s)$ ont également été choisies uniformes. Puisque O_s désigne le nombre d'observations du capteur s , nous avons $P([M_s = m_s]) = (O_s + 1)^{-1}$. Nous notons cette valeur β_s ;
- les distributions $P(z_{s,i_s} | E_x x [M_s \neq i_s])$ ont été choisies uniformes, nous pouvons donc faire les mêmes simplifications que celles données par les expressions (4.10) et (4.11).

Le résultat de l'inférence devient :

$$P(E_x | z_{1,1} \dots z_{S,O_S} x) = \frac{P(E_x | x)}{\alpha} \left(\sum_{m_1 \dots m_S} \prod_{s=1}^S \beta_s \left(U_Z^{O_s} + U_Z^{O_s} \cdot P(z_{s,m_s} | E_x x [M_s = m_s]) \right) \right) \quad (4.17)$$

En marge de l'estimation de l'état de chaque cellule, une autre question intéressante peut être posée à notre description. Il s'agit de :

$$P(M_1 \dots M_S | z_{1,1} \dots z_{S,O_S}). \quad (4.18)$$

Cette question estime la probabilité sur la conjonction des variables M_i sachant les différentes observations des différents capteurs. La valeur de $P(m_1 \dots m_S | z_{1,1} \dots z_{S,O_S})$ peut donc être interprétée comme la cohérence entre les observations m_i des différents capteurs. Plus la valeur de cette probabilité est grande, plus les observations proviennent d'une même zone de l'espace. On peut donc en déduire que les observations des différents capteurs correspondent à la détection d'un même objet. Cette question peut donc être vue comme la résolution du problème d'association entre les observations des différents capteurs.

Le résultat de l'inférence pour cette question donne :

$$\begin{aligned} P(M_1 \dots M_S | z_{1,1} \dots z_{S,O_S}) &= \frac{1}{\alpha} \sum_{X, E_X} \left(\frac{P(x)P(E_X | X)}{\prod_{s=1}^S P(M_s) \prod_{i_s=1}^{O_s} P(z_{i_s,s} | M_s E_X X)} \right) \\ &= \frac{1}{\alpha} \sum_{X, E_X} \left(P(E_X | X) \prod_{s=1}^S \beta_s \prod_{i_s=1}^{O_s} P(z_{i_s,s} | M_s E_X X) \right) \end{aligned}$$

Pour discuter des résultats expérimentaux, nous utiliserons ces deux questions : estimation de la probabilité d'occupation d'une cellule et estimation de la probabilité des associations des observations.

La figure 4.5 résume le programme bayésien que nous venons de détailler.

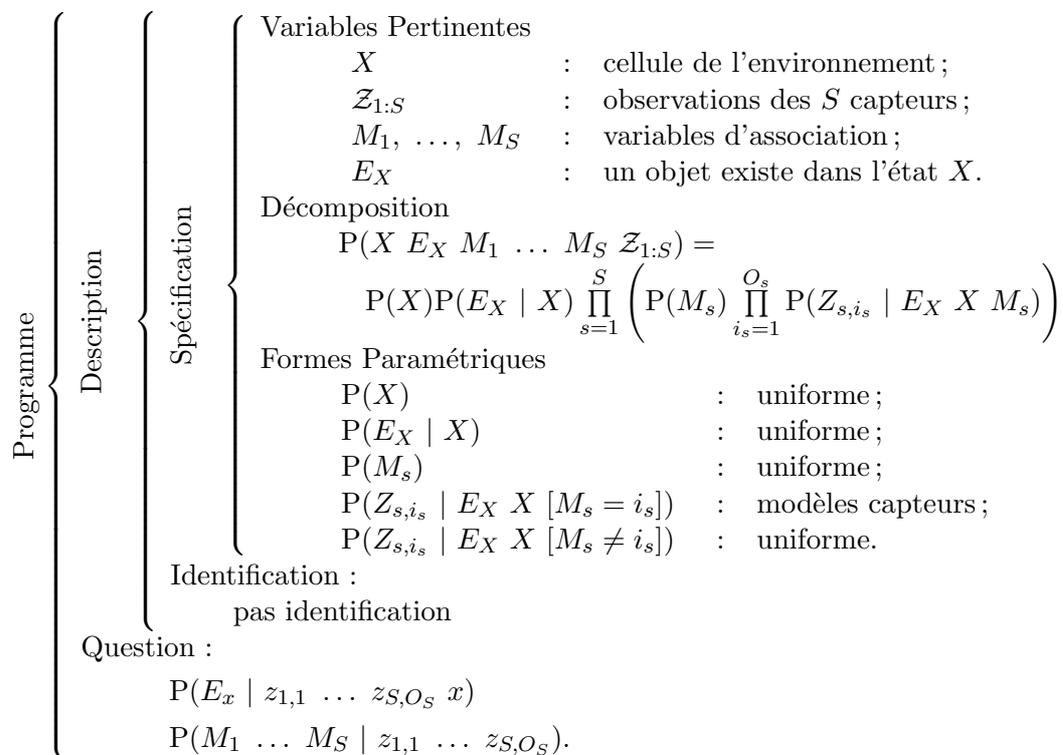


FIG. 4.5: Estimation d'une grille d'occupation à partir des observations de plusieurs capteurs : programme bayésien.

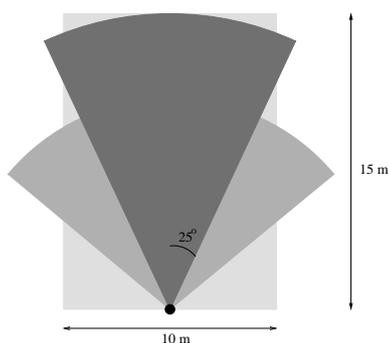


FIG. 4.6: Champ de vue du deuxième capteur (foncé), ajouté à la situation décrite dans la figure 4.2.

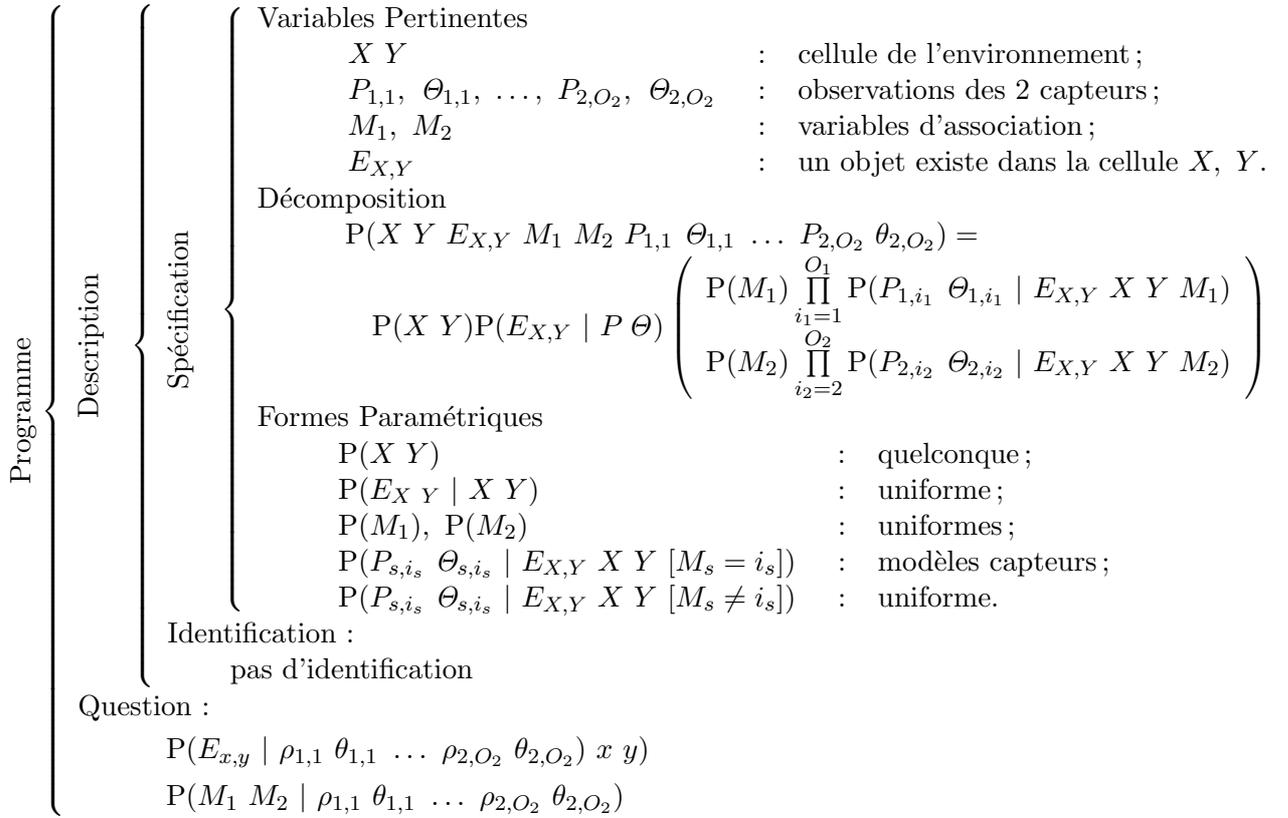


FIG. 4.7: Estimation d'une grille d'occupation à partir des observations de plusieurs capteurs dans le cas multiobjet : programme bayésien.

4.2.2 Illustration

Pour illustrer l'association et l'estimation de la grille d'occupation quand plusieurs objets sont présents dans l'environnement du véhicule, nous nous plaçons dans la même situation que celle utilisée pour la sous-section 3.3.3. La figure 4.6 reprend cette situation.

L'adaptation du programme bayésien décrit dans la partie précédente à ce cas particulier est résumée par la figure 4.7. Une cellule est représentée dans le repère cartésien X, Y , les observations des différents capteurs sont représentées dans le repère polaire P, θ . Nous ne détaillons pas ce programme.

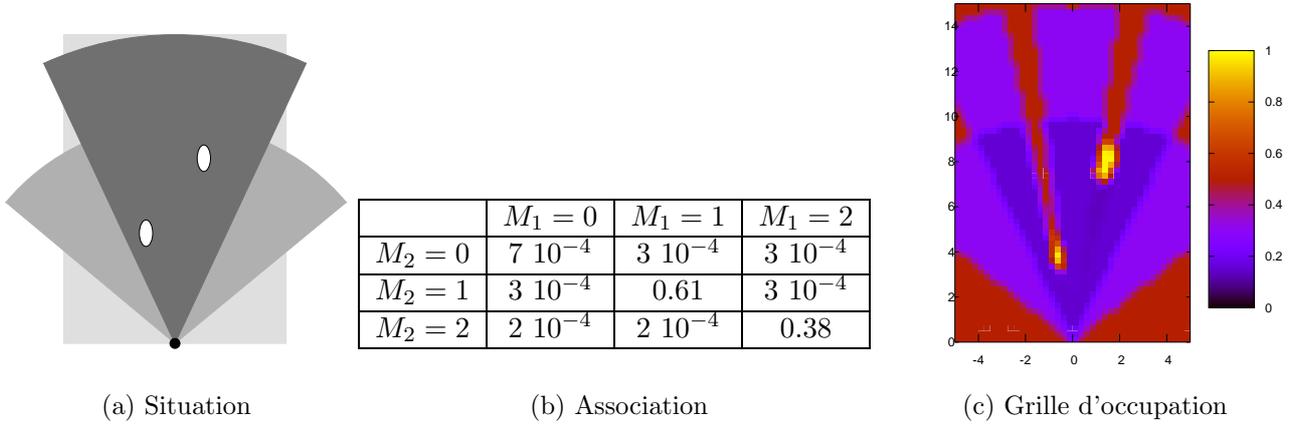


FIG. 4.8: Premier exemple d'association et d'estimation.

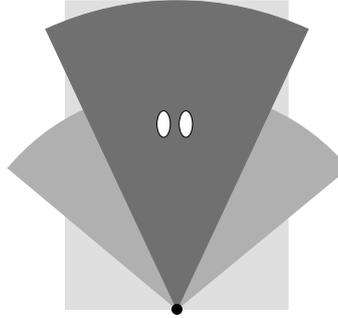


FIG. 4.9: Positions des objets dans l'expérience destinée à illustrer la sensibilité de l'association et de l'estimation à la précision des capteurs.

La figure 4.8 montre un premier exemple d'association et de fusion, impliquant les capteurs et deux objets. A gauche, on peut voir une vue d'oiseau de la situation. Les champs de vue des deux capteurs sont représentés par des surfaces grisées, les objets par des ellipses blanches. Dans cette situation, les deux objets sont situés dans le champ de vue commun aux deux capteurs. Ils sont tous les deux détectés par les deux capteurs, donc chaque capteur renvoie deux observations.

Le tableau présenté au centre de la figure 4.8 montre le résultat de l'association, c'est à dire l'estimation de la distribution de probabilité $P(M_1 M_2 | \rho_{1,1} \theta_{1,1} \rho_{1,2} \theta_{1,2} \rho_{2,1} \theta_{2,1} \rho_{2,2} \theta_{2,2})$. Puisque chaque capteur renvoie deux observations, M_1 et M_2 peuvent tous les deux prendre trois valeurs, 0, 1 ou 2. Le tableau présente donc pour chaque valeur possible du couple (M_1, M_2) la probabilité que ces deux mesures soient issues du même objet. Dans la situation présente, les objets sont situés suffisamment loin l'un de l'autre par rapport à la précision des capteurs. On voit donc clairement apparaître deux valeurs de probabilité très supérieures aux autres, correspondant à la meilleure association possible. Si l'on avait à choisir une association, on répondrait sans hésiter : $(M_1 = 1, M_2 = 1)$, $(M_1 = 2, M_2 = 2)$.

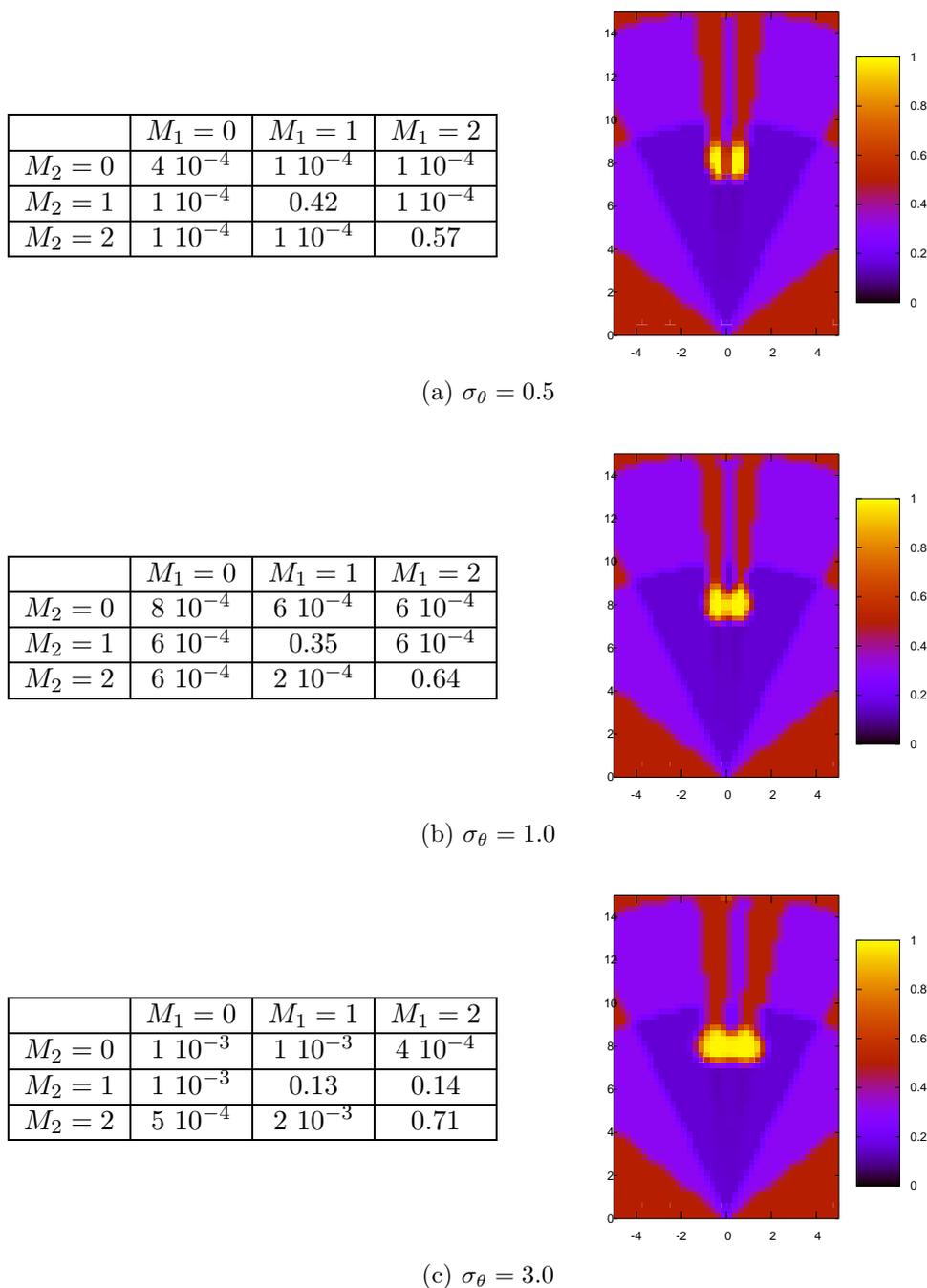


FIG. 4.10: Résultats d'association et d'estimation pour différentes précisions angulaires des capteurs.

A droite de ce tableau sur la figure 4.8 est présentée l'estimation de la grille d'occupation, basée sur les mêmes mesures capteurs. Nous rappelons que cette estimation se fait sans prendre de décision sur l'association des mesures provenant des différents capteurs.

Comme dans le cas mono-objet, on retrouve le fait qu'en dehors des champs de vue des capteurs, la probabilité d'occupation des cellules est égale à 0,5. On ne peut donc rien dire sur l'occupation de ces zones. La probabilité d'occupation dans le champ de vue des capteurs loin des mesures de ceux-ci est inférieure à 0,5, ce qui signifie que l'on pense qu'il n'y a pas d'objet dans ces zones. Cette conviction est renforcée dans la zone observée par les deux capteurs. Enfin on voit clairement apparaître deux « taches » de probabilité d'occupation forte, chacune correspondant à l'un des objets présent devant le véhicule. On voit donc qu'il est possible d'estimer la grille d'occupation sans chercher à résoudre le problème d'association des mesures capteurs.

Au vu des résultats de la fusion présentés dans la sous-section 3.3.3, il est raisonnable de penser que les résultats de l'association et de l'estimation de la grille d'occupation sont influencés par la précision des capteurs.

Pour illustrer cette sensibilité à la précision du capteur, nous avons mené plusieurs expériences sur la situation décrite par la figure 4.9. Dans cette situation, deux cibles sont placées très proches l'une de l'autre, quelques mètres devant le véhicule, dans le champ de vue commun aux deux capteurs. Dans tous les résultats suivants, les deux cibles sont détectées par les deux capteurs.

Pour ces différentes expériences, les précisions en distance des deux capteurs sont constantes. Nous jouons sur la précision angulaire des capteurs, les problèmes d'association et d'estimation de la grille d'occupation étant résolus pour chaque valeur de cette précision angulaire. Ces différentes précisions angulaires sont obtenues en modifiant l'écart-type des gaussiennes représentant la fonction de vraisemblance associée à chaque capteur. Les précisions angulaires des deux capteurs sont considérées égales, et notées σ_θ .

La figure 4.10 donne les résultats de l'association et de l'estimation de la grille d'existence pour différentes précisions angulaires des deux capteurs. Les tableaux de gauche présentent les résultats de l'association. A droite sont présentées les différentes estimations de la grille d'occupation.

En haut, les capteurs ont une bonne précision angulaire. Concernant l'association, on voit apparaître une association évidente : $(M_1 = 1, M_2 = 1)$, $(M_1 = 2, M_2 = 2)$. Sur la grille d'occupation, on réussit à distinguer deux taches de probabilité d'occupation forte. Il sera donc possible de décider, à condition de vouloir prendre cette décision, que deux objets sont présents dans l'environnement.

Lorsque la précision angulaire des capteurs diminue, l'association devient de moins en moins évidente. Ainsi dans le tableau en bas à gauche, les associations $(M_1 = 1, M_2 = 1)$ et $(M_1 = 1, M_2 = 2)$ sont quasiment équiprobables. Décider de l'association est donc plus difficile que dans le cas précédent. Bien sûr, si on veut qu'une mesure du capteur 1 corresponde au plus à une mesure du capteur 2, il semble préférable de choisir l'association $(M_1 = 1, M_2 = 1)$, $(M_1 = 2, M_2 = 2)$, compte tenu de la probabilité de l'association $(M_1 = 2, M_2 = 2)$ très élevée.

Concernant l'estimation de la grille d'occupation, on voit que lorsque la précision angulaire des capteurs diminue, il devient de plus en plus difficile de distinguer deux zones. Dans le cas extrême présenté en bas à droite, on ne voit plus qu'une seule grande « tache » de probabilité de présence importante. En ne considérant que cette grille d'occupation, il

est donc difficile de conclure que deux objets sont présents devant le véhicule.

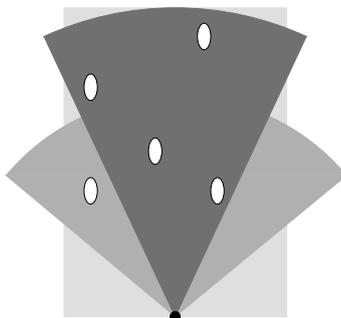
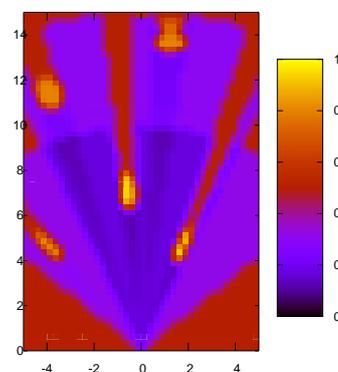


FIG. 4.11: Position des cinq objets dans l'environnement du véhicule.

	$M_1 = 0$	$M_1 = 1$	$M_1 = 2$	$M_1 = 3$
$M_2 = 0$	5.10^{-4}	5.10^{-4}	2.10^{-4}	2.10^{-4}
$M_2 = 1$	4.10^{-4}	8.10^{-5}	1.10^{-4}	1.10^{-4}
$M_2 = 2$	2.10^{-4}	1.10^{-4}	0,61	2.10^{-4}
$M_2 = 3$	2.10^{-4}	6.10^{-5}	1.10^{-4}	1.10^{-4}
$M_2 = 4$	2.10^{-4}	1.10^{-4}	2.10^{-4}	0,38

(a) Association



(b) Grille d'occupation

FIG. 4.12: Résultats de l'association (à gauche) et de l'estimation de la grille d'occupation.

Pour l'instant, nous nous sommes contentés de montrer des exemples d'association et d'estimation relativement simples : deux objets seulement étaient présents devant le véhicule, dans le champ de vue commun aux deux capteurs, ce qui rendait le problème d'association simple.

La figure 4.11 montre une situation plus complexe. Dans cet exemple, cinq objets sont présents devant le véhicule, et ne sont pas tous situés dans le champ de vue commun aux deux capteurs. Deux objets seulement sont dans ce champ de vue commun, un objet n'est présent que dans le champ de vue du premier capteur, et deux objets ne sont présent que dans le champ de vue du deuxième capteur. Pour notre expérience, tous les objets ont été détectés. Le premier capteur renvoie donc une liste de trois observations, le second une liste de quatre observations. Les observations de chaque capteur sont numérotées de gauche à droite.

La figure 4.12 présente les résultats de l'association et de l'estimation dans cette situation. A gauche, le résultat de l'association, à droite, le résultat de l'estimation de la grille d'occupation.

Concernant l'association, on remarque deux associations évidentes, $(M_1 = 2, M_2 = 2)$ et $(M_1 = 3, M_2 = 4)$ qui totalisent à elles deux 99% de la probabilité d'association. Ces deux associations correspondent aux mesures des deux objets détectés par les deux capteurs. Les autres associations sont moins évidentes. Cependant, en cherchant à optimiser la somme des probabilités sur l'association, et en considérant que seule la « mesure vide » d'un capteur peut être associée à plusieurs mesures de l'autre capteur, on décidera probablement pour les associations : $(M_1 = 0, M_2 = 1)$, $(M_1 = 0, M_2 = 3)$, et $(M_1 = 1, M_2 = 0)$. On remarque donc ici que la décision sur les associations peut ne pas être facile. Mais dans notre approche des grilles d'occupation, cette décision n'a pas à être prise.

Concernant l'estimation de la grille, on remarque bien cinq « taches » de probabilité forte d'occupation. On remarque que dans le cas où des observations des deux capteurs concordent, les probabilités d'occupation des cellules correspondantes sont légèrement plus fortes (correspondant à une couleur plus claire) que dans le cas où les objets ne sont détectés que par un seul capteur. Les résultats de l'estimation paraissent donc moins ambigus que les résultats de l'association.

4.3 Conclusion sur l'estimation des grilles d'occupation

Dans ce chapitre, nous avons présenté la méthode d'estimation des grilles d'occupation lorsque plusieurs objets sont présents autour du véhicule, et donc lorsque le ou les capteurs fournissent plusieurs observations.

Les résultats expérimentaux présentés ont montré l'importance de la modélisation des capteurs sur le résultat final obtenu.

Nous insistons ici sur le fait que la notion d'objet n'est pas présente dans les grilles d'occupation. Quand nous parlons de cinq taches correspondant aux cinq objets présents, il s'agissait uniquement d'une interprétation du résultat connaissant le protocole de test.

Pour l'instant, nous estimons nos grilles d'occupation de manière statique, c'est à dire uniquement à partir des dernières observations capteur disponibles, et sans tenir compte de la dynamique de la situation. Cette estimation statique ne peut être satisfaisante dans un environnement dynamique et encombré, comme l'illustre la section suivante.

4.4 Limites de l'estimation statique des grilles

Pour terminer ce chapitre, nous présentons dans la figure 4.13 une séquence d'estimation de grille d'occupation, pour une scène dynamique. Dans cet exemple, la grille utilisée est de dimension 4 : elle modélise la position relative de l'objet au véhicule et la vitesse relative de l'objet par rapport au véhicule. Nous supposons que le capteur est capable de mesurer cette position et cette vitesse. La première ligne de la figure 4.13 présente l'évolution de la situation : le véhicule portant le capteur est immobile, deux objets mobiles se croisent devant lui. Dans les cas des figures 4.13(a), 4.13(b), et 4.13(d), les deux objets sont visibles

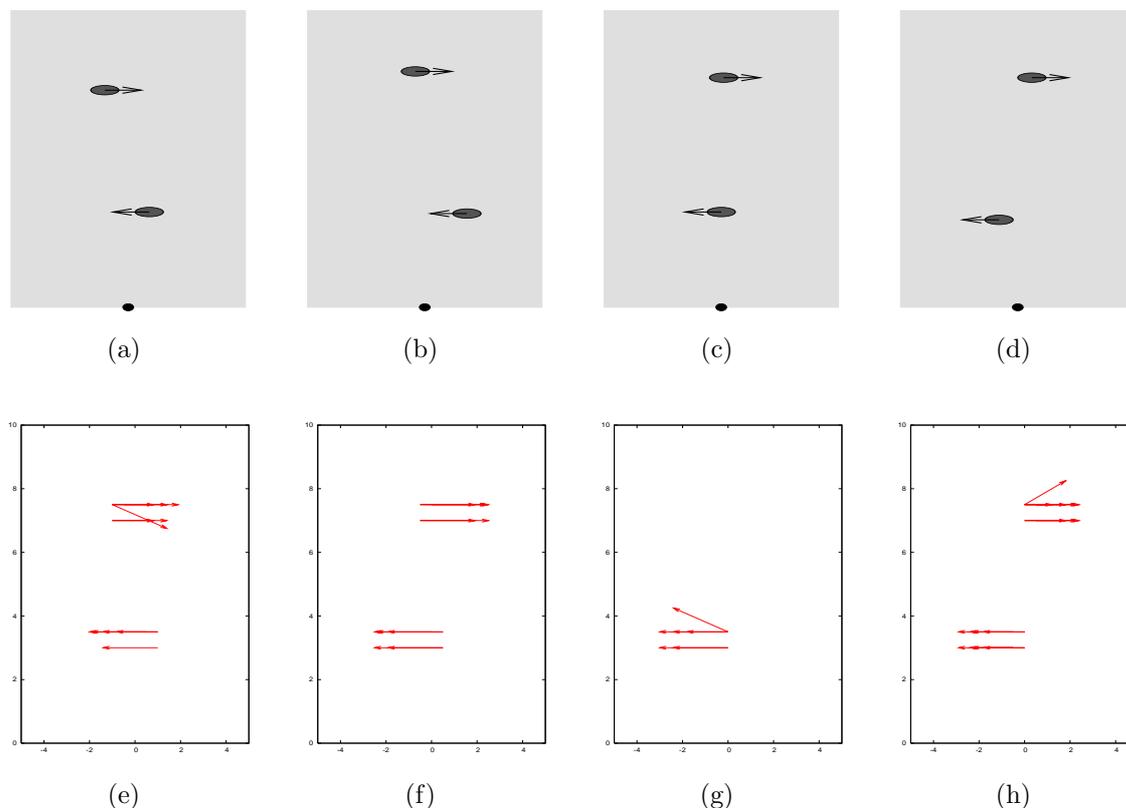


FIG. 4.13: Exemple de scène dynamique. L'estimation statique de la grille ne permet pas de gérer les occultations.

et détectés par le capteur. Dans le cas de la figure 4.13(c), l'objet le plus proche du véhicule cache à celui-ci le deuxième objet. Un unique objet est donc détecté par le capteur.

La seconde ligne présente les résultats de l'estimation de la grille à chaque instant. Comme notre grille est de dimension 4, il est difficile de tout représenter. Nous avons donc choisi de ne représenter dans ces figures que les cellules de la grille ayant une probabilité d'occupation supérieure à 0.5. Dans ce cas, une flèche représente la cellule de la grille : la flèche a pour origine la position, sa direction est celle de la vitesse relative et sa longueur est proportionnelle à la valeur de cette vitesse.

Pour les figures 4.13(e), 4.13(f), on distingue bien deux groupes de flèches, correspondant à chacun des deux objets. Pour le cas de la figure 4.13(g), un seul groupe de flèches est visible. En effet, puisque le deuxième objet n'est pas visible et donc pas détecté par le capteur, il n'est pas présent dans la grille finale. Enfin pour la figure 4.13(h), deux zones de fortes probabilité d'occupation sont à nouveau présentes dans la grille.

Cet exemple montre que l'estimation de la grille en tenant compte des seules dernières observations capteur ne permet pas d'être robuste aux occultations d'objets. De plus, elle n'est pas robuste aux pannes des capteurs et ne peut rien dire en dehors du champ d'obser-

vation des capteurs. Il est donc nécessaire de tenir compte de l'historique des observations capteurs, et de la dynamique de la scène. C'est le but des deux chapitres suivants.

Chapitre 5

Le filtre d'occupation bayésien

Comme nous l'avons signalé en conclusion du chapitre précédent, une bonne représentation d'un environnement dynamique se doit d'être robuste aux défaillances des capteurs et aux occultations d'objets. Afin d'assurer cette robustesse, on ne peut pas estimer la grille d'occupation de l'environnement du véhicule en ne tenant compte que des dernières mesures capteurs disponibles. On doit donc impérativement prendre en compte l'historique des données capteurs.

Le but de ce chapitre est de présenter notre méthode pour estimer la grille d'occupation en tenant compte de l'historique des observations et de la dynamique de la scène.

Nous commençons par présenter l'outil utilisé, le filtre bayésien, de manière théorique. Ensuite nous présentons différentes implantations de ce filtre, qui par leur efficacité, ont contribué à le rendre populaire.

Nous présentons ensuite la contribution principale de cette thèse : le rapprochement du filtre bayésien et de la grille d'occupation. Nous appelons *filtre d'occupation bayésien* la méthode issue de ce rapprochement. Cette présentation inclut un algorithme d'approximation spécifique pour répondre au problème du coût calculatoire.

Il faut noter que dans ce chapitre et dans le suivant, nous reprenons la même progression que dans les chapitres 3 et 4. Nous considérons donc ici le cas où le ou les capteurs fournissent au plus une observation. Le cas multiobjet sera considéré dans le chapitre suivant.

5.1 Le filtre bayésien

Estimer l'état d'un système dynamique à partir d'un ensemble d'observations bruitées est un problème classique. L'approche probabiliste de ce problème [Jaz70] a débouché sur la notion de *filtre Bayésien*.

Dans cette approche, le système dynamique dont on veut estimer l'état est supposé être un système de Markov d'ordre 1. Cela signifie qu'entre deux pas de temps consécutifs, l'évolution du système est modélisable par une fonction bruitée :

$$S^k = f(S^{k-1}, w^k). \quad (5.1)$$

à notre connaissance utilisées dans des applications de poursuite, nous ne les détaillerons pas ici.

Ces dernières années, grâce à des implantations efficaces, le filtre bayésien, dans sa version discrète, est devenu très populaire, non seulement pour la poursuite, mais également dans nombre d'applications robotiques, comme la construction de carte [Thr02] et la localisation [Cro89][ATS01].

Nous ne faisons pour l'instant aucune supposition sur le support des variables aléatoires représentant les observations et l'état du système. La notation $P(S)$ pourra donc représenter la distribution de probabilité d'une variable discrète ou par abus de notation, la densité de probabilité d'une variable continue.

5.1.1 Programme bayésien

Nous présentons ici le filtre Bayésien sous sa forme la plus générale, en utilisant bien entendu un programme bayésien. Nous rappelons que le but de ce programme est d'implanter la boucle *prédiction-estimation* décrite sur la figure 5.1.

5.1.1.1 Variables

Nous nous plaçons sur un horizon de temps discret. Un « instant » sur cet horizon de temps est représenté par un entier k , que nous placerons en exposant. Notre horizon de temps s'étend depuis un instant initial 0 jusqu'à un instant final noté N .

Nous représentons l'état du système à un instant k , $k = 0 \dots N$ par une variable S^k . L'observation capteur à chaque instant k est représentée par la variable Z^k .

Afin de faciliter l'écriture et la lecture, nous notons $Z^{0:k}$ la conjonction des variables d'observation entre l'instant initial et l'instant k : $\{Z^0 \dots Z^k\}$. De la même manière, nous notons $S^{0:k}$ la conjonction $\{S^0 \dots S^k\}$.

Ces variables définissent la distribution de probabilité conjointe :

$$P(Z^0 \dots Z^N S^0 \dots S^N) = P(S^{0:N} Z^{0:N}). \quad (5.3)$$

5.1.1.2 Décomposition

Le but de cette décomposition est de faire apparaître comme distributions de probabilité élémentaires le modèle dynamique et le modèle capteur, c'est à dire les distributions de probabilités $P(S^k | S^{k-1})$ et $P(Z^k | S^k)$, k variant entre 0 et N .

En appliquant une première fois la règle du produit à la distribution de probabilité conjointe (5.3), on obtient la décomposition :

$$P(S^{0:N} Z^{0:N}) = P(S^{0:N-1} Z^{0:N-1})P(S^N | S^{0:N-1} Z^{0:N-1})P(Z^N | S^{0:N} Z^{0:N-1}). \quad (5.4)$$

Comme nous l'avons mentionné en introduction du filtre bayésien, le système dynamique est supposé être un système de Markov d'ordre 1. La connaissance de l'état du

système à l'instant $N - 1$ suffit donc à l'estimation de l'état du système à l'instant N . Cela s'écrit :

$$P(S^N | S^{0:N-1} Z^{N-1}) = P(S^N | S^{N-1}). \quad (5.5)$$

De plus, on fait l'hypothèse que la connaissance de l'état du système à l'instant N suffit à modéliser la réponse du capteur à ce même instant, indépendamment des états du système et des observations aux instants précédents. Cela se traduit :

$$P(Z^N | S^{0:N} Z^{0:N-1}) = P(Z^N | S^N). \quad (5.6)$$

La décomposition (5.4) devient donc :

$$P(S^{0:N} Z^{0:N}) = P(S^{0:N-1} Z^{0:N-1})P(S^N | S^{N-1})P(Z^N | S^N). \quad (5.7)$$

Pour obtenir la décomposition finale, on applique de manière récursive le même raisonnement à la distribution de probabilité $P(S^{0:N-1} Z^{0:N-1})$. On obtient finalement la décomposition suivante de notre distribution de probabilité conjointe (5.3) :

$$P(S^{0:N} Z^{0:N}) = P(S^0)P(Z^0 | S^0) \prod_{k=1}^N P(S^k | S^{k-1})P(Z^k | S^k). \quad (5.8)$$

5.1.1.3 Formes paramétriques

Dans la décomposition (5.8), quasiment toutes les distributions de probabilité élémentaires sont données par les modèles sur lesquels est basé le filtre bayésien.

Le modèle dynamique du système, défini par l'équation (5.1), permet d'associer une forme paramétrique à chacune des distributions de probabilité $P(S^k | S^{k-1})$, $k = 1 \dots N$.

Le modèle capteur, défini par l'équation (5.2), permet d'associer une forme paramétrique à chacune des distributions de probabilité $P(Z^k | S^k)$, $k = 0 \dots N$.

Il reste à associer une forme paramétrique à la distribution de probabilité $P(S^0)$. Cette distribution représente la connaissance dont nous disposons sur l'état du système avant la première mesure capteur. Si aucun *a priori* n'est connu, cette distribution de probabilité est choisie uniforme.

La description est maintenant terminée, nous pouvons l'utiliser au moyen de questions.

5.1.1.4 Questions

Afin de réaliser la boucle décrite par la figure 5.1, deux questions sont posées à cette description à chaque instant k .

La phase de prédiction projette vers l'instant k l'état du système à partir des observations des instants 0 à $k - 1$. Nous posons donc la question :

$$P(S^k | z^{0:k-1}). \quad (5.9)$$

La phase d'estimation à l'instant k utilise les observations des instants 0 à k pour estimer l'état du système. Nous posons donc la question :

$$P(S^k | z^{0:k}). \quad (5.10)$$

Les résultats de l'inférence sur ces deux questions permettent de montrer la relation entre les phases de prédiction et d'estimation à des instants successifs, et d'explicitier la boucle que constitue le filtre bayésien.

Pour la phase de prédiction à l'instant k , le résultat de l'inférence donne :

$$\begin{aligned} P(S^k | z^{0:k-1}) &= \int_{S^{0:k-1}} \left(\int_{\substack{Z^{k:N} \\ S^{k+1:N}}} \frac{P(S^{0:N} | Z^{0:N})}{P(z^{0:k-1})} \right) \\ &= \int_{S^{0:k-1}} \left(\int_{\substack{Z^{k:N} \\ S^{k+1:N}}} \frac{P(S^0)P(z^0|S^0) \prod_{i=1}^N P(S^i|S^{i-1})P(Z^i|S^i)}{P(z^{0:k-1})} \right) \\ &= \int_{S^{0:k-1}} \left(\frac{P(S^0)P(Z^0|S^0) \left(\prod_{i=1}^{k-1} P(S^i|S^{i-1})P(Z^i|S^i) \right) P(S^k|S^{k-1})}{P(z^{0:k-1})} \right. \\ &\quad \left. \times \underbrace{\int_{\substack{Z^{k:N} \\ S^{k+1:N}}} P(Z^k|S^k) \prod_{i=k+1}^N P(S^i|S^{i-1})P(Z^i|S^i)}_{=1} \right) \\ &= \int_{S^{0:k-1}} \left(\frac{P(S^0)P(Z^0|S^0) \left(\prod_{i=1}^{k-1} P(S^i|S^{i-1})P(Z^i|S^i) \right) P(S^k|S^{k-1})}{P(z^{0:k-1})} \right) \end{aligned} \quad (5.11)$$

$$= \int_{S^{k-1}} \left(\frac{P(S^k|S^{k-1})}{P(z^{0:k-1})} \int_{S^{0:k-2}} \left(\frac{P(S^0)P(Z^0|S^0) \left(\prod_{i=1}^{k-2} P(S^i|S^{i-1})P(Z^i|S^i) \right)}{\times P(S^{k-1}|S^{k-2})} \right) \right) \quad (5.12)$$

Dans ce résultat, nous notons :

$$E^{k-1} \stackrel{\text{déf}}{=} \frac{P(z^{k-1}|S^{k-1})}{P(z^{0:k-1})} \int_{S^{0:k-2}} \left(\frac{P(S^0)P(Z^0|S^0) \left(\prod_{i=1}^{k-2} P(S^i|S^{i-1})P(Z^i|S^i) \right)}{\times P(S^{k-1}|S^{k-2})} \right) \quad (5.13)$$

Le terme E^{k-1} sera relié au résultat de l'inférence pour la phase d'estimation.

Pour la phase d'estimation à l'instant k , le résultat de l'inférence donne :

$$\begin{aligned}
P(S^k | z^{0:k}) &= \int_{S^{0:k-1}} \left(\int_{\substack{z^{k+1:N} \\ S^{k+1:N}}} \frac{P(S^{0:N} | Z^{0:N})}{P(z^{0:k})} \right) \\
&= \int_{S^{0:k-1}} \left(\int_{\substack{z^{k+1:N} \\ S^{k+1:N}}} \frac{P(S^0)P(z^0|S^0) \prod_{i=1}^N P(S^i|S^{i-1})P(Z^i|S^i)}{P(z^{0:k})} \right) \\
&= \int_{S^{0:k-1}} \left(\frac{P(S^0)P(z^0|S^0) \prod_{i=1}^k P(S^i|S^{i-1})P(z^i|S^i)}{P(z^{0:k})} \times \underbrace{\int_{\substack{z^{k+1:N} \\ S^{k+1:N}}} \prod_{j=k+1}^N P(S^j|S^{j-1})P(Z^j|S^j)}_{=1} \right) \\
&= \int_{S^{0:k-1}} \left(\frac{P(S^0)P(z^0|S^0) \prod_{i=1}^k P(S^i|S^{i-1})P(z^i|S^i)}{P(z^{0:k-1})P(z^k|z^{0:k-1})} \right) \\
&= \frac{P(z^k|S^k)}{P(z^k|z^{0:k-1})} \int_{S^{0:k-1}} \left(\frac{P(S^0)P(z^0|S^0) \left(\prod_{i=1}^{k-1} P(S^i|S^{i-1})P(z^i|S^i) \right) P(S^k|S^{k-1})}{P(z^{0:k-1})} \right) \quad (5.14)
\end{aligned}$$

Dans ce résultat, nous notons I^k l'intégrale :

$$I^k \stackrel{\text{déf}}{=} \int_{S^{0:k-1}} \left(\frac{P(S^0)P(z^0|S^0) \left(\prod_{i=1}^{k-1} P(S^i|S^{i-1})P(z^i|S^i) \right) P(S^k|S^{k-1})}{P(z^{0:k-1})} \right) \quad (5.15)$$

Il nous reste à identifier les termes E^{k-1} apparaissant dans l'expression (5.12) et le terme I^k apparaissant dans l'expression (5.14) pour montrer la structure de boucle du filtre bayésien.

L'identification du terme I^k est immédiate avec l'expression (5.11). I^k est donc égale au résultat de la prédiction à l'instant k . On a donc pour le résultat de l'inférence pour la phase d'estimation :

$$P(S^k | z^{0:k}) = \frac{P(z^k|S^k)}{P(z^k|z^{0:k-1})} P(S^k|z^{0:k-1}) \quad (5.16)$$

$$= \frac{1}{\alpha} P(z^k|S^k) P(S^k|z^{0:k-1}). \quad (5.17)$$

La phase d'estimation consiste donc en une confrontation de la dernière observation capteur avec le résultat de la prédiction.

Quant au terme E^{k-1} , il s'identifie avec l'expression (5.14), l'exposant étant décalé de 1. Le résultat de l'inférence pour la phase de prédiction donne donc au final :

$$P(S^k | z^{0:k-1}) = \int_{S^{0:k-1}} (P(S^k | S^{k-1})P(S^{k-1} | z^{0:k-1})). \quad (5.18)$$

En pratique, le filtrage bayésien revient donc à l'estimation successive des expressions (5.18) et (5.17) pour chaque instant k . La phase de prédiction à l'instant k s'appuie sur le résultat de l'estimation à l'instant $k-1$ et sur le modèle dynamique. La phase d'estimation à l'instant k confronte cette prédiction avec la nouvelle observation capteur, et ainsi de suite.

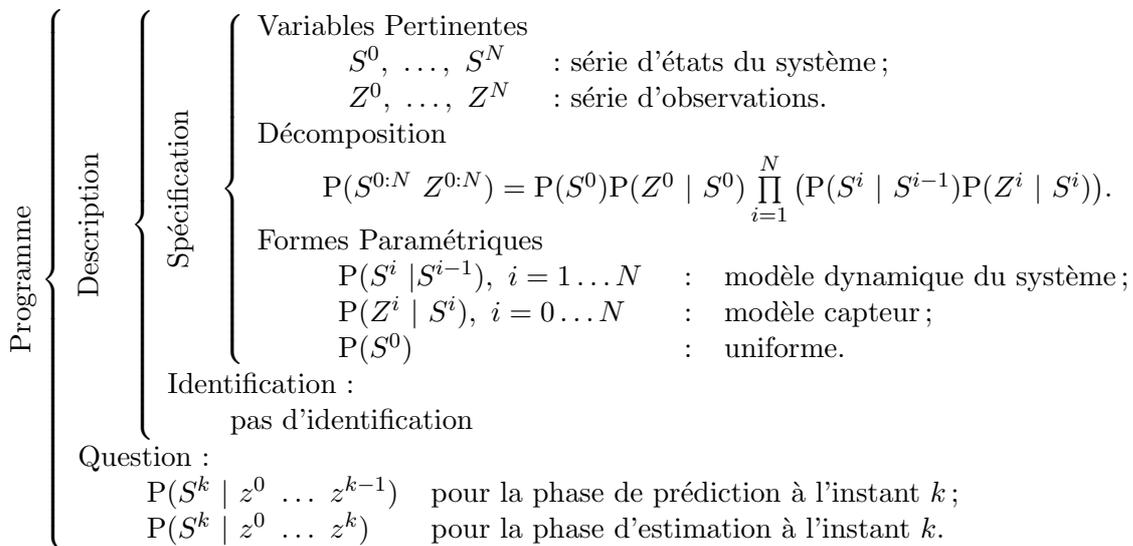


FIG. 5.2: Filtre bayésien : programme.

La figure 5.2 résume la programme bayésien que nous venons de détailler.

5.1.2 Remarque : modèle dynamique incluant des actions du véhicule

Le modèle dynamique défini par l'expression (5.1) ne prend en compte que les mouvements de l'objet poursuivi, entre les instant k et $k-1$. Si le capteur est monté sur une plate-forme elle-même en mouvement, comme c'est le cas pour notre véhicule automobile, le modèle dynamique doit intégrer ces mouvements. Ceci se fait en ajoutant un ensemble de variables supplémentaires, $\{A^0, \dots, A^N\}$, pour modéliser la série d'« actions » du capteur.

On peut discuter du terme « filtre bayésien » pour les formalismes probabilistes intégrant une telle variable d'action. Certains auteurs préfèrent parler de « localisation markovienne » pour ces modèles (voir par exemple [Dia03]). Pour nous, la différence étant faible

par rapport à ce que nous avons jusqu'ici appelé filtre bayésien, nous continuerons à utiliser ce nom.

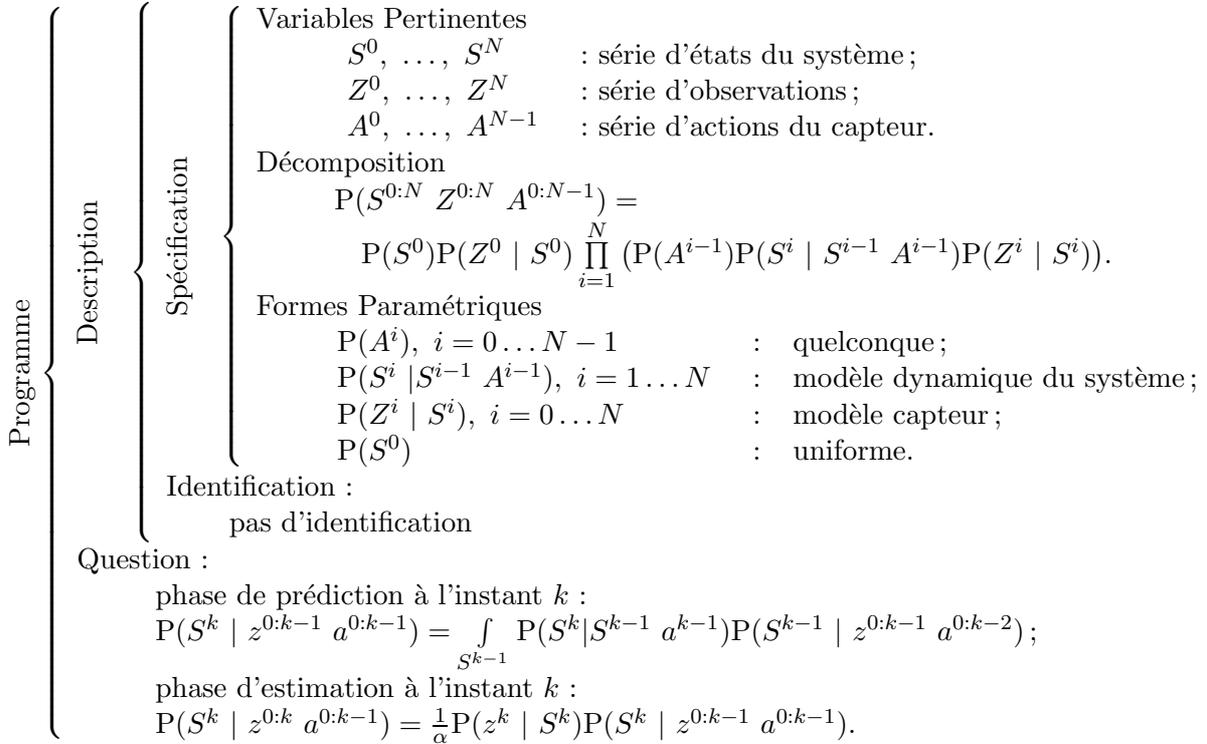


FIG. 5.3: Filtre Bayésien incluant des actions du véhicule portant le capteur : programme bayésien

La figure 5.3 présente le programme bayésien incluant les actions du véhicule portant le capteur. Nous ne présentons pas ce programme en détail, les points importants à souligner étant :

- l'ensemble des variables d'action est ajouté à l'ensemble des variables pertinentes ;
- dans la décomposition, on fait apparaître la distribution de probabilité $P(A^{0:k-1})$. La forme paramétrique associée à cette distribution a peu d'importance, l'ensemble des variables d'action apparaissant en partie droite des questions ;
- le modèle dynamique dans la phase de prédiction prend en compte l'action du capteur à l'instant $k - 1$ et devient donc $P(S^k | S^{k-1} A^{k-1})$;
- les questions deviennent respectivement $P(S^k | z^{0:k-1} a^{0:k-1})$ pour la phase de prédiction et $P(S^k | z^{0:k} a^{0:k-1})$ pour la phase d'estimation.

En étudiant les résultats de l'inférence, on comprend bien que l'ajout de la variable d'action change peu de chose. En effet, les distributions de probabilité $P(A^{0:k-1})$ n'interviennent ni dans la prédiction, ni dans l'estimation. La seule différence significative est la prise en compte de la variable A^{k-1} dans le modèle dynamique au cours de la phase de prédiction.

On ne peut présenter le filtre bayésien sans parler des implantations classiques qui ont rendu son utilisation possible. C'est le but de la sous-section suivante. De plus, ces implantations sont souvent utilisées comme méthode de filtrage dans les méthodes classiques de poursuite multipiste que nous présenterons dans le chapitre suivant. Afin de ne pas surcharger, nous choisissons de présenter ces implantations sans mentionner la variable d'action. Il faut garder à l'esprit que l'ajout de cette variable ne change que le modèle dynamique, et n'est jamais une limite aux implantations.

5.1.3 Les implantations classiques du filtre bayésien

La principale difficulté pour l'implantation du filtre bayésien tel que nous venons de le présenter réside dans le calcul de l'intégrale sur l'ensemble des états possibles du système lors de la phase de prédiction (voir expression (5.18)).

Dans certains cas très particuliers, il est possible de déterminer une solution analytique à cette expression. C'est le cas par exemple du filtre de Kalman[Kal60][WB].

Mais dans la plupart des cas, on ne peut déterminer de solution analytique pour $P(S_k | Z_{0:k-1})$. Il est alors nécessaire d'approcher la solution. Nous présentons dans ce cas le filtre de Kalman étendu[WB] et le filtre à particules [AMGC02].

5.1.3.1 Le filtre de Kalman

Le filtre de Kalman [Kal60] présente l'avantage d'être un algorithme exact. Les hypothèses nécessaires à son application sont par contre très restrictives. Une bonne introduction intuitive est donnée par Welch et Bishop [WB]. La preuve des résultats analytiques qui vont suivre est présentée dans [BBW95].

Le filtre de Kalman travaille avec des variables d'état S^k et d'observations Z^k à supports continus.

La principale hypothèse du filtre de Kalman est de dire qu'à chaque instant k , la distribution *a posteriori* $P(S^k | Z^{0:k})$ est une Gaussienne. On peut donc la représenter par sa moyenne s^k et sa matrice de covariance P^k .

En effet, si $P(S^{k-1} | Z^{0:k-1})$ est une distribution Gaussienne, il est possible de prouver que $P(S^k | Z^{0:k})$ est également Gaussienne, à condition de supposer :

- le modèle dynamique est une fonction linéaire de S^{k-1} et de w^k . De plus, le bruit w^k est supposé être un bruit Gaussien de moyenne nulle. En notant Q^k sa matrice de covariance, l'expression (5.1) peut être ré-écrite sous forme matricielle :

$$s^k = F^k s^{k-1} + Q^k ;$$

- le modèle capteur est une fonction linéaire de S^k et de v^k . De plus, le bruit v^k est supposé être un bruit Gaussien de moyenne nulle. En notant R^k sa matrice de covariance, l'expression (5.2) peut être ré-écrite sous forme matricielle :

$$z^k = H^k s^k + R^k ;$$

– la forme paramétrique associée à $P(S^0)$ est une Gaussienne.

On note souvent s^{k-} , P^{k-} la moyenne et la matrice de covariance de la distribution *a priori*, et \hat{s}^k , P^k celles de la distribution *a posteriori*.

La phase de prédiction met à jour la moyenne et la matrice de covariance de la distribution *a priori*, en suivant les équations :

$$s^{k-} = F^{k-1} \hat{s}^{k-1} \quad (5.19)$$

$$P^{k-} = F^{k-1} P^{k-1} {}^t F^{k-1} + Q^{k-1}. \quad (5.20)$$

La phase d'estimation met à jour la moyenne et la matrice de covariance de la distribution *a posteriori*. Pour cela on suit les équations :

$$K^k = P^{k-} {}^t H^k (H^k P^{k-} {}^t H^k + R^k)^{-1} \quad (5.21)$$

$$\hat{s}^k = \hat{s}^{k-} + K^k (z^k - H^k \hat{s}^{k-}) \quad (5.22)$$

$$P^k = (I - K^k H^k) P^{k-}. \quad (5.23)$$

K^k est souvent nommé le gain du filtre. Le terme $(z^k - H^k \hat{s}^{k-})$ apparaissant dans l'expression (5.22) est nommé « l'innovation » du filtre.

Grâce à ces résultats analytiques, le filtre de Kalman est particulièrement efficace. Malheureusement les hypothèses ne peuvent toujours être satisfaites, particulièrement les conditions de linéarité du modèle dynamique et du modèle capteur. Une solution est alors d'utiliser le filtre de Kalman étendu.

5.1.3.2 Le filtre de Kalman étendu

Lorsque le modèle de transition et le modèle capteur ne sont pas linéaires, il est souvent suffisant d'utiliser un développement de Taylor au premier ordre pour les considérer localement linéaires. Dans les équations (5.20), (5.21) et (5.23), les matrices F^k et H^k sont remplacées par les Jacobiennes des modèles de transition et capteurs. Pour les équations (5.19) et (5.22), on peut utiliser le modèle non linéaire.

Ainsi, en notant :

– A^k la Jacobienne des dérivées partielles de f par rapport à s^k , c'est à dire :

$$A_{[i,j]}^k = \frac{\partial f_{[i]}}{\partial s_{[j]}}(s^k, 0)$$

– W la Jacobienne des dérivées partielles de f par rapport au bruit w^k , c'est à dire :

$$W_{[i,j]}^k = \frac{\partial f_{[i]}}{\partial w_{[j]}}(s^k, 0)$$

– H^k la Jacobienne des dérivées partielles de h par rapport à s^k , c'est à dire :

$$H_{[i,j]}^k = \frac{\partial h_{[i]}}{\partial s_{[j]}}(s^k, 0)$$

– V la Jacobienne des dérivées partielles de h par rapport au bruit v^k , c'est à dire :

$$W_{[i,j]}^k = \frac{\partial h_{[i]}}{\partial v_{[j]}}(s^k, 0)$$

les équations de la phase deviennent :

$$s^{k-} = f(\hat{s}^{k-1}, 0) \quad (5.24)$$

$$P^{k-} = A^{k-1} P^{k-1} {}^t A^{k-1} + W^{k-1} Q^{k-1} {}^t W^{k-1}, \quad (5.25)$$

et celles de la phase d'estimation :

$$K^k = P^{k-} {}^t H^k (H^k P^{k-} {}^t H^k + V^k R^k {}^t V^k)^{-1} \quad (5.26)$$

$$\hat{s}^k = \hat{s}^{k-} + K^k (z^k - H^k \hat{s}^{k-}) \quad (5.27)$$

$$P^k = (I - K^k H^k) P^{k-}. \quad (5.28)$$

Récemment, la transformation « non parfumée »³ a été utilisée dans le cadre d'un filtre de Kalman étendu [JUDW95, JUDW00]. L'idée consiste à remplacer la phase de prédiction de filtre de Kalman étendu par les trois étapes suivantes :

1. un ensemble de points est choisi, non pas aléatoirement mais de manière déterministe, selon la distribution *a posteriori* $P(S^k | Z^{0:k})$, qui rappelons le est une distribution Gaussienne ;
2. on applique à chacun de ces points le vrai modèle dynamique du système, non-linéaire ;
3. les paramètres de la distribution *a priori*, soit la moyenne et l'écart-type, sont estimés grâce à cet ensemble de points.

Dans certains cas, il a été prouvé que les résultats de ce genre de filtre sont meilleurs que ceux du filtre de Kalman étendu, la non-linéarité du modèle dynamique étant mieux approchée.

Le principal inconvénient du filtre de Kalman étendu (et du « filtre non parfumé ») reste l'hypothèse Gaussienne sur la distribution *a priori* et la distribution *a posteriori*. Comme il est souligné dans [SBFC01], une telle distribution ne pourra jamais bien approximer une distribution multi-modale par exemple.

5.1.3.3 Le filtre à particules

Le filtre à particules connaît un vif succès depuis quelques années, y compris dans le domaine de la poursuite de cible [AMGC02]. Dans la littérature, il est présenté sous divers noms : « bootstrap filtering », « condensation algorithm », « particle filtering », « interacting partical approximations », et « survival of the best ». Il s'agit d'une implantation du filtre bayésien utilisant des méthodes de tirage de Monte-Carlo [DGA00].

³traduction mot-à-mot de *unscented transform*

« **Sequential Importance Sampling** » Cet algorithme est la forme la plus générale du filtre à particules. L'idée principale est de représenter la distribution *a posteriori* par un ensemble de N particules. A chacune de ces particules est associé un poids, dont la valeur est proportionnelle à la probabilité.

Soit $\{(s_i^k, w_i^k), i = 1 \dots N\}$ un ensemble aléatoire caractérisant la distribution *a posteriori* $P(S^k | Z^{0:k})$. $\{s_i^k, i = 1 \dots N\}$ est un ensemble de points auxquels sont associés les poids $\{w_i^k, i = 1 \dots N\}$. Les poids sont choisis de manière à être normalisés. On a alors une approximation de la distribution *a posteriori*⁴ :

$$P(S^k | Z^{0:k}) \approx \sum_{i=1}^N w_i^k \delta(S^k - s_i^k) \quad (5.29)$$

Quand le nombre d'échantillons N devient très grand, cette représentation tend vers la distribution *a posteriori* exacte.

Les poids sont choisis suivant la méthode de l'« Importance sampling », qui est une méthode de tirage indirecte de Monte Carlo. Pour illustrer cette méthode, prenons le cas d'une distribution de probabilité $P(X)$ de dimension 1. On suppose être capable d'évaluer cette distribution pour toute valeur de la variable X . Supposons aussi qu'on ne dispose pas de méthode directe pour effectuer des tirages à partir de cette distribution.

Supposons maintenant que l'on connaisse une autre distribution de probabilité $Q(X)$, « proche » de $P(X)$, que l'on sait évaluer pour toutes les valeurs de X , et à partir de laquelle on est capable de générer directement des tirages. Cette distribution est appelée *la distribution de tirage*⁵.

On génère alors N tirages $\{x_{(i)}\}_{i=1}^N$ à partir de $Q(x)$. La notion de poids est introduite pour prendre en compte le fait que ces tirages ont été générés à partir de $Q(X)$ et non à partir de $P(X)$:

$$w_i = \frac{P([X = x_i])}{Q([X = x_i])}. \quad (5.30)$$

On a alors une approximation de la distribution $P(X)$:

$$P([X = x]) \approx \sum_{i=1}^N w_i \delta(x - x_i). \quad (5.31)$$

Dans le cas du filtre bayésien, on souhaite pouvoir utiliser la propriété de récursivité du filtre. On dispose d'un ensemble de particules représentant la distribution $P(S^{k-1} | Z^{0:k-1})$, et on veut un nouvel ensemble de particules représentant la distribution $P(S^k | Z^{1:k})$.

⁴Dans cette équation, le symbole δ désigne la *fonction delta* [HS98], ou encore *masse de Dirac*, qui peut être caractérisée par l'intégrale :

$$\int_x^y \delta(t - a) dt = \begin{cases} 1 & \text{si } x < a < y \\ 0 & \text{si } a < x \text{ ou } y < a \end{cases}$$

La fonction $f(x) = \delta(x - a)$ est nulle partout, sauf en a où elle est infinie.

⁵*Importance distribution* en anglais

En choisissant la distribution de tirage de la forme $Q(S^k | S^{k-1} Z^{0:k})$, on peut avoir une approximation de la nouvelle distribution *a posteriori* :

$$P(S^k | Z^{0:k}) = \sum_{i=1}^N w_i^k \delta(S^k - s_i^k), \quad (5.32)$$

les poids w_i^k étant calculés de manière récursive par :

$$w_i^k \propto w_i^{k-1} \frac{P(z^k | s_i^k) P(s_i^k | s_i^{k-1})}{Q(s_i^k | s_i^{k-1} z^k)}. \quad (5.33)$$

L'algorithme du « sequential importance sampling » est alors le suivant :

- Pour i de 1 à N
 - tirage de s_i^k suivant $Q(S^k | s_i^{k-1} z^k)$;
 - le poids w_i^k est associé à cette particule d'après (5.33).
- Fin Pour

Un des gros problèmes dans la mise en place d'un filtre à particules réside dans le choix du nombre de particules pour représenter la distribution *a posteriori*. En effet, un nombre trop important de particules peut entraîner un temps de calcul trop grand, et au contraire, un nombre trop faible de particules peut entraîner une approximation trop grossière de la distribution *a posteriori*. Peu de travaux semblent avoir été menés sur ce problème. On peut néanmoins citer [Fox01], qui dans un contexte de localisation, détermine le nombre de particules suivant la cohérence entre les deux distributions $P(S^k | z^k)$ et $P(S^k | z^{0:k})$.

5.2 Grille d'occupation et filtres bayésien

Nous présentons maintenant l'adaptation du filtre bayésien aux grilles d'occupation, adaptation qui définit le *filtre d'occupation bayésien*. Le but de cette adaptation est d'intégrer à l'estimation de la grille d'occupation les propriétés temporelles du filtre bayésien.

Dans cette section, nous nous plaçons dans le cas monocapteur, puisque nous n'avons pour l'instant présenté le filtre bayésien que pour ce cas.

Nous commençons par présenter le programme bayésien. Des résultats issus de simulation illustrent ensuite l'utilisation de ce programme.

5.2.1 Programme bayésien

Nous avons pour l'instant présenté le filtre bayésien sous une forme très générale. L'adaptation de ce filtre à un cas particulier nécessite principalement de préciser la forme des variables d'état S^k du système et des variables d'observation Z^k .

Pour l'adaptation du filtre bayésien aux grilles d'occupation, la variable d'état S^k est formée de la conjonction de deux variables : X^k et E_X^k , qui représentent une cellule et son état d'occupation à l'instant k .

5.2.1.1 Variables

Puisque la variable d'état du filtre bayésien est constituée d'une part de la cellule sur laquelle nous travaillons et d'autre part de l'état de cette cellule, nous travaillons avec les variables :

- X^0, \dots, X^N , qui précise sur quelle cellule nous travaillons ;
- E_X^0, \dots, E_X^N qui représentent l'état de la cellule X à chaque instant compris entre 0 et N ;
- Z^0, \dots, Z^N qui représentent la série des observations depuis l'instant initial.

Ces variables définissent la distribution de probabilité conjointe :

$$P(X^{0:N} E_X^{0:N} Z^{0:N}). \quad (5.34)$$

5.2.1.2 Décomposition

La décomposition de la distribution de probabilité conjointe donne (voir figure 5.2) :

$$P(X^{0:N} E_X^{0:N} Z^{0:N}) = P(E_X^0 X^0) P(Z^0 | E_X^0 X^0) \prod_{k=1}^N P(E_X^k X^k | E_X^{k-1} X^{k-1}) P(Z^k | E_X^k X^k). \quad (5.35)$$

5.2.1.3 Formes paramétriques

Nous associons les formes paramétriques suivantes aux distributions de probabilité élémentaires apparaissant dans la décomposition (5.35) :

- $P(X^0 E_X^0)$: distribution uniforme si nous ne connaissons rien sur l'état de la cellule à l'instant initial ;
- $P(Z^k | E_X^k X^k)$, $k = 0 \dots N$ est donnée par le modèle capteur que nous avons décrit dans la section 3.2.2 ;
- $P(E_X^k X^k | E_X^{k-1} X^{k-1})$ correspond au modèle dynamique de notre système. Puisque la notion de piste n'est pas présente dans le filtre d'occupation, le modèle dynamique est ici associé aux cellules. Connaissant l'état d'occupation des cellules de la grille à l'instant $k - 1$, il est en effet possible de prédire l'état d'occupation de la cellule X à l'instant k .

Notre décomposition est maintenant terminée, nous pouvons l'utiliser au travers de questions.

5.2.1.4 Questions

À chaque instant k , nous posons la succession de questions :

$$P(E_X^k X^k | z^{0:k-1}) \quad (5.36)$$

pour la phase de prédiction, et :

$$P(E_X^k X^k | z^{0:k}) \quad (5.37)$$

pour la phase d'estimation.

En reprenant les résultats de l'inférence du filtre bayésien (5.17) et (5.18), nous pouvons donner les résultats de l'inférence dans ce cas particulier. Pour la phase de prédiction :

$$P(E_X^k X^k | z^{0:k-1}) = \int_{\substack{X^{k-1} \\ E_X^{k-1}}} \left(P(E_X^{k-1} X^{k-1} | z^{0:k-1}) P(E_X^k X^k | E_X^{k-1} X^{k-1}) \right); \quad (5.38)$$

et pour la phase d'estimation :

$$P(E_X^k X^k | z^{0:k}) = \frac{1}{\alpha} P(z^k | E_X^k X^k) P(E_X^k X^k | z^{0:k-1}). \quad (5.39)$$

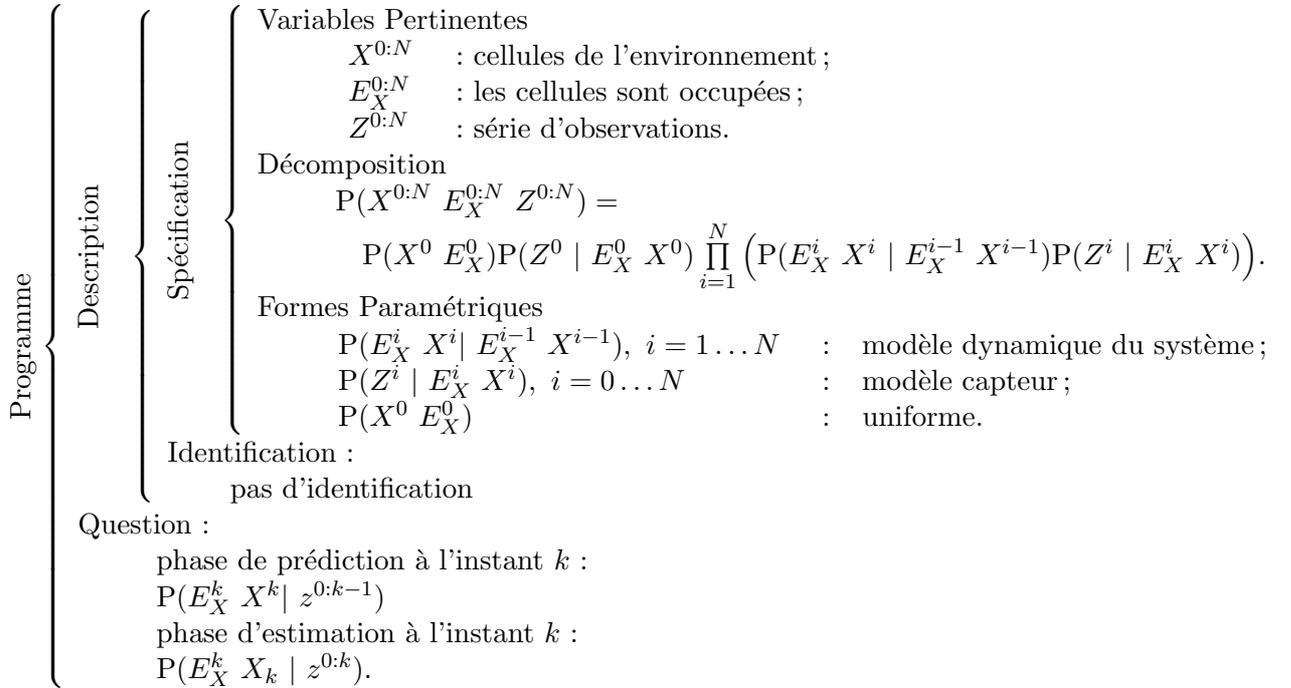


FIG. 5.4: Le filtre d'occupation bayésien monocapteur : programme bayésien

Ces deux questions permettent de conserver la structure de boucle prédiction/estimation du filtre bayésien. Elles ne correspondent toutefois pas exactement à ce que nous voudrions estimer. En effet, nous voudrions estimer la probabilité d'occupation d'une cellule connaissant la série d'observation soit

$$P(E_x^k | z^{0:k} x^k).$$

Poser cette question, et l'équivalente pour la phase de prédiction, brise la récursivité du filtre bayésien. Par contre, les valeurs $P([E_x^k = e] | z^{0:k} x^k)$ et $P([E_x^k = e] x^k | z^{0:k})$ ne sont liées que par une constante de normalisation, égale à $P(x^k | z^{0:k})$. La même relation est vraie pour la phase de prédiction.

Nous poserons donc la série de questions (5.36) et (5.37) afin de conserver la structure récursive du filtre bayésien. La normalisation étant effectuée en dernier lieu pour estimer l'état d'une cellule.

La figure 5.4 résume le programme bayésien que nous venons de détailler.

Nous rappelons que les questions (5.36) et (5.37) permettent d'estimer l'état d'une seule cellule de la grille. Autant de question que de cellule dans la grille seront donc posées à chaque instant k .

La prédiction de l'état d'une cellule à l'instant k nécessite une intégrale sur l'ensemble des cellules de la grille. Cette intégrale ne pourra être calculée de manière exacte dans un temps raisonnable si la taille de la grille devient trop grande. Il est donc nécessaire d'en calculer une approximation.

La sous-section suivante présente l'algorithme d'approximation que nous avons développé pour ce cas particulier.

5.2.2 Approximation de l'intégrale lors de la prédiction

La phase de prédiction nécessite le calcul d'une intégrale portant sur toutes les cellules X de notre grille. De plus, il ne s'agit pas de calculer une seule intégrale, mais une par cellule, puisque le résultat de l'équation (5.38) donne la probabilité d'occupation *a priori* d'une seule cellule de l'environnement.

Pour l'application que nous allons présenter dans le chapitre 7, la grille à évaluer comporte environ 30000 cellules. Il est donc impensable de calculer de manière exacte l'intégrale pour chaque cellule de l'environnement. On va donc chercher à calculer cette intégrale de manière approchée, en ne considérant qu'une partie des cellules à l'instant $k - 1$. En cela, c'est une intégration de Monte-Carlo.

Dans un premier temps, nous pouvons ré-écrire le résultat (5.38) de l'inférence pour la phase de prédiction sous la forme :

$$P(E_X^k | X^k | z^{0:k-1}) = \int_{\substack{X^{k-1} \\ E_X^{k-1}}} \left(P(E_X^{k-1} | X^{k-1} | z^{0:k-1}) P(X^k | X^{k-1}) P(E_X^k | E_X^{k-1} | X^{k-1} | X^k) \right)$$

$P(X^k | X^{k-1})$ représente le modèle dynamique du système décrit par l'expression (5.1).

Les distributions $P(E_X^k | E_X^{k-1} | X^{k-1} | X^k)$ sont quant à elle représentées par des Dirac ; si la cellule X^{k-1} est occupée à l'instant $k - 1$ et que l'objet se déplace jusqu'à la cellule X^k , celle-ci est occupée à l'instant k . Ceci se traduit :

- $P([E_x^k = 1] | [E_x^{k-1} = 1] | X^{k-1} | x^k) = 1$;
- $P([E_x^k = 0] | [E_x^{k-1} = 1] | X^{k-1} | x^k) = 0$;
- $P([E_x^k = 1] | [E_x^{k-1} = 0] | X^{k-1} | x^k) = 0$;
- $P([E_x^k = 0] | [E_x^{k-1} = 0] | X^{k-1} | x^k) = 1$.

Grâce à ces hypothèses, le résultat de l'inférence devient :

$$P(E_X^k | X^k | z^{0:k-1}) = \int_{X^{k-1}} P(E_X^{k-1} | X^{k-1} | z^{0:k-1}) P(X^k | X^{k-1}). \quad (5.40)$$

Dans cette expression $P(E_X^{k-1} X^{k-1} | z^{0:k-1})$ est le résultat de la phase d'estimation de la grille d'occupation à l'instant $k - 1$, et $P(X^k | X^{k-1})$ est le modèle dynamique associé aux cellules. Les formes paramétriques de ces deux distributions sont donc connues.

Notre but est donc de calculer l'intégrale pour chaque cellule x^k de l'environnement. Afin d'approximer au mieux l'intégrale, il convient de choisir judicieusement pour chaque cellule x^k l'ensemble des cellules x^{k-1} qui participeront au calcul de l'intégrale.

Notre idée consiste alors à inverser le problème : plutôt que de définir pour chaque x^k l'ensemble des cellules x^{k-1} qui participeront au calcul de l'intégrale, nous choisissons pour chaque cellule x^{k-1} l'ensemble des cellules x^k pour lesquelles l'état de x^{k-1} sera utilisé pour approximer l'intégrale. Le choix de ces cellules x^k se fait en utilisant le modèle dynamique $P(X^k | X^{k-1})$.

Cette idée est assez proche du filtre à particules : on fait « avancer chaque cellule » de l'instant $k - 1$ vers son nouvel état à l'instant k .

Première phase :

- Pour chaque cellule de l'environnement x^{k-1} :
 - estimer la distribution $P(X^k | X^{k-1})$;
 - tirer un point x^k selon cette distribution ;
 - ajouter x^{k-1} à L_{x^k} ;
 - $N_{x^k} += 1$;

Deuxième phase :

- pour chaque cellule x^k de l'environnement :
 - $\int_{X^{k-1}} (\dots) \approx \frac{1}{N_{x^k}} \sum_{x_i^{k-1} \in L_{x^k}} P(E_{x_i}^{k-1} x_i^{k-1} | z^{0:k-1}) P(x^k | x_i^{k-1})$.

Fig. 5.5: Algorithme d'approximation de l'intégrale dans le filtre d'occupation bayésien.

La figure 5.5 résume notre algorithme pour le cas où l'on choisit une seule cellule X^k pour chaque cellule X^{k-1} . L_{x^k} désigne la liste des cellules qui seront utilisées pour estimer la probabilité d'occupation prédite de la cellule x^k , N_{x^k} désigne le nombre d'éléments de cette liste après la première phase de l'algorithme. Le « tirage » de la cellule x^k selon la distribution $P(X^k | X^{k-1})$ peut se résumer à prendre le point de probabilité maximale.

Le principal avantage de cet algorithme est que la complexité de l'estimation de l'intégrale devient linéaire en fonction du nombre de cellules de l'environnement, alors que la complexité du calcul exact de cette intégrale est fonction du carré du nombre de ces cellules. Cette méthode permet également de propager toutes les connaissances de la grille d'occupation à l'instant $k - 1$ vers l'instant k . On ne transmet pas uniquement les connaissances sur les cellules occupées, mais également les connaissances sur les cellules masquées et les cellules inoccupées.

Le principal inconvénient de cet algorithme tient au fait que nous ne sommes pas capables d'assurer un nombre minimum de points pour estimer l'état de la cellule x^k , c'est à dire le nombre N_{x^k} à la fin de la première phase. Ce nombre de points est même inconnu. Nous pouvons juste dire qu'en moyenne, un point est utilisé pour l'estimation de l'intégrale. Il peut arriver qu'aucune cellule x^{k-1} ne soit retenue comme participant à la prédiction de

l'état d'une cellule x^k . Dans ce cas ($N_{x^k} = 0$), le résultat de la prédiction de la probabilité d'occupation pour cette cellule sera égale à 0,5. Nous ne pourrions donc tirer de conclusion sur l'occupation de cette cellule à la fin de la phase de prédiction.

Finalement, nous verrons dans les illustrations à venir sur des cas de scènes dynamiques que cette méthode d'intégration donne des résultats satisfaisants, malgré le peu de points utilisés pour estimer chaque intégrale. Ceci vient du fait que les points sont choisis comme étant les plus pertinents.

5.2.3 Illustration

Nous présentons ici des résultats d'estimation de grille d'occupation incluant une phase de prédiction. Le but de cette illustration est de montrer le rôle de la phase de prédiction pour la robustesse de l'estimation.

Nous nous plaçons pour l'instant dans la situation décrite dans la sous-section 3.2.3.2 : la grille d'occupation que nous cherchons à estimer est 2-D, une cellule étant représentée dans le repère cartésien, les observations du capteur étant représentées dans le repère polaire.

5.2.3.1 Véhicule immobile

Le premier exemple que nous présentons est volontairement très simple. Dans cet exemple, le véhicule qui porte le capteur est immobile. Un objet est présent dans le champ de vue du capteur. Cet objet est également immobile.

La figure 5.6 présente le résultat des phases de prédiction (colonne de gauche) et d'estimation (colonne de droite) des grilles d'occupation pour trois instants successifs, notés $k = 1$, $k = 2$, et $k = 3$. La couleur d'une cellule représente la probabilité que cette cellule soit occupée. La colonne centrale présente la position de l'objet par rapport au capteur, pour chacun des trois instants. Dans cette expérience, l'objet est systématiquement détecté par le capteur.

À l'instant initial, c'est à dire $k = 1$, aucune mesure capteur n'étant encore disponible, la phase de prédiction conduit à l'uniforme sur la probabilité d'occupation, quelle que soit la cellule (figure 5.6(a)). Pour la phase d'estimation, la première mesure capteur est disponible. Comme la connaissance *a priori* est uniforme, le résultat de l'estimation (figure 5.6(c)) est semblable à ceux obtenus dans la sous-section 3.2.3.2.

La figure 5.6(d) présente le résultat de la prédiction pour $k = 2$. Cette prédiction est basée sur le résultat de l'estimation à l'instant $k = 1$ et sur le modèle dynamique. On voit que le résultat de cette phase de prédiction est très proche du résultat précédent de l'estimation. L'évolution entre ces deux phases dépend notamment du bruit associé au modèle dynamique du système.

La figure 5.6(f) présente le résultat de l'estimation pour $k = 2$, qui se base sur une deuxième observation capteur. Pour cette estimation, on utilise comme *a priori* le résultat de la phase de prédiction précédente, et non plus l'uniforme. Le résultat de cette phase d'estimation est bien entendu très similaire à celui de la première estimation, mais on remarque tout de même que la probabilité d'occupation pour les cellules correspondant à

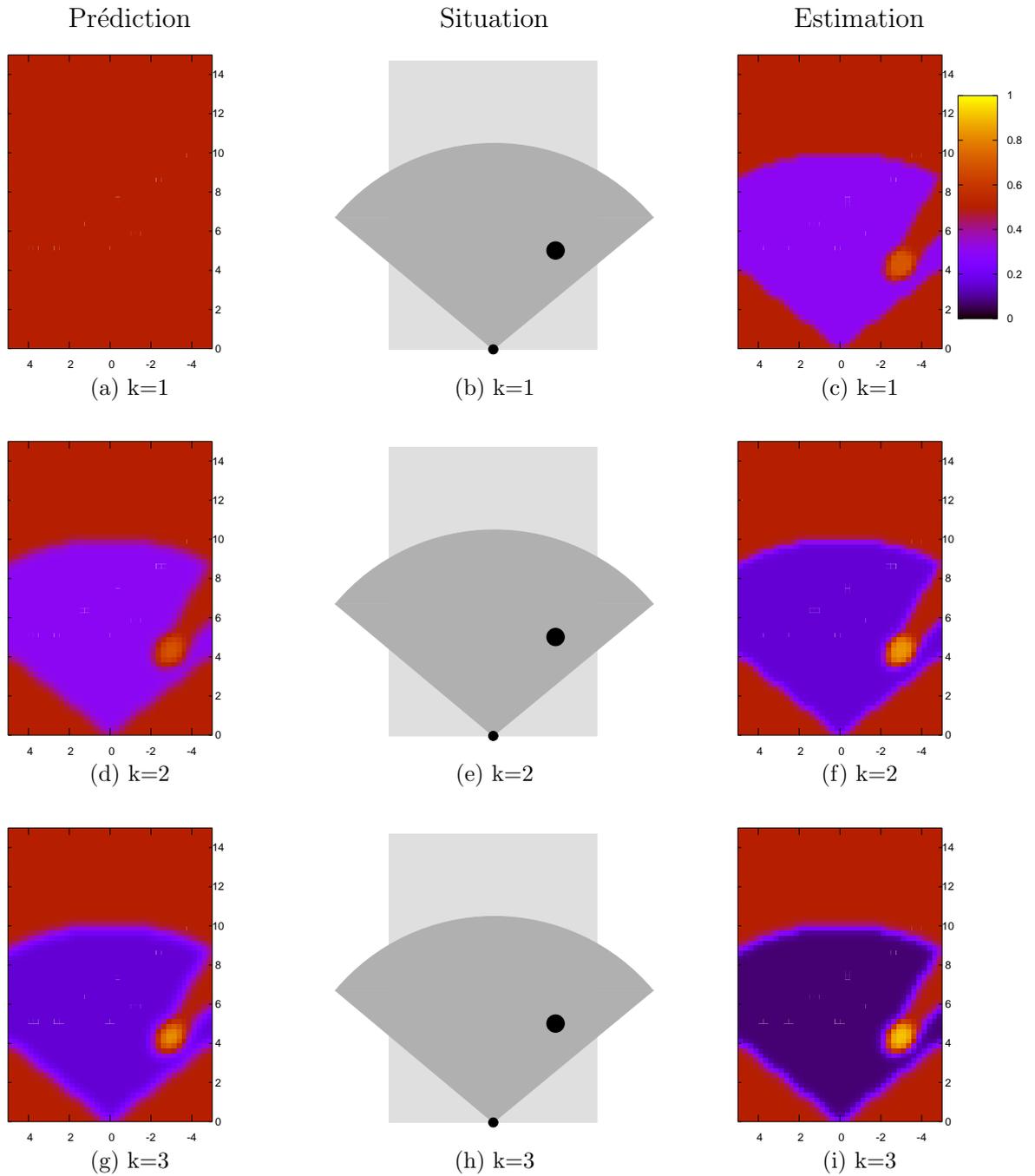


FIG. 5.6: Premier exemple d'estimation de grille d'occupation incluant une phase de prédiction. La colonne de gauche présente la probabilité d'occupation prédite pour chaque cellule de la grille. La colonne centrale présente une vue d'oiseau de la situation pour les différents instants. Enfin la colonne de droite présente le résultat de la phase d'estimation de la probabilité d'occupation de chaque cellule de la grille.

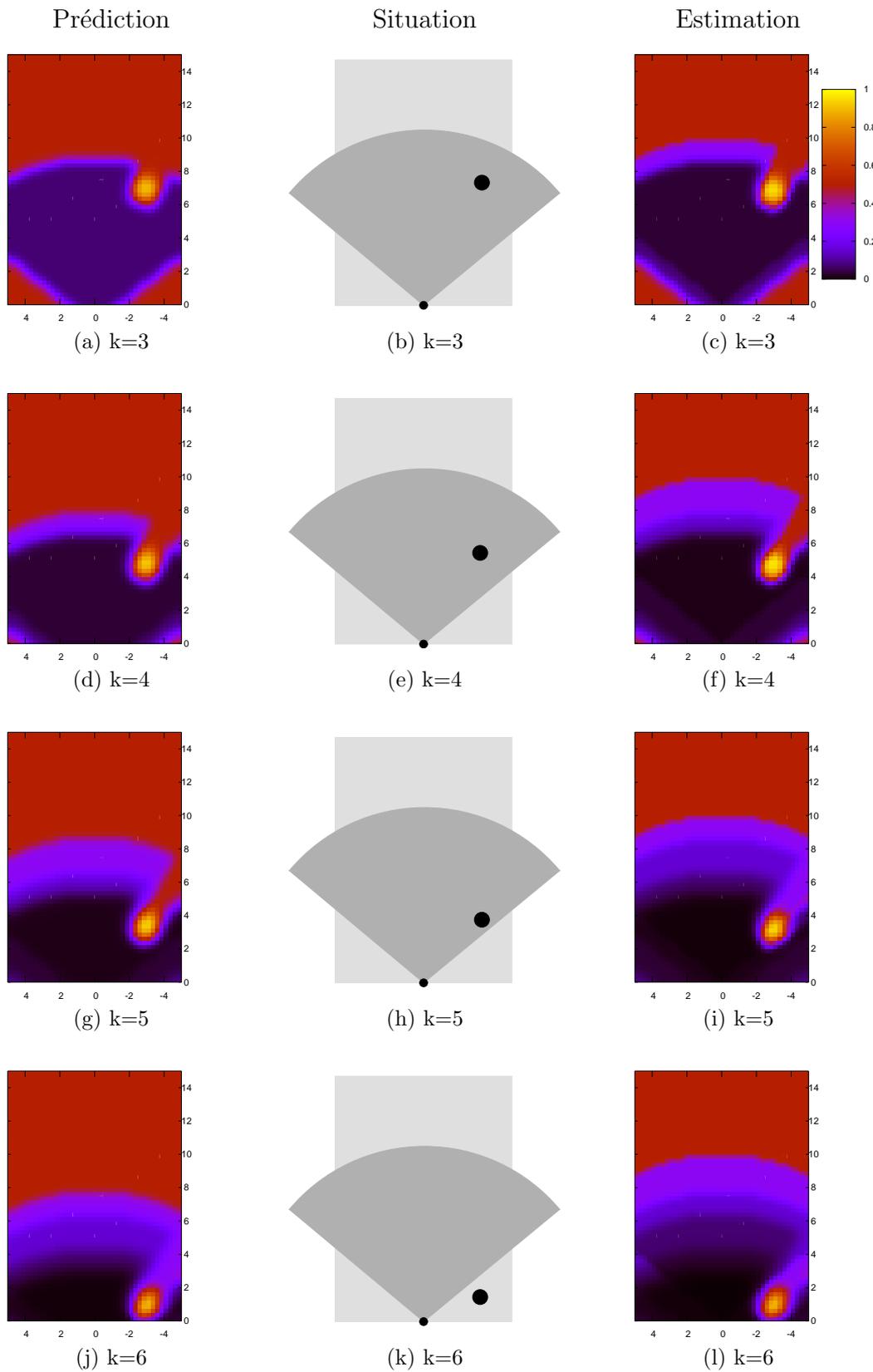


FIG. 5.7: Autre exemple d'estimation, le cycab avançant vers une cible immobile.

la mesure capteur est plus élevée que pour la première estimation. Ceci est dû à la phase de prédiction, puisque la probabilité *a priori* d'occupation de ces cellules était supérieure à 0,5. De la même manière, loin de la mesure capteur, la probabilité d'occupation est plus faible que pour la première estimation, puisque l'*a priori* est cette fois inférieur à 0,5.

Les figures 5.6(g) et 5.6(i) montrent le résultat de la phase de prédiction et de la phase d'estimation à la troisième itération. Les mêmes remarques que pour $k = 2$ sont valables, la probabilité d'occupation dans les cellules proches des mesures capteurs augmentant encore, et celle des cellules éloignées de la mesure capteurs faiblissant encore.

5.2.3.2 Véhicule en mouvement

Dans le deuxième exemple présenté par la figure 5.7, le véhicule portant le capteur est cette fois mobile. Il avance en direction d'une cible immobile. Comme précédemment, la colonne de gauche présente le résultat de la phase de prédiction, la colonne centrale la situation vue d'oiseau et la colonne de droite le résultat de l'estimation.

Pour les trois lignes, correspondant aux instants $k = 3$ à $k = 5$, les résultats sont très similaires à l'exemple précédent, puisque l'objet est systématiquement détecté par le capteur. On remarque tout de même que le mouvement du véhicule portant le capteur a une incidence sur le résultat de la phase de prédiction. Par exemple sur la grille présentée figure 5.7(d), le résultat de la phase de prédiction pour les cellules situées à plus de huit mètres du capteur donne une probabilité d'occupation égale à 0,5, donc égale à l'uniforme. Cette phase de prédiction est estimée à partir du résultat de la phase d'estimation précédente, présentée figure 5.7(c). Puisque la portée du capteur est de dix mètres, on ne peut rien dire au delà de cette distance. Puisque la phase de prédiction à l'instant suivant ne tient pas compte de nouvelles observations capteur, si le véhicule a avancé de deux mètres entre les instants $k = 3$ et $k = 4$, la phase de prédiction ne peut rien dire pour une distance supérieure à huit mètres. Si le véhicule décélère, la portée de la phase de prédiction augmente. C'est pourquoi à l'instant $k = 5$ la portée de la phase de prédiction est légèrement supérieure à huit mètres (figure 5.7(g)).

Si la « portée » de la phase de prédiction est réduite par rapport à la phase d'estimation quand le véhicule avance, cette phase de prédiction va par contre permettre de donner des informations sur des zones situées hors du champ de vue du capteur, mais plus près du véhicule.

Tout d'abord, on remarque à $k = 5$ (figure 5.7(i)) que la probabilité de la zone située derrière l'objet n'est pas égale à 0,5. Ceci est dû au fait que cette zone était observable à $k = 3$ et $k = 4$, et qu'aucun objet ne l'occupait à ces instants.

Le cas le plus intéressant est illustré par les figures 5.7(j), 5.7(k) et 5.7(l). On voit sur la figure 5.7(k) que l'objet est situé en dehors du champ de vue du capteur. En conséquence, cet objet n'est pas détecté. Pourtant, on remarque sur la figure 5.7(l) une zone de forte probabilité d'occupation, correspondant à cet objet. Ceci est possible grâce à la phase de prédiction, présentée figure 5.7(j). En effet, cette phase de prédiction tient compte du résultat de l'estimation à l'instant précédent (figure 5.7(i)) et du mouvement vers l'avant du véhicule portant le capteur. Une zone de forte probabilité d'occupation est donc prédite,

sans tenir compte du modèle du capteur.

La phase de prédiction permet donc d'enrichir la connaissance par rapport à l'utilisation des observations capteur seules. Elle permet également une robustesse aux défaillances du capteur. En effet, si pour une raison quelconque le capteur ne voit pas une cible, du fait de la prédiction la probabilité d'occupation de la zone correspondante ne tombera pas immédiatement sous 0,5. Ceci sera notamment illustré dans le cadre des occultations entre objets dans le chapitre suivant.

5.3 Fusion de capteurs

Pour l'instant, nous avons décrit le filtre bayésien et son adaptation aux grilles d'occupation lorsqu'un capteur unique fournit une observation sur l'environnement du véhicule. Nous allons maintenant étendre cette présentation pour le cas multicapteur. Pour l'instant, nous faisons les mêmes hypothèses que pour la section 3.3. Nous considérons donc le cas dans lequel S capteurs fournissent au plus une observation à chaque instant.

Nous commençons par présenter le programme bayésien avant d'illustrer la méthode par des résultats issus de simulation.

5.3.1 Programme bayésien

5.3.1.1 Variables

Comme pour le cas monocapteur, nous avons à considérer les variables $X^{0:N}$ et $E_X^{0:N}$. Nous devons par contre considérer les observations des différents capteurs entre l'instant initial et l'instant final. Ces observations sont représentées par la conjonction de variables notée $Z_{1:S}^{0:N}$.

Ces variables définissent la distribution de probabilité conjointe :

$$P(X^{0:N} E_X^{0:N} Z_{1:S}^{0:N}). \quad (5.41)$$

5.3.1.2 Décomposition

Pour décomposer cette distribution de probabilité conjointe, nous faisons deux hypothèses d'indépendance conditionnelle :

- connaissant l'état de la cellule considérée à l'instant k , l'observation d'un capteur s à cet instant est indépendante des observations de tout capteur aux autres instants, ce qui se traduit :

$$\forall l \neq k, P(Z_s^k | Z_{1:S}^l E_X^k X^k) = P(Z_s^k | E_X^k X^k);$$

- connaissant l'état de la cellule considérée à l'instant k , l'observation d'un capteur s à cet instant est indépendante de l'observation d'un autre capteur à ce même instant k , ce qui se traduit :

$$p(Z_s^k | Z_{1:S}^k E_X^k X^k) = P(Z_s^k | E_X^k X^k).$$

Ces hypothèses nous permettent de décomposer la distribution de probabilité conjointe (5.41) sous la forme :

$$P(X^{0:N} E_X^{0:N} Z_{1:S}^{0:N}) = \left(\begin{array}{l} P(E_X^0 X^0) \left(\prod_{s=1}^S P(Z_s^0 | E_X^0 X^0) \right) \\ \left(\prod_{k=0}^N \prod_{s=1}^S P(E_X^k X^k | E_X^{k-1} X^{k-1}) P(Z_s^k | E_X^k X^k) \right) \end{array} \right). \quad (5.42)$$

5.3.1.3 Formes paramétriques

Dans la décomposition (5.42), chacune des distributions

$$P(Z_s^k | E_X^k X^k), \quad s = 1 \dots S, k = 0 \dots N$$

est donnée par la modèle capteur associé au capteur S . Les distributions

$$P(E_X^k X^k | E_X^{k-1} X^{k-1}), \quad k = 1 \dots N$$

sont quant à elles données par le modèle dynamique du système. Enfin la distribution $P(E_X^0 X^0)$ est choisie uniforme si l'on ne connaît rien sur notre environnement à l'instant initial.

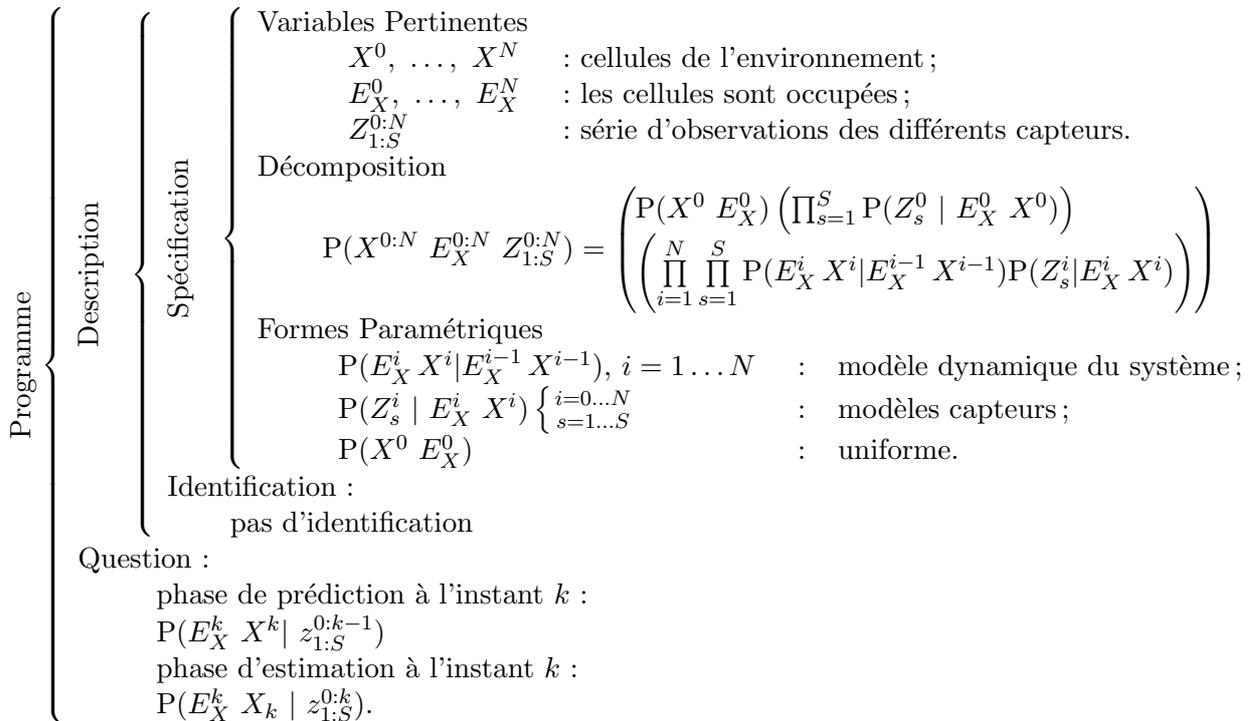


FIG. 5.8: Le filtre d'occupation bayésien multicapteur : programme bayésien

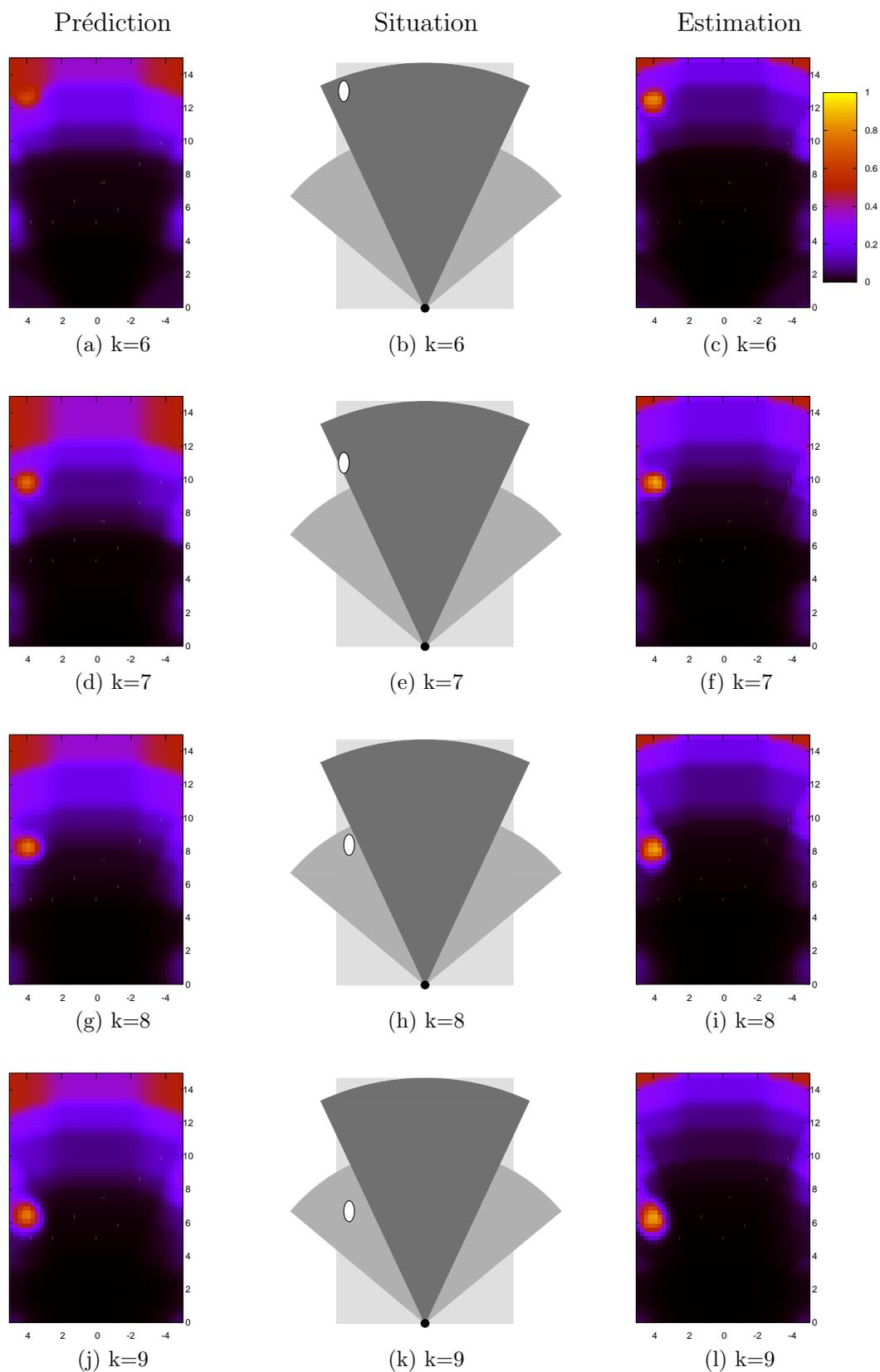


FIG. 5.9: Exemple d'estimation dynamique de grille d'occupation dans le cas multicapteur. Le cycab avance vers une cible immobile.

5.3.1.4 Questions

Comme dans le cas monocapteur, nous posons à chaque instant k la succession de questions :

$$P(E_X^k X^k | z_{1:S}^{0:k-1}) \quad (5.43)$$

pour la phase d'estimation, et :

$$P(E_X^k X^k | z_{1:S}^{0:k}) \quad (5.44)$$

pour la phase de prédiction.

Pour ces deux questions, le résultat de l'inférence donne :

$$P(E_X^k X^k | z_{1:S}^{0:k-1}) = \int_{\substack{X^{k-1} \\ E_X^{k-1}}} \left(P(E_X^{k-1} X^{k-1} | z_{1:S}^{0:k-1}) P(E_X^k X^k | E_X^{k-1} X^{k-1}) \right); \quad (5.45)$$

et pour la phase d'estimation :

$$P(E_X^k X^k | z_{1:S}^{0:k}) = \frac{1}{\alpha} \left(\prod_{s=1}^S P(z_s^k | E_X^k X^k) \right) P(E_x^k X^k | z_{1:S}^{0:k-1}). \quad (5.46)$$

La figure 5.8 résume le programme bayésien que nous venons de détailler.

5.3.2 Illustration

Pour cette illustration, nous reprenons la situation décrite dans la sous-section 3.3.3, et résumée dans la figure 3.11. La grille que nous cherchons à estimer est donc 2-D, deux capteurs de champs de vue différents fournissant des observations sur l'environnement du véhicule.

La figure 5.9 présente une séquence de résultats à quatre instants successifs. Comme précédemment, la colonne de gauche présente le résultat de la phase de prédiction à l'instant k , la colonne centrale présente une vue d'oiseau de la situation au même instant, et la colonne de droite présente le résultat de la phase d'estimation, toujours au même instant k .

Lors de cette expérience, le véhicule portant le capteur avance en direction d'une cible immobile, qui se trouve successivement dans le champ du premier puis du second capteur. Nous supposons que lorsque l'objet est présent dans le champ de vue d'un capteur, il est systématiquement détecté par ce capteur.

Concernant les phases de prédiction, on retrouve les mêmes remarques que dans le cas monocapteur : les mouvements du véhicule ont une incidence sur la grille d'occupation prédite.

Le moment le plus intéressant de cette figure se trouve entre les instants $k = 7$ et $k = 8$. Entre ces deux instants, l'objet a quitté le champ de vue du premier capteur, dans lequel il était situé les instants précédents. Par contre, il est entré dans le champ de vue du second capteur, duquel il était absent jusqu'à présent. Puisque nous cherchons

à fusionner des informations capteurs et ceci au cours du temps, le second capteur doit pouvoir profiter des informations délivrées par le premier. Ce « partage » d'information se fait naturellement grâce à la phase de prédiction. Puisque cette phase de prédiction ne dépend que du résultat de l'estimation au pas de temps précédent, son résultat ne dépend en rien des capteurs, et en particulier ne dépend pas de leurs champs de vue respectifs. Ainsi on voit sur la figure 5.9(g) une zone de forte probabilité d'occupation, liée aux observations du premier capteur aux instants précédents. Cette zone est située hors du champ de vue de ce premier capteur, mais dans le champ de vue du second capteur. Lors de la phase d'estimation suivante, dont le résultat est présenté par la figure 5.9(i), cette connaissance est naturellement prise en compte. Ainsi, même si le second capteur observe pour la première fois un objet, la probabilité des cellules correspondantes est déjà forte, puisque l'observation est en accord avec la prédiction.

5.4 Conclusion

Dans ce chapitre, nous avons présenté le rapprochement de deux méthodes classiques, le filtre bayésien et la grille d'occupation. Ce rapprochement débouche sur une représentation originale d'un environnement dynamique, que nous avons appelé *filtre d'occupation bayésien*.

Le filtrage des grilles d'occupation permet d'accroître la robustesse de l'estimation aux défaillances des capteurs. Elle permet également d'avoir une idée de l'occupation de cellules situées en dehors du champ de vue du capteur, dans le cas où le capteur se déplace. Dans ce filtrage, le modèle dynamique n'est bien sûr pas attaché à un objet, puisque cette notion est absente des grilles d'occupation, mais aux cellules elles-mêmes. Compte tenu de la grille à un instant k , il est en effet possible d'estimer l'état de la grille à l'instant $k + 1$.

Nous avons également présenté un algorithme d'approximation spécifique pour la phase de prédiction du filtrage. Cet algorithme permet de répondre au coût calculatoire de manière très efficace.

Pour reprendre la même progression que dans les chapitres précédents, nous nous sommes dans ce chapitre restreint à une unique observation pour chaque capteur à chaque instant. Le but du chapitre suivant est d'étendre notre méthode au cas général, lorsque plusieurs objets génèrent plusieurs observations à chaque instant.

Chapitre 6

Le filtre d'occupation bayésien : cas multiobjet

Nous présentons maintenant le filtre d'occupation bayésien quand plusieurs objets sont présents dans l'environnement du véhicule, et donc que le ou les capteurs fournissent plusieurs observations à chaque instant.

Nous commençons par revenir sur les difficultés de la poursuite multicible, mentionnées en introduction, et présentons les algorithmes classiques d'association de données.

Nous présentons ensuite l'extension finale de notre représentation aux environnements dynamiques et encombrés, d'abord dans le cas monocapteur, puis dans le cas multicapteur. Nous illustrons les capacités de cette méthode à gérer les occultations temporaires entre objets, les apparitions d'objet, etc, sur des exemples de scènes dynamiques.

6.1 Poursuite multicible : problèmes spécifiques et solutions classiques

La principale difficulté de la poursuite multiobjet vient du fait qu'on ne sait pas associer *a priori* une observation du capteur à un objet présent dans le volume d'observation du capteur. Cette difficulté est renforcée si on considère que le capteur peut générer des *fausses alarmes*, c'est à dire des observations ne correspondant à aucun objet existant. Enfin, la difficulté est encore renforcée si on ne connaît pas le nombre d'objets présent dans le volume d'observation, et si ce nombre peut évoluer au cours du temps.

La figure 6.1 présente les différentes phases d'un algorithme classique de poursuite multiobjet. Ces algorithmes cherchent en général à maintenir une liste de *pistes*, c'est à dire une liste d'objets identifiés et actuellement poursuivis. Quand une nouvelle série d'observations capteur est disponible, la première chose à faire est d'associer ces nouvelles observations aux pistes maintenues, c'est à dire de décider quelle piste est à l'origine de chaque observation. Au vu de cette association, La phase de maintenance consiste à décider si on doit :

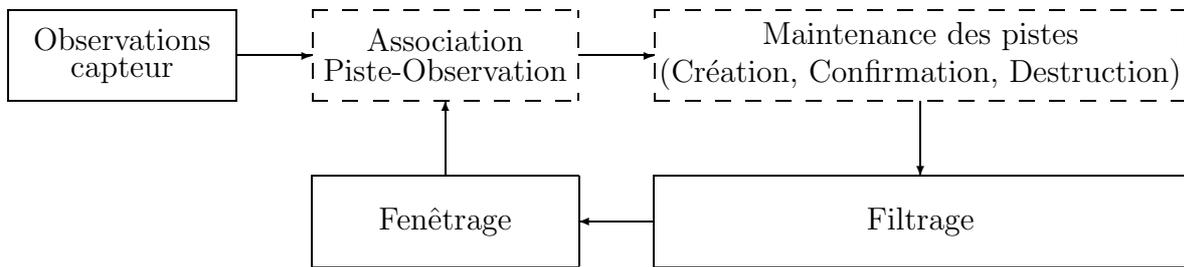


FIG. 6.1: Phases d'un algorithme classique de poursuite multipiste [BP00]. Les phases inscrites dans des cadres en pointillés sont spécifiques à la poursuite multiobjet.

- créer de nouvelles pistes, si un nouvel objet non encore poursuivi est entré dans le champ de vue du capteur ;
- détruire d'anciennes pistes, si des objets poursuivis ont disparu du champ de vue du capteur ;
- maintenir une piste existante, qu'une nouvelle observation lui ait été attribuée ou non.

La phase de filtrage n'est pas spécifique à la poursuite multipiste, puisqu'elle consiste pour chaque piste en la mise à jour et la prédiction de l'état de l'objet. Ceci est généralement fait à l'aide d'un filtre de Kalman étendu, ou plus récemment à l'aide d'un filtre à particules.

Enfin la dernière phase est une phase de fenêtrage. Cette phase de fenêtrage a pour but de simplifier la phase d'association : une observation capteur pourra être considérée comme pouvant provenir de cette piste si sa distance à l'état prédit de l'objet est inférieure à un seuil. Déterminer ce seuil n'est pas trivial. La « fenêtre de validation » doit être suffisamment grande pour qu'un impact provenant d'une piste s'y trouve et suffisamment petite pour pouvoir efficacement filtrer les impacts.

Les deux phases spécifiques à la poursuite multicible, association piste-observation et maintenance des pistes, sont en général résumées sous le terme d'*association de données*.

De nombreux algorithmes ont été développés pour résoudre la phase d'association piste-observation, qui peut être considérée comme la plus difficile. Très souvent, la distinction entre les trois phases association, maintenance et filtrage n'est pas si nette que dans la figure 6.1, et l'algorithme les intègre toutes les trois.

Nous allons maintenant présenter les algorithmes les plus classiques de l'association de données. Nous nous limitons pour l'instant au cas monocapteur. L'extension au cas multicapteur sera exposé par la suite.

6.1.1 Algorithmes classiques d'association de données

Nous présentons ici les algorithmes les plus classiques d'association et de filtrage multiobjet. La littérature étant abondante sur le sujet [BSF88][BSL95][BP00], nous nous

contentons ici d'une présentation très synthétique des méthodes.

6.1.1.1 Méthode du plus proche voisin

Cette méthode [Sea71][Sin73] est la plus simple et la plus utilisée pour répondre au problème de l'association. C'est une méthode purement séquentielle : pour chaque nouvel ensemble d'observations, le but est de trouver l'association la plus probable entre une observation et une piste existante ou entre une nouvelle observation et l'hypothèse d'une nouvelle piste.

Une analyse des performances de cette méthode est présentée dans [LBS96]

Le principal inconvénient de la méthode du plus proche voisin tient au fait que la décision sur l'association est immédiate et irrévocable : l'hypothèse la plus probable est considérée comme vraie. Son principal avantage, et donc sa popularité, tient dans son faible coût calculatoire.

L'augmentation importante de la puissance de calcul tend à rendre populaire des méthodes utilisant *toutes* les observations voisines d'une piste pour la mise à jour de cette piste.

6.1.1.2 Filtre à association de données probabilistes (PDAF)

Le PDAF¹ [BST74][BST75] n'est pas un filtre de poursuite multiobjet, mais permet de prendre en compte la présence de fausses alarmes. Plusieurs observations peuvent donc être retournées à un instant k par le capteur, alors que l'on suppose la présence d'un seul objet dans le champ de vue du capteur.

La méthode du plus proche voisin choisirait l'observation la plus proche de l'état prédit de l'objet comme étant l'observation correspondant à l'objet, et se servirait de cette observation pour mettre à jour le filtre de Kalman permettant de poursuivre l'objet.

Au contraire, Le PDAF tient compte de toutes les observations capteur se trouvant dans la fenêtre de validation pour mettre à jour le filtre de Kalman. Cette mise à jour se fait sous les hypothèses suivantes :

H. 1 *Une mesure peut provenir d'un objet ou être une fausse alarme.*

H. 2 *À un instant donné, un objet peut être à l'origine d'aucune ou d'au plus une mesure.*

L'innovation² classique du filtre de Kalman est remplacée par une combinaison d'innovations associées à chaque hypothèse d'association, et pondérée par la probabilité de l'association. Le PDAF commet donc l'approximation fondamentale qui consiste à représenter la distribution *a priori* par une distribution Gaussienne, donc unimodale, alors qu'il s'agit en réalité d'une distribution multimodale.

¹Probabilistic Data Association Filter

²voir la sous-section 5.1.3.1

6.1.1.3 Filtre joint à association de données probabilistes (JPDAF)

Le JPDAF³ [FBSS83] est une extension du PDAF pour la poursuite multiobjet. Pour cette méthode, le nombre d'objets M est supposé connu et constant.

L'estimation de l'état de chaque objet se fait selon un filtre de Kalman comme dans le PDAF. L'hypothèse d'approximation selon laquelle chaque distribution *a priori* est une distribution Gaussienne est à nouveau faite. La seule différence entre PDAF et JPDAF réside dans l'évaluation des probabilités des associations.

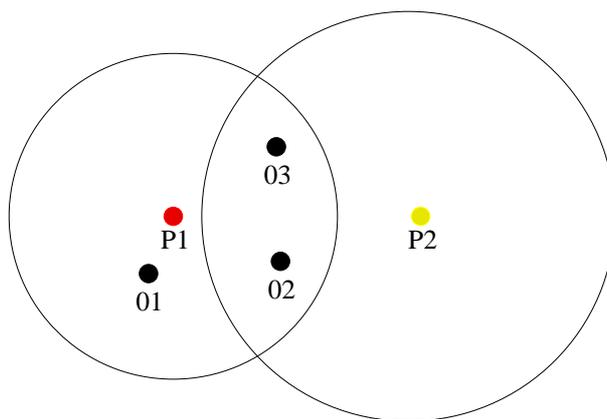


FIG. 6.2: Exemple de situation conflictuelle. P1 et P2 représentent les états prédits de deux objets poursuivis. Les cercles représentent les fenêtres de validations associées à ces prédictions. O1, O2 et O3 représentent les dernières observations capteurs.

La figure 6.2 permet de comprendre de manière intuitive le principe du JPDAF dans une situation conflictuelle. Dans cet exemple, deux pistes sont formées et poursuivies. Les états prédits de ces deux pistes à un instant donné sont représentées par les points P1 et P2, la fenêtre de validation de chacune des pistes par des cercles. Les deux fenêtres de validation se chevauchent partiellement. À ce même instant, le capteur envoie trois observations. La première est située dans la fenêtre de validation de la première piste uniquement, les deux autres sont situées dans la zone commune aux deux fenêtres de validation. Comme les trois observations sont dans la fenêtre de validation de la première piste, la mise à jour de cette piste se fera en tenant compte de ces trois observations, sous forme d'une somme pondérée par la probabilité d'association. Toutefois, la probabilité d'association, et donc le poids, des observations O2 et O3 sera réduite pour tenir compte de leur présence dans la fenêtre de validation de la deuxième piste. La mise à jour de la deuxième piste sera faite en tenant compte des deux observations O1 et O2.

Récemment, le JPDAF a été adapté à l'utilisation de filtres à particules plutôt que de filtres de Kalman [SBFC01]. Le but est essentiellement de pouvoir représenter des distributions de probabilité multimodales, ce que ne peut pas faire le filtre de Kalman.

³Joint Probabilistic Data Association Filter

Création et destruction de pistes Le JPDA original suppose un nombre connu et constant de pistes. Il a depuis été étendu pour gérer la création et la destruction de pistes [MES94].

La création d'une nouvelle piste est envisagée pour les observations ayant obtenues une faible probabilité d'association avec une piste existante.

Pour chaque piste, et pour chaque tentative de nouvelle piste, la probabilité que cette piste corresponde à un objet existant est estimée. La destruction de pistes existantes et la confirmation de nouvelles pistes sont décidées par un seuillage sur cette probabilité.

6.1.1.4 Filtre à hypothèses multiples (MHT)

Le nombre d'objet est supposé supérieur à un mais non connu *a priori* et pouvant croître au cours du temps. Les hypothèses **H. 1** et **H. 2** des PDAF et JPDAF sont conservées pour cette méthode. Le principe du MHT⁴ est de considérer systématiquement toutes les hypothèses d'association possibles et de créer une piste pour chacune de ces hypothèses. L'idée est d'attendre que de nouvelles observations capteurs lèvent l'ambiguïté et permettent de décider quelles hypothèses sont à abandonner et quelles hypothèses sont à garder.

Hypothèse	01	02	03
1	P1	P2	N
2	P1	N	P2
3	P1	N	N
4	N	P1	P2
5	N	P1	N
6	N	P2	P1
7	N	P2	N
8	N	N	P1
9	N	N	P2
10	N	N	N

TAB. 6.1 – Hypothèses d'associations possibles pour l'exemple présenté figure 6.2. N signifie que l'observation est considérée comme provenant d'une nouvelle piste.

Si l'on reprend la situation décrite par la figure 6.2, dix hypothèses d'association sont possibles. Ces hypothèses sont listées dans la table 6.1. Dans la cas du JPDAF, ces dix hypothèses sont bien prises en compte, mais combinées pour ne maintenir que deux pistes. Dans le cas du MHT, les dix hypothèses d'association vont être maintenues. Le nombre de pistes existantes après cette phase d'association passe donc de deux à dix. L'ensemble des associations possibles au cours du temps est souvent représenté sous forme d'arbre. À chaque branche de l'arbre est associée une valeur de probabilité d'existence.

La taille de l'arbre croissant exponentiellement avec le nombre d'observation capteur, cette méthode est inexploitable tel quel. Il est nécessaire d'élaguer les branches de l'arbre

⁴*Multiple Hypothesis Tracking*

dont les probabilités sont trop faibles. Le nombre de branches supprimées doit être assez élevé pour que la complexité de l'algorithme soit réduite. Il peut être cependant fatal d'éliminer trop rapidement une hypothèse de probabilité à un instant donné qui se serait révélée intéressante par la suite.

Les variations et optimisations autour de cette méthode de filtrage sont nombreuses, nous renvoyons par exemple à [BP00], chapitre 16 pour plus de détails.

6.1.1.5 Filtre Probabiliste à Hypothèses Multiples (PMHT)

Le PMHT⁵ se distingue des précédentes sur deux points essentiels. Premièrement, elle cherche à estimer l'état des cibles à tout moment d'un intervalle de temps, en utilisant toutes les observations capteur disponibles pendant cet intervalle de temps. En ce sens, il s'agit plus d'un algorithme de lissage que d'un algorithme de filtrage. Deuxièmement, l'hypothèse **H. 2** est remplacée par l'hypothèse suivante :

H. 3 *À un instant donné, un objet peut être à l'origine d'aucune ou de plusieurs mesures.*

Le PMHT, qui a été initialement proposé dans [SL94][SL95] puis étendu dans [GLJ97], cherche à estimer un nombre M connu de piste sur un intervalle de temps. Pour cela, M filtres de Kalman sont menés indépendamment les uns des autres. Comme pour le PDA et le JPDA, une piste est lissée en utilisant une somme pondérée de toutes les observations possibles. Le poids (*i.e.* la vraisemblance) d'une association est estimé par un algorithme EM⁶ [DLR77][Bil98].

En pratique, la mise en œuvre du PHMT peut poser plusieurs problèmes, parmi lesquels :

- l'initialisation de l'algorithme, qui nécessite d'affecter une valeur initiale aux vecteurs d'état et d'association pour l'ensemble des instants de l'intervalle de temps. Le PMHT est très sensible à la qualité de cette initialisation ;
- le critère d'arrêt de l'algorithme EM doit être fixé. On peut utiliser un nombre fixe d'itérations ou utiliser un critère d'arrêt mesurant la différence entre deux estimées successives.

Par contre, l'algorithme PMHT peut être facilement étendu à la présence de plusieurs capteurs.

6.1.1.6 Le filtre à particules pour le suivi multicible

Des travaux ont mis en avant les avantages de filtre à particules, que nous avons présenté rapidement dans la section 5.1.3.3, pour traiter du filtrage d'objets en environnement bruité, conduisant au problème d'association des données. L'une des propriétés du filtre à particule est de résumer par l'ensemble des particules et de leur poids toute l'information apportée par les observations entre l'instant initial et l'instant courant. En particulier,

⁵*Probabilistic Multiple Hypothesis Tracking*

⁶expectation-maximization

le nuage courant résume directement les associations passées. De plus, aucune hypothèse supplémentaire, comme supposer la distribution *a priori* Gaussienne dans le cas du PDA et du JPDA n'est nécessaire.

Les travaux de Gordon [Gor97] proposent une extension du filtre à particules pour la poursuite d'une unique cible en présence de fausses alarmes.

Pour l'extension du filtre à particules au suivi multiobjet, l'espace d'état des particules est le plus souvent choisi comme étant celui du vecteur (S_1^t, \dots, S_M^t) des M objets poursuivis. Un seul filtre est donc nécessaire pour la poursuite des différents objets, chaque particule constituant une hypothèse pour l'état de chaque objet. Un autre choix possible est de mener un filtre par objet poursuivi, comme dans les travaux de Schulz [SBFC01] que nous avons déjà mentionné.

La difficulté du filtre à particules pour le suivi multiobjet réside alors classiquement dans l'estimation des probabilités d'association. Les travaux d'Orton [OF02] prennent en compte la seule probabilité *a priori* des associations, qui ne dépend pas des observations. Hue [HLCF02][Hue03] utilise un échantillonneur de Gibbs afin d'estimer au mieux la probabilité *a posteriori* d'association à l'aide de toutes les mesures disponibles. Les réalisations obtenues avec l'échantillonneur de Gibbs sont également mises à profit pour éventualiser l'apparition ou la disparition d'un objet. Cette extension au nombre variable d'objets a été menée de manière théorique dans les travaux de Hue.

6.1.2 Extension au cas multicateur

Comme nous l'avons mentionné dans la section 3.3.1, le choix d'une architecture de fusion et de poursuite est primordial lors du développement d'une application de poursuite multiobjet multicateur.

Un choix architectural important concerne le niveau auquel est fait le filtrage et la maintenance des pistes. Il existe en gros deux méthodes. La première consiste à faire une

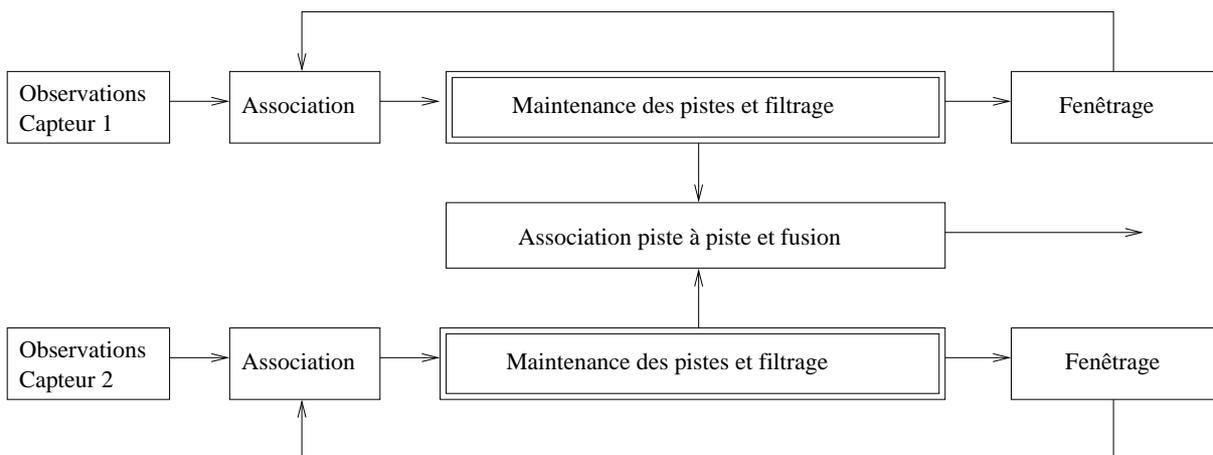


FIG. 6.3: Architecture de poursuite multicateur « niveau capteur ». Chaque capteur fait sa propre poursuite multiobjet, les pistes des différents capteurs sont ensuite fusionnées.

poursuite multicible au niveau du capteur. Chaque capteur possède donc sa propre logique d'association de données, de maintenance et de filtrage des pistes. Le principe de cette méthode est illustré par la figure 6.3.

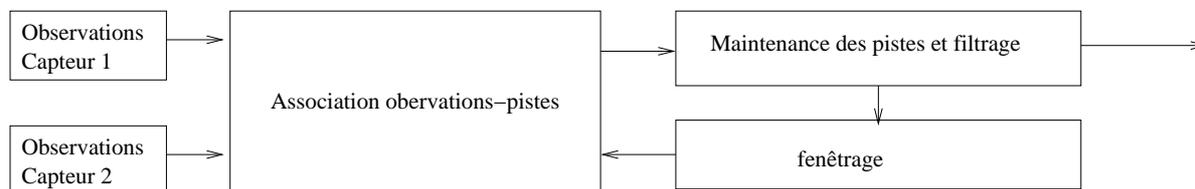


FIG. 6.4: Architecture de poursuite multicapteur « globale ». Les observations capteurs sont directement passées au niveau global.

La deuxième méthode consiste à considérer les observations des différents capteurs de manière globale, pour réaliser l'association de données, la maintenance et le filtrage des pistes. Le principe de cette méthode est illustré par la figure 6.4.

Méthode	Avantages	Inconvénients
niveau capteur	<ul style="list-style-type: none"> – moins de transfert d'informations ; – alignement temporel des pistes est plus direct ; – plus robuste aux pannes capteurs. 	<ul style="list-style-type: none"> – besoin d'une logique spéciale pour l'association piste à piste ; – association et poursuite moins précise.
globale	<ul style="list-style-type: none"> – Adaptation des algorithmes d'association est directe, particulièrement le MHT ; – Association et poursuite plus précise. 	<ul style="list-style-type: none"> – transfert d'informations ; – alignement temporel des observations capteurs non direct ; – moins robuste aux pannes capteurs.

TAB. 6.2 – Principaux avantages et inconvénients des architectures globales et niveau capteur pour la poursuite multicapteur.

La table 6.2 liste les principaux avantages et inconvénients de chacune des architectures. Dans le développement d'une application de poursuite multiobjet multicapteur, par exemple sur un véhicule automobile, le taux de transfert de données entre les différents composants du système peut être un critère de choix décisif. L'architecture « niveau capteur » demande moins de transfert de données, puisque seules les pistes validées sont transmises, et non toutes les observations des capteurs.

Le principal inconvénient de l'architecture « niveau capteur » réside dans le besoin de définir une logique spéciale pour fusionner les pistes des différents capteurs. On trouve une revue des différentes méthodes pour l'association piste à piste dans [BP00], chapitre 9.

L'adaptation des algorithmes classiques d'association de données est plus directe dans le cas d'une architecture globale. Par exemple, Houles [HBS89] détaille l'application du PDA à la poursuite d'une cible à l'aide d'un radar et d'un capteur infrarouge. L'association des données est faite de manière séquentielle, les données du premier capteur servant à faire une première estimation, qui est ensuite remise à jour en utilisant les données du second capteur.

Cette méthode séquentielle permet également une application directe du MHT à une architecture de fusion de niveau global [KL88][Sha93].

D'autres extensions des méthodes classiques utilisent de manière simultanée les mesures des différents capteurs. Par exemple, [PO93] détaille l'application du JPDA au cas multicapteur sous l'hypothèse que les bruits des différents capteurs sont indépendants.

Concernant les filtres à particules, Hue [Hue03] montre que l'adaptation de son filtre au cas multicapteur dans une architecture globale est directe si on suppose l'indépendance des mesures fournies par des capteurs différents.

6.2 Extension de notre méthode au cas multiobjet

Dans le chapitre 4, nous avons montré comment la grille d'occupation pouvait être estimée en tenant compte d'une liste d'observations d'un ou plusieurs capteurs à un instant donné.

Dans le chapitre précédent, nous avons présenté comment l'estimation des grilles d'occupation pouvait intégrer l'historique des observations d'un ou plusieurs capteurs, chaque capteur fournissant au plus une observation à chaque instant.

Nous faisons ici la synthèse de ces deux chapitres en montrant comment les grilles peuvent être estimées dynamiquement lorsque chaque capteur fournit à chaque instant une liste d'observations. Comme nous ne faisons aucune hypothèse sur le nombre d'objets présent autour du véhicule, nous ne connaissons pas *a priori* le nombre d'observations fournies par le capteur à chaque instant.

Pour reprendre la même progression que dans les chapitres précédents, nous nous limitons dans cette section au cas monocapteur

Nous commençons par présenter le programme bayésien avant d'en illustrer l'utilisation sur des résultats issus de simulation.

6.2.1 Programme bayésien

6.2.1.1 Variables

Comme dans le cas dynamique monoobjet, nous avons à considérer les variables $X^{0:N}$ et $E_X^{0:N}$, qui représentent les cellules de la grilles et leur état à chaque instant.

Ici, nous devons considérer les différentes observations de notre capteur à chaque instant k . Chacune de ces observations est représentée par une variable Z_i^k , $i = 1 \dots O^k$, O^k étant le nombre d'observations retournées par le capteur à l'instant k . Nous notons \mathcal{Z}^k la

conjonction des observations du capteur à l'instant k , et $\mathcal{Z}^{0:N}$ la conjonction de l'ensemble des observations entre l'instant initial et l'instant final.

Comme dans le cas statique (voir section 4.1), nous considérons à chaque instant une variable d'association M^k . La conjonction des variables d'association à chaque instant est notée $M^{0:N}$.

Ces variables définissent la distribution de probabilité conjointe :

$$P(X^{0:N} E_X^{0:N} \mathcal{Z}^{0:N} M^{0:N}). \quad (6.1)$$

6.2.1.2 Décomposition

Pour décomposer la distribution de probabilité conjointe (6.1), nous faisons les mêmes hypothèses d'indépendance conditionnelle sur les observations que dans les sections 5.2 et 4.1 :

- sachant l'état de la cellule considérée à l'instant k , les observations du capteur à un instant k sont supposées indépendantes des observations du capteurs aux autres instants ;
 - sachant l'état de la cellule considérée à l'instant k et la valeur de la variable d'association, les différentes observations du capteur à l'instant k sont supposées indépendantes.
- Nous supposons de plus que les associations à chaque instant sont indépendantes.

Ces différentes hypothèses nous permettent d'écrire la décomposition :

$$P(X^{0:N} E_X^{0:N} \mathcal{Z}^{0:N} M^{0:N}) = \left(\begin{array}{l} P(E_X^0 X^0)P(M^0) \left(\prod_{i=1}^{O^0} P(Z_i^0 | M^0 E_X^0 X^0) \right) \\ \times \prod_{k=1}^N \left(\begin{array}{l} P(M^k)P(E_X^k X^k | E_X^{k-1} X^{k-1}) \\ \times \prod_{j=1}^{O^k} P(Z_j^k | M^k E_X^k X^k) \end{array} \right) \end{array} \right) \quad (6.2)$$

6.2.1.3 Formes paramétriques

Dans cette décomposition :

- les $P(M^k)$ sont choisies uniformes ;
- les $P(Z_j^k | [M^k = j] E_X^k X^k)$ sont données par le modèle capteur (voir section 4.1) ;
- les $P(Z_j^k | [M^k \neq j] E_X^k X^k)$ sont choisies uniformes. (voir section 4.1) ;
- les $P(E_X^k X^k | E_X^{k-1} X^{k-1})$ sont données par le modèle dynamique associé au système ;
- enfin $P(E_X^0 X^0)$ est choisie uniforme.

6.2.1.4 Questions

Notre but ici est toujours d'utiliser la structure de boucle prédiction-estimation du filtre bayésien, décrit par la figure 5.1.

Afin de conserver cette structure, nous posons à chaque instant k les deux questions :

- $P(E_X^k X^k | z^{0:k-1})$ pour la phase de prédiction ;
- $P(E_X^k X^k | z^{0:k})$ pour la phase d'estimation.

Pour la phase de prédiction, le résultat de l'inférence donne :

$$P(E_X^k X^k | z^{0:k-1}) = \int_{E_X^{k-1} X^{k-1}} P(E_X^k X^k | E_X^{k-1} X^{k-1}) P(E_X^{k-1} X^{k-1} | z^{0:k-1}). \quad (6.3)$$

La phase de prédiction s'appuie donc toujours sur le résultat à l'instant précédent de l'estimation et sur le modèle dynamique du système.

Et pour la phase d'estimation, le résultat de l'inférence donne :

$$P(E_X^k X^k | z^{0:k}) = \frac{1}{\alpha} \sum_{M^k=0}^{O^k} \left(P(M^k) \prod_{j=1}^{O^k} P(z_j^k | M^k E_X^k X^k) \right) \times P(E_X^k X^k | z^{0:k-1}) \quad (6.4)$$

La phase d'estimation consiste donc bien dans la confrontation entre la phase de prédiction et les différentes observations du capteur à l'instant k .

Comme nous l'avons mentionné dans le chapitre précédent, la probabilité d'occupation de la cellule à partir de ces questions est obtenue par une simple normalisation.

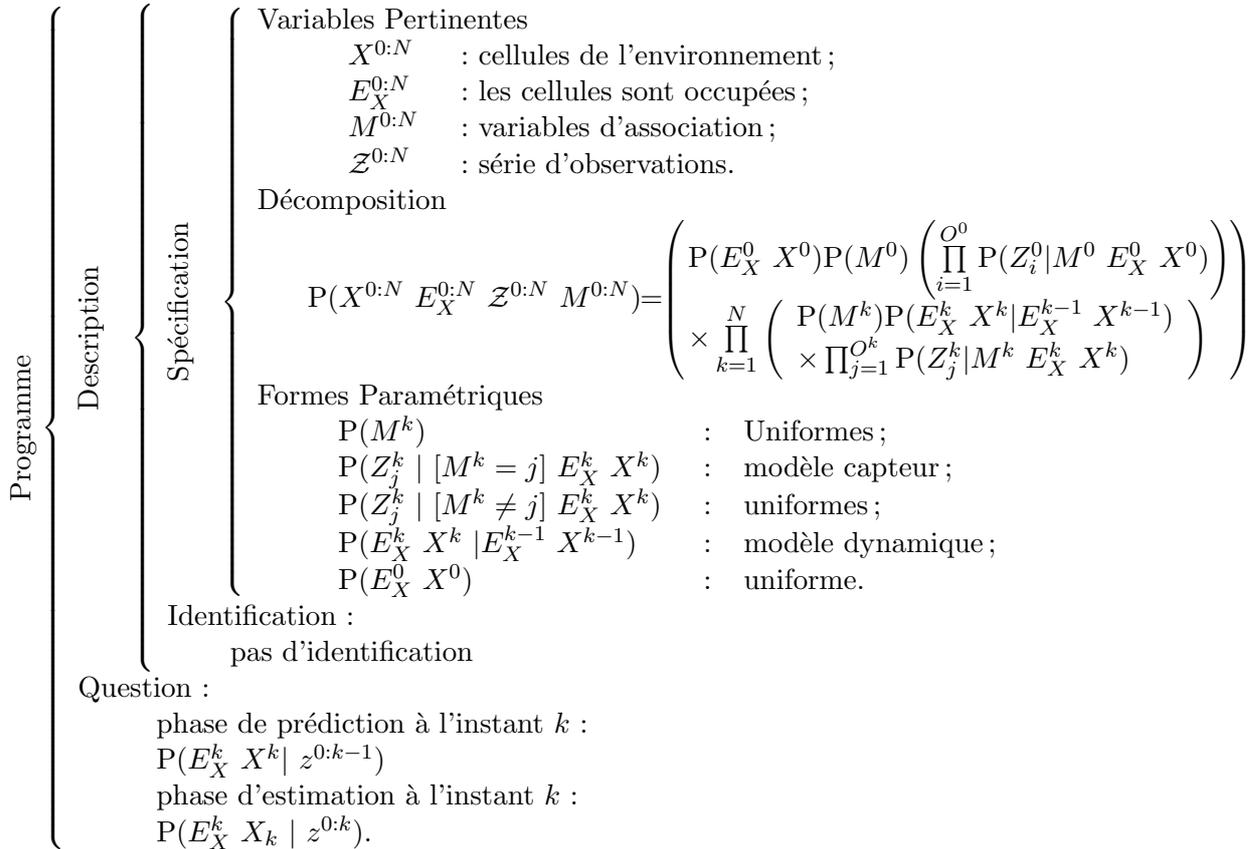


FIG. 6.5: Estimation dynamique des grilles d'occupation dans le cas monocapteur et multiobjet : programme bayésien.

La figure 6.5 résume le programme bayésien que nous venons de détailler.

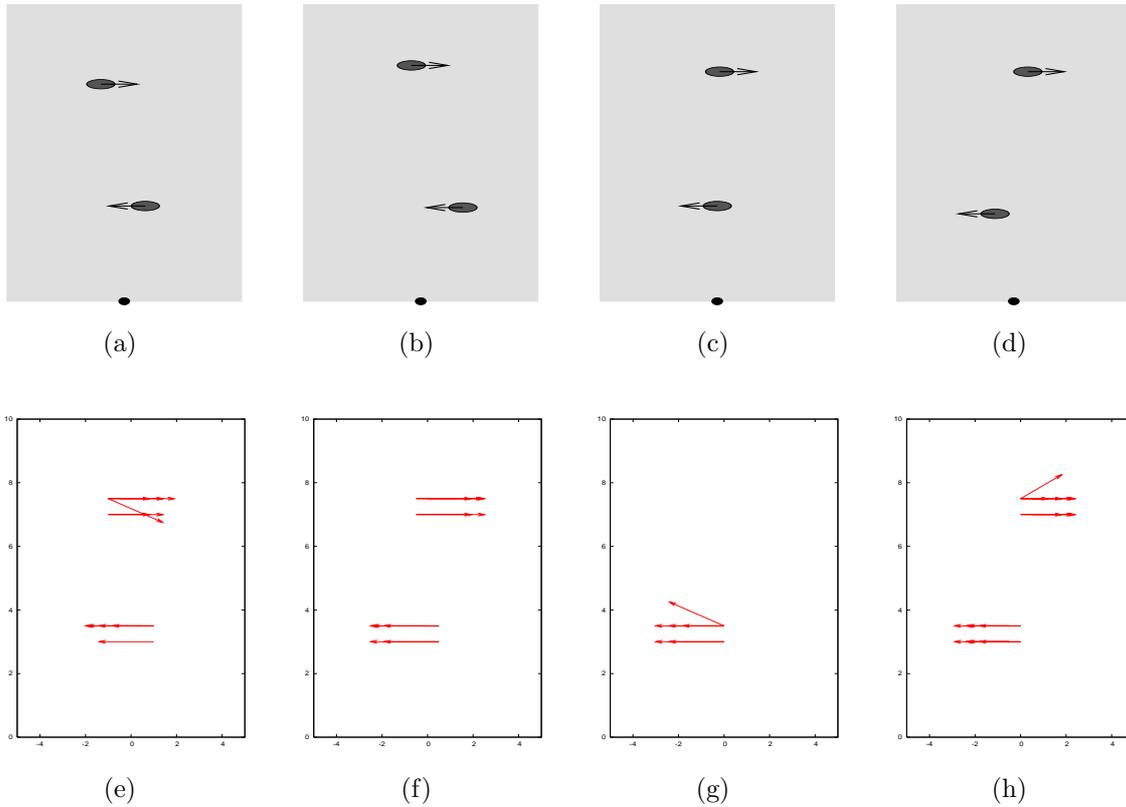


FIG. 6.6: Exemple de scène dynamique. L'estimation statique de la grille ne permet pas de gérer les occultations.

6.2.2 Illustrations

Jusqu'à présent, nous utilisons pour nos illustrations des situations statiques, le véhicule évoluant parmi des objets immobiles. Deux dimensions suffisaient donc à représenter la position relative (x, y) des cellules par rapport au véhicule.

Ici nous passons à des exemples dynamiques, les objets ne sont plus supposés immobiles. La représentation de l'environnement doit tenir compte de cette mobilité, et donner une estimation de la vitesse relative des objets par rapport au cycab.

Par conséquent, nos grilles d'occupation seront maintenant de dimension quatre, pour représenter la position (x, y) et la vitesse relative (\dot{x}, \dot{y}) des objets par rapport au cycab. En position, l'espace représenté par notre grille est de 10 mètres sur 10 mètres, le pas de discrétisation étant de 50 cm. Pour les vitesses relatives, \dot{x} varie entre -3 m.s^{-1} et 1 m.s^{-1} , et \dot{y} entre -2 m.s^{-1} et 2 m.s^{-1} . Dans le deux cas, le pas de discrétisation est de 20 cm.s^{-1} . Ces valeurs ont été choisies de manière à assurer un temps de calcul raisonnable, de l'ordre du dixième de seconde.

Les observations du capteur incluent une estimation de la position et de la vitesse de l'objet détecté.

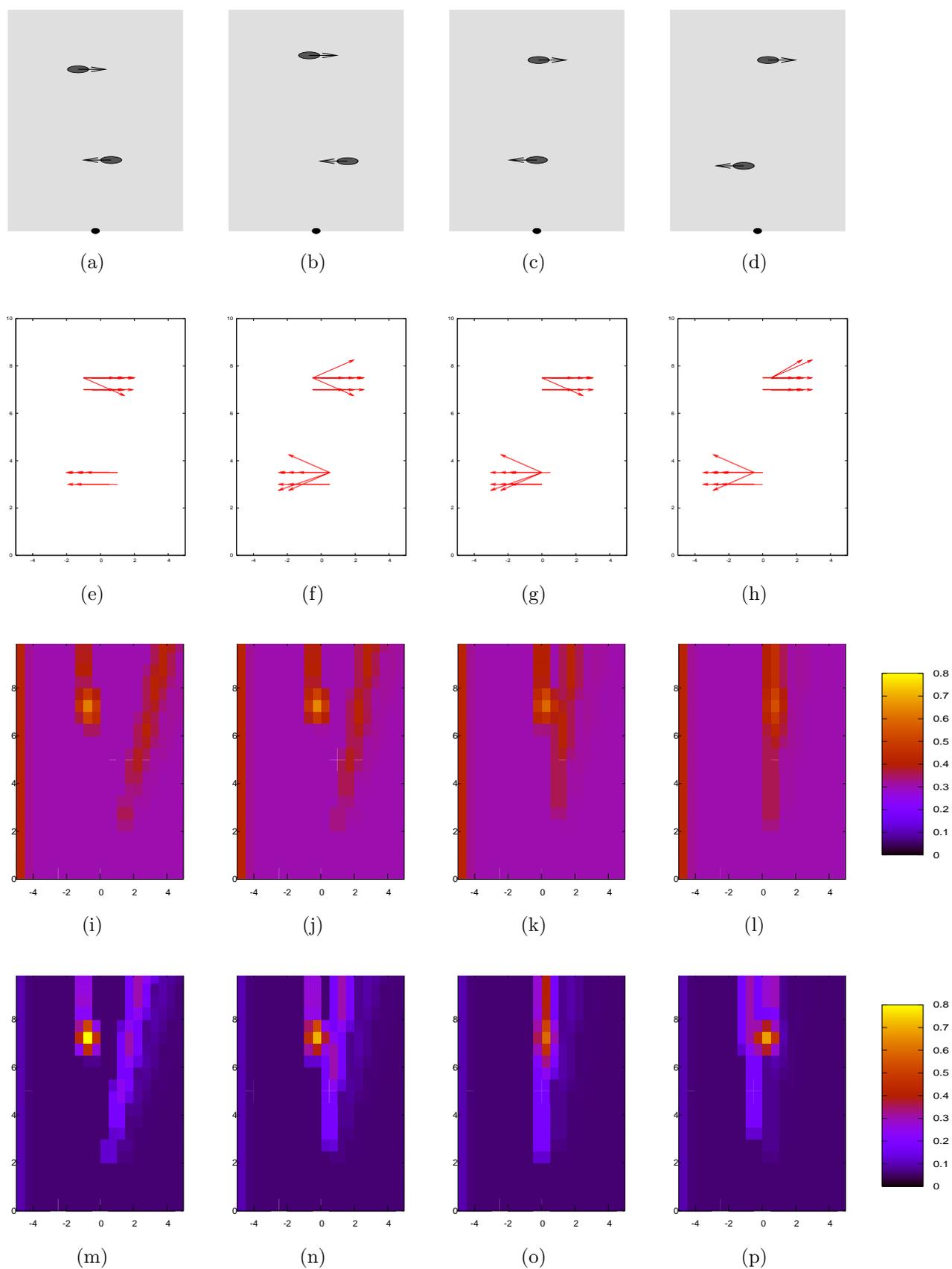


FIG. 6.7: Exemple dynamique de grille d'occupation. Un premier objet cache temporairement un second objet au capteur.

Dans ces exemples, l'algorithme d'approximation présenté dans la section 5.2.2 a été utilisé pour la prédiction de la probabilité d'occupation des cellules.

L'espace que nous avons à représenter est donc de dimension 5 (x, y, \dot{x}, \dot{y} et probabilité d'occupation). Comme la représentation complète de cet espace sur un plan n'est pas facile, nous avons choisi deux types de représentation partielle :

- soit la figure présente toutes les cellules de l'environnement, sous forme de vecteurs, l'origine du vecteur indiquant la position de la cellule, la direction et la longueur du vecteur représentant la vitesse de la cellule. Nous ne représentons alors que les cellules ayant une probabilité d'occupation supérieure à un seuil précisé dans la figure. En effet, la figure serait illisible si tous les vecteurs étaient présents. Cette présentation a déjà été utilisée dans la section 4.4 ;
- soit la figure ne présente qu'une partie des cellules de l'environnement, par exemple toutes les cellules correspondant à une vitesse relative fixée. Dans ce cas, la probabilité d'occupation de chaque cellule présentée est donnée par un code couleur, comme dans la majorité des figures que nous avons présentées jusqu'à présent.

6.2.2.1 Occultation d'objets

Ce premier exemple reprend la situation décrite en conclusion du chapitre 4, dans la figure 4.13. Dans cette situation, le véhicule portant le capteur est immobile, deux objets mobiles se croisent devant lui. L'objet situé le plus loin du capteur est temporairement caché au capteur par le premier objet. Dans la suite, le « premier objet » désigne toujours l'objet le plus proche du capteur, et le « deuxième objet » désigne toujours l'objet temporairement masqué. Cette séquence et le résultat de l'estimation statique de la grille, c'est à dire sans prise en compte de l'historique des observations, est reprise dans la figure 6.6. Lorsque le second objet est caché, aucune zone de forte probabilité d'occupation correspondant à cet objet ne se retrouve dans la grille.

La deuxième ligne de la figure 6.7 reprend la même séquence d'estimation de la grille d'occupation, incluant cette fois-ci une étape de prédiction entre chacune. Seules les cellules de probabilité supérieure à 0.5 sont représentées, sous forme d'un vecteur. On remarque sur la figure 6.7(g) que les deux objets sont identifiables sur cette grille, alors que le second est masqué. Ceci est bien sûr permis par la phase de prédiction.

Les troisièmes et quatrièmes lignes de la figure 6.7 permettent d'analyser plus finement ce résultat. Pour ces figures, nous choisissons notre deuxième méthode de représentation, la vitesse relative étant fixée comme nulle selon l'axe vertical, et comme étant égale à $0.8m.s^{-1}$ selon l'axe horizontal, ce qui est proche de la vitesse du deuxième objet.

La troisième ligne présente le résultat des phases de prédiction à différents instants, la quatrième ligne présente le résultat des phases d'estimation aux mêmes instants.

Une première constatation peut choquer sur le résultat des phases d'estimation, par exemple sur la figure 6.7(m) : si l'on retrouve bien une zone qui semble correspondre à la zone cachée par le premier objet (zone bleue clair, derrière l'objet situé approximativement en $(x = 3, 5; y = 1)$), sa probabilité d'occupation n'est pas égale à 0.5, mais est plus faible. Ceci est dû à la phase de prédiction, et à la présence des vitesses relatives dans la grille

d'occupation. En effet si un objet de vitesse relative ($\dot{x} = 0; \dot{y} = 0,8$) devait être caché à l'instant k par un objet allant en sens inverse, il aurait dû être observé à l'instant $k - 1$, et se retrouver dans la grille prédite à l'instant $k - 1$, présentée figure 6.7(i). Puisque ce n'était pas le cas, la probabilité qu'un objet de vitesse ($\dot{x} = 0; \dot{y} = 0,8$) soit caché derrière le premier objet est inférieure à 0,5.

On retrouve par contre sur le résultat de la phase de prédiction à l'instant $k + 1$ (la figure 6.7(j) pour notre séquence) une zone de la grille pour laquelle rien ne peut être prédit, correspondant à cette zone masquée à l'instant k , légèrement décalée du fait de la vitesse non nulle en cet endroit de la grille.

On remarque enfin sur les résultats des différentes phases de prédiction une « bande » de probabilité égale à 0,5, sur le bord gauche de la grille. En effet, nous prédisons ici la probabilité d'occupation de cellules « se déplaçant vers la droite ». Si des objets se déplaçant vers la droite devaient occuper ces cellules à l'instant k , ils auraient été présents à l'instant $k - 1$ à gauche de la grille, dans une zone pour laquelle nous n'estimons pas la probabilité d'occupation. Comme on ne peut pas dire à l'instant k si un tel objet existe ou non, on ne peut prédire l'état des cellules se trouvant sur le bord gauche de notre grille. Nous reprendrons ce cas pour illustrer l'apparition d'objet dans un prochain paragraphe.

Passons maintenant à l'occultation du deuxième objet proprement dite. Cette occultation correspond aux résultats présentés dans la troisième colonne. Sur la figure 6.7(k), qui présente le résultat de la phase de prédiction, on voit une zone de forte probabilité d'occupation en $(x = 7, y = 0)$ environ, qui correspond à notre deuxième objet. Pour le résultat de l'estimation au même instant, présenté sur la figure 6.7(k), une seule mesure capteur est utilisée, correspondant au premier objet se déplaçant vers la gauche, puisque cet objet masque au capteur le deuxième. On remarque cependant une zone de forte probabilité correspondant au deuxième objet. La probabilité de cette zone est plus faible que lors des estimations précédentes, puisque l'objet n'a pas été détecté. Elle reste cependant supérieure à 0,5, ce qui permet de conclure que les cellules correspondantes sont occupées. À l'instant suivant, cette zone se retrouve dans la grille d'occupation prédite (figure 6.7(l)), ce qui permet d'avoir une probabilité plus forte quand l'objet est à nouveau détecté (figure 6.7(p)) que s'il s'agissait de sa première détection.

L'inclusion de la phase de prédiction dans l'estimation des grilles d'occupation permet donc de gérer les occultations entre objets de manière naturelle, sans ajouter de logique supplémentaire.

6.2.2.2 Apparition d'objets et fausses alarmes

Comme nous l'avons signalé plus haut, un des aspects important - et difficile - de la poursuite multiobjet est la gestion des pistes. Il faut en particulier être capable de décider à un instant donné si une nouvelle mesure doit avoir pour conséquence la création d'une nouvelle piste, ou bien si elle doit être considérée comme une fausse alarme.

La figure 6.8 présente un premier exemple d'apparition d'objet dans le champ de vue du capteur. Un objet situé auparavant hors champ de vue du capteur entre dans celui-ci. Nous ne représentons dans cette figure que les cellules de l'environnement ayant pour

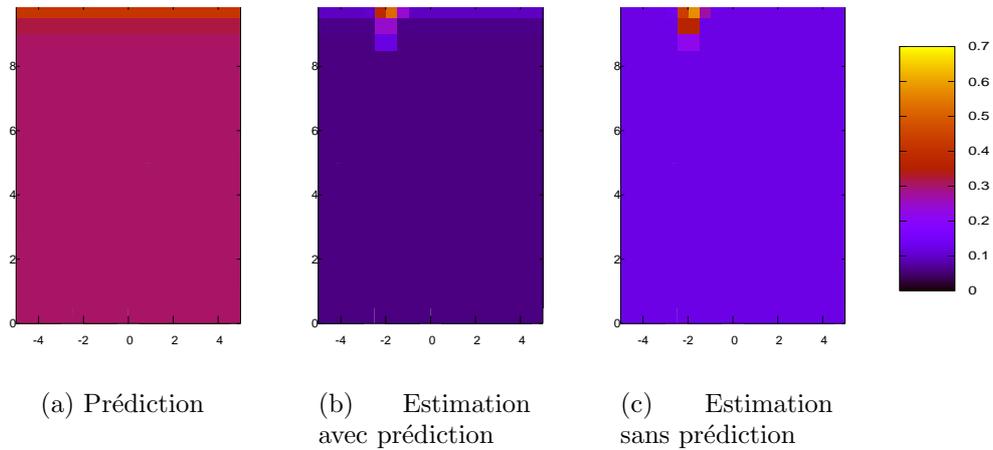


FIG. 6.8: Premier exemple d'apparition d'objet. À gauche, résultat de la phase de prédiction à l'instant k , pour $(\dot{x} = -1, 5; \dot{y} = 0)$. Au centre, résultat de la phase d'estimation prenant en compte la prédiction. À droite, résultat de l'estimation ne prenant pas en compte la prédiction.

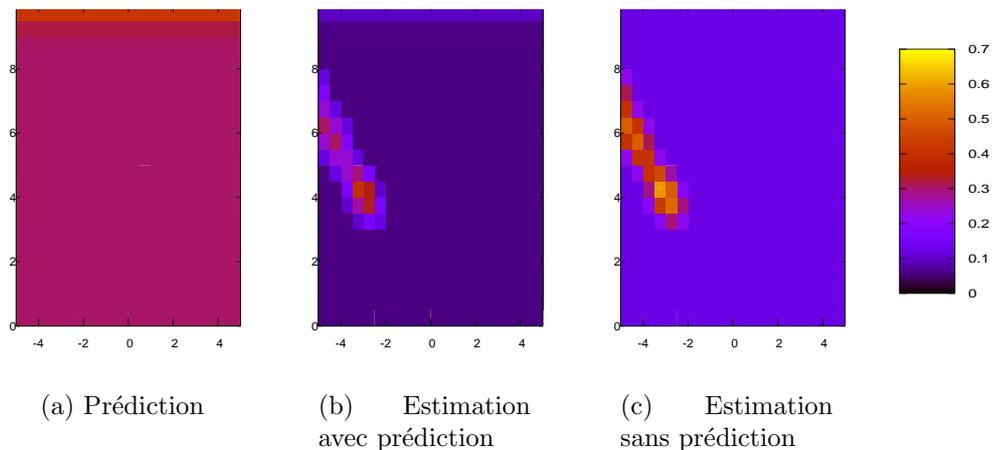


FIG. 6.9: Deuxième exemple d'apparition d'objet. À gauche, résultat de la phase de prédiction à l'instant k . Au centre, résultat de la phase d'estimation prenant en compte la prédiction. À droite, résultat de l'estimation ne prenant pas en compte la prédiction.

vitesse relative $(\dot{x} = -1, 5; \dot{y} = 0)$, vitesse proche de celle de l'observation capteur. La figure 6.8(a) présente le résultat de la phase de prédiction à l'instant k . Comme l'objet n'avait pas été détecté aux instants précédents, aucune zone de forte probabilité d'occupation pouvant concerné cet objet n'est visible. On remarque cependant une zone pour laquelle la probabilité d'occupation est égale à 0,5. En effet, on ne peut prédire si un objet situé en dehors de l'espace représenté dans la grille va entrer à l'intérieur ou non. Les

figures 6.8(b) et 6.8(c) représentent le résultat de l'estimation à l'instant k , en utilisant respectivement la phase de prédiction ou l'uniforme comme connaissance *a priori*.

Sur ces deux estimations, on remarque qu'une zone de forte probabilité d'occupation correspondant à cet objet apparaît. La probabilité est même égale pour cette zone, quelle que soit la connaissance *a priori* utilisée. Ceci tient au fait que l'observation du capteur est située dans une zone pour laquelle la phase de prédiction ne pouvait donner une probabilité d'occupation différente de 0,5. La valeur de la probabilité d'occupation observée sur les phases d'estimation dépend bien sûr des valeurs de probabilité de détection et de fausses alarmes du capteur, comme nous l'avons expliqué dans le chapitre 3.

La figure 6.9 présente un deuxième exemple d'apparition d'objet. Cette fois-ci, le capteur retourne à l'instant k une observation : $(x = 4; y = -3; \dot{x} = -1,5; \dot{y} = 0)$. Cette observation est située dans une zone pour laquelle la phase de prédiction (figure 6.9(a)) donne une probabilité d'occupation faible, puisqu'aucun objet n'y avait été observé aux instants précédents. Cette fois ci, le résultat de l'estimation à l'instant k diffère suivant la connaissance *a priori* utilisée :

- si l'uniforme est utilisée comme connaissance *a priori* (figure 6.9(c)), une zone de forte probabilité d'occupation correspondant à l'observation est visible. La probabilité de cette zone dépend uniquement de la probabilité de détection et de la probabilité de fausse alarme du capteur ;
- si la phase de prédiction est utilisée, on voit apparaître une zone de probabilité légèrement inférieure à 0,5. Cette probabilité dépend également de la probabilité de détection et de la probabilité de fausse alarme du capteur, mais aussi de la phase de prédiction. L'observation du capteur a donc tendance à être considérée comme une fausse alarme. Une nouvelle observation de cet objet à l'instant suivant permettra de trancher définitivement.

Ces deux exemples illustrent un avantage important de l'estimation des grilles d'occupation par rapport aux méthodes classiques. En plus des informations sur les objets présents dans l'environnement, la grille d'occupation fournit des informations sur les zones de l'environnement dans lesquelles aucune observation n'avait eu lieu aux instants précédents.

Comme nous ne gérons pas de pistes, nous n'avons bien sûr pas à décider si l'observation d'un capteur dans une zone jusqu'à présent inoccupée correspond à un objet ou non. L'information fournie par la phase de prédiction prend alors toute son importance :

- si on ne peut rien prédire sur la zone dans laquelle a lieu cette observation, la probabilité d'occupation des cellules correspondantes sera supérieure à 0,5, la valeur dépendant des probabilités de détection et de fausse alarme du capteur dans cette zone ;
- si la zone dans laquelle a lieu cette observation est prédite inoccupée, la probabilité d'occupation des cellules correspondantes sera inférieure à 0,5, la valeur dépendant également des probabilités de détection et de fausse alarme du capteur dans cette zone.

6.3 Fusion de capteurs

Nous avons vu dans la section 6.1.2 que le choix de l'architecture de fusion et de poursuite joue un rôle important lorsque l'on aborde le cas multicapteur. Nous avons également mentionné que dans le cas d'une architecture centralisée, la plupart des adaptations des méthodes d'association de données est directe.

C'est également le cas pour notre méthode. Si la fusion se fait au niveau global, la seule différence avec le cas monocapteur tient au fait que l'on doit tenir compte des observations des différents capteurs. Une variable d'association est définie pour chaque capteur, comme nous l'avons fait dans la section 4.2.

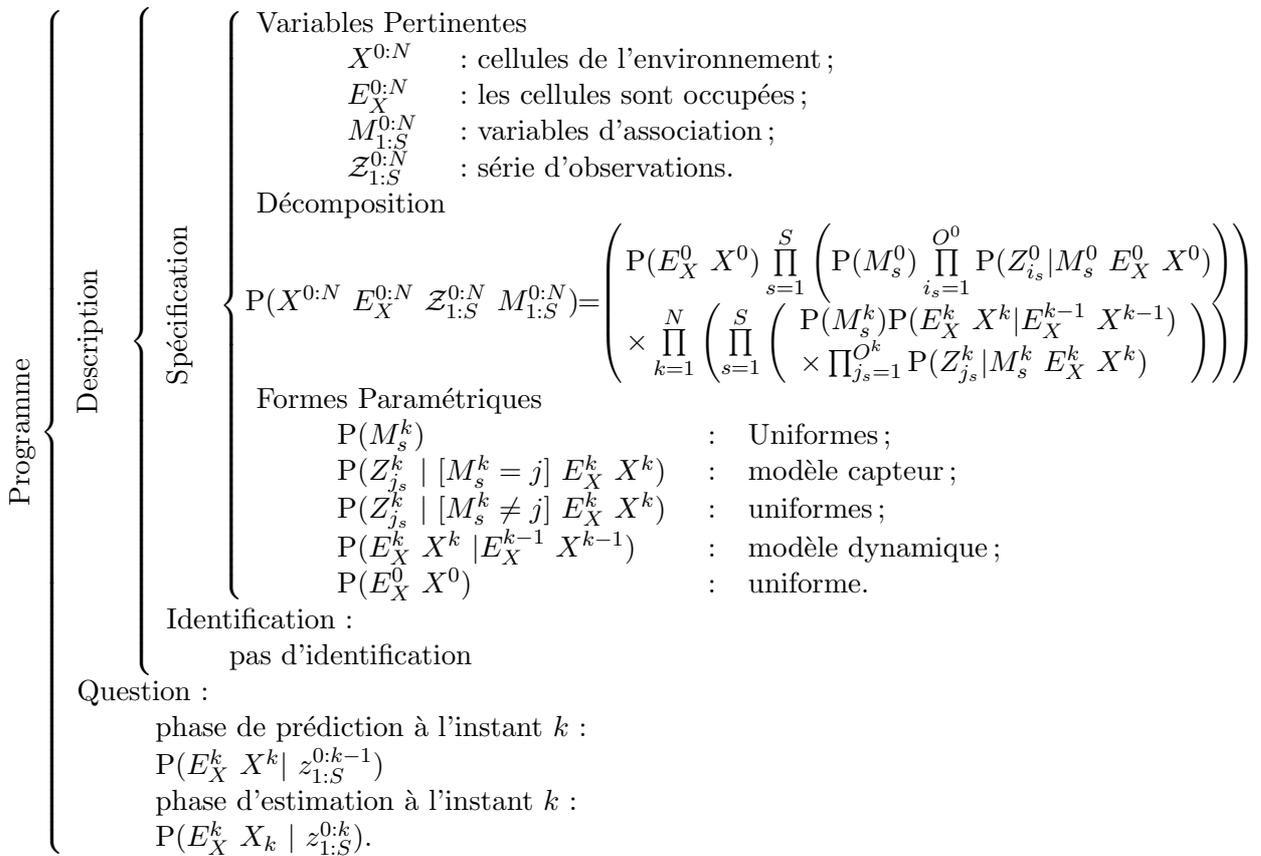


FIG. 6.10: Estimation dynamique des grilles d'occupation dans le cas multicapteur et multiobjet : programme bayésien.

La figure 6.10 présente le programme bayésien correspondant. Nous ne détaillerons pas ce programme.

Nous n'avons pas abordé le cas d'une architecture de fusion distribuée. Dans cette architecture, notre méthode consisterait à estimer une grille d'occupation à l'instant k pour chaque capteur, puis à fusionner ces différentes grilles pour former la grille d'occupation

finale. D'un point de vue théorique, cette fusion ne pose pas de problèmes particuliers si l'on suppose l'indépendance des capteurs [Mor88]. D'un point de vue pratique, il peut se poser un problème de transfert d'information, la quantité de données à transmettre étant très importante.

6.4 Discussion et conclusion sur le filtre d'occupation bayésien

Dans ce chapitre, nous avons présenté la version finale de notre représentation d'environnements dynamiques et encombrés : le filtre d'occupation bayésien.

L'absence de la notion de piste dans cette représentation permet de s'affranchir des difficultés de la phase d'association de données. En particulier, l'expression dans le modèle capteur des connaissances relatives à l'occultation d'une zone de l'environnement par une observation capteur permet un traitement des occultations entre objets sans définition d'une logique supplémentaire. Nous avons vu également que cette forme de représentation utilise toute la connaissance fournie par les observations et le modèle dynamique. En particulier les zones inoccupées de l'environnement, et qui ne peuvent logiquement être occupées rapidement, sont identifiées. Ceci permet de filtrer les fausses alarmes du capteur sans définir de logique supplémentaire ni prendre de décision.

L'utilisabilité d'une méthode de représentation pour la programmation d'un robot ne va pas de soit, et donc l'implantation d'une application, même simple, sur un robot est primordiale dans l'évaluation de cette méthode. C'est le but du chapitre suivant, qui met en jeu le véhicule de démonstration de l'INRIA Rhône-Alpes, le cycab.

Chapitre 7

Application : évitement de collision

Le but de ce chapitre est de montrer l'utilisabilité du filtre d'occupation bayésien, et ce à deux niveaux :

- utilisabilité « grandeur nature » sur un vrai robot, puisque les résultats que nous avons montrés jusqu'à présent provenaient de simulations. En particulier, l'utilisation sur un vrai robot montre que les calculs nécessaires au filtre d'occupation bayésien peuvent être menés en un temps raisonnable ;
- utilisabilité dans le contexte de l'assistance à la conduite automobile, et donc dans des scénarios mettant en jeu des voitures et des piétons par exemple.

Pour l'application, nous avons pour l'instant choisi d'éviter le problème de la fusion de commande que nécessite un système d'assistance à la conduite automobile. Dans cette application, le cycab est donc en « conduite automatique ». Le but est cependant proche de la tâche initiale de l'assistance, puisqu'il s'agit d'éviter les collisions.

Nous commençons par présenter la plateforme expérimentale, c'est à dire le cycab, et les situations dans lesquelles nous allons le faire évoluer. Nous présentons ensuite l'application d'évitement de collision proprement dite.

7.1 Plate-forme expérimentale : le cycab



FIG. 7.1: Le cycab de l'INRIA Rhône-Alpes.

Le cycab (voir figure 7.1) est un véhicule électrique initialement développé par l'INRIA et commercialisé par la société Robosoft¹. Il est actuellement utilisé comme plate-forme de recherche dans des laboratoires français comme l'INRIA ou étrangers, par exemple à l'Université Technologique de Nanyang (NTU) à Singapour.

Le cycab que nous utilisons est équipé d'un télémètre laser Sick LMS-219. Placé à une cinquantaine de centimètres du sol, sur la face avant du Cycab, il balaye un demi-plan horizontal en donnant un couple de mesures tous les demi-degrés : la distance mesurée par le capteur et l'intensité du faisceau réfléchi. L'intérêt de cette dernière donnée vient du fait que la valeur de l'intensité n'est remarquable que sur certains matériaux très réfléchissants de type catadioptré.

Les capacités de perception du cycab sont donc pauvres. En particulier le télémètre laser ne peut fournir d'estimation des vitesses des objets évoluant autour du cycab. Une phase de prétraitement est nécessaire pour estimer ces vitesses avant de construire la grille d'occupation. Ce prétraitement est basé sur un algorithme du plus proche voisin entre deux mesures successives, sans filtrage. Si la mesure des positions est très bonne avec le télémètre, la mesure des vitesses est donc largement moins précise.

Un ordinateur portable puissant (processeur Pentium IV 2 GHz) est installé sur le cycab. Les calculs coûteux sont assurés par celui-ci.

7.2 L'application

Nous avons choisi comme application de notre méthode d'implanter un évitement de collision sur le cycab. Cette application simple a pour mérite d'être pertinente dans le cadre de l'assistance à la conduite automobile. Notre but est donc de contrôler longitudinalement le cycab, afin qu'il évite les objets, piétons ou voitures, qui circulent dans son environnement. Nous avons choisi de ne pas contrôler le cycab en rotation. En effet, les mouvements de rotations sont particulièrement dangereux en automobile, puisqu'ils peuvent engendrer des changements de voies. Ces mouvements sont d'autant plus dangereux avec le cycab, puisque ce robot ne dispose d'aucune information sur les objets qui se trouvent derrière lui, et donc en particulier sur les véhicules qui pourraient être en train de le dépasser.

Dans notre application, le cycab ira donc en ligne droite, et sera contrôlé en vitesse. Le cycab est donc ici en conduite automatique, et nous n'aborderons pas le problème de la fusion de commandes, nécessaire pour l'assistance.

Bien entendu, puisque le cycab ne peut aller qu'en ligne droite, toutes les collisions ne peuvent être évitées. Par exemple, si un véhicule fonce droit sur le cycab, le choc est inéluctable. Cependant, le cycab devra faire en sorte d'avoir une vitesse aussi faible que possible au moment du choc.

Pour l'instant, nous ne disposons pour développer l'évitement de collision que de la grille d'occupation de l'environnement à chaque instant k . Nous ne disposons donc pas d'informations précises sur les différents objets de l'environnement. Le nombre d'objet est lui-même inconnu.

¹<http://www.robosoft.fr>

7.2.1 Clustering de la grille d'occupation

Une première idée peut être de retrouver ces différentes informations à partir de la grille d'occupation avant l'évitement de collision. Il s'agirait donc de trouver les différentes zones de la grille d'occupation ayant une probabilité forte d'être occupées, en considérant que chaque zone correspond à un objet. Il s'agit donc d'un problème de « clustering ». Résoudre ce problème n'est pas simple, puisqu'il faut trouver des zones cohérentes dans notre grille d'occupation. Il s'agit également d'un problème de *décision*, puisque nous aurons alors à décider par exemple d'un seuil de distance pour conclure que deux cellules occupées correspondent au même objet ou non. Selon nous, cette décision arrive trop tôt, et nous perdrons l'avantage de notre méthode, qui était justement le report de cette prise de décision sur l'existence des objets.

Nous allons donc chercher à développer notre évitement de collision sans résoudre ce problème de clustering, et donc sans prendre de décision sur les objets présents dans l'environnement du cycab.

7.2.2 Contrôle du cycab : programme bayésien

Nous détaillons ici le programme bayésien pour le contrôle du Cycab.

7.2.2.1 Variables

Les variables pertinentes ici sont :

- V_k : la vitesse consigne. Le but de ce programme est de calculer cette vitesse ;
- $\mathcal{Z}^{0:k}$: les observations fournies par le Sick depuis le début de l'expérience ;
- V_{k-1} : la vitesse courante du cycab, cette vitesse peut être fournie par les capteurs internes du cycab, ou être estimée par la vitesse consigne calculée à l'instant $k - 1$;
- G^k : la grille d'occupation à l'instant k . Cette variable décrit l'ensemble des grilles d'occupation possible. Puisque chaque cellule de la grille peut prendre deux valeurs (vide ou occupée), cette variable possède 2^N valeurs, N étant le nombre de cellules de la grille.

Ces variables définissent la distribution de probabilité conjointe :

$$P(V^k \mathcal{Z}^{0:k} V^{k-1} G^k). \quad (7.1)$$

7.2.2.2 Décomposition

Dans cette décomposition, nous souhaitons faire apparaître le terme $P(G^k | \mathcal{Z}^{0:k})$. En effet, cette distribution est liée à la probabilité d'occupation de chaque cellule de la grille $P(E_X^k | X^k \mathcal{Z}^{0:k})$. Nous détaillerons ce lien dans la partie « Formes Paramétriques ».

Nous choisissons donc de décomposer la distribution de probabilité conjointe (7.1) sous la forme :

$$P(V^{k-1} \mathcal{Z}^{0:k} V^k G^k) = P(\mathcal{Z}^{0:k})P(V^{k-1})P(G^k | \mathcal{Z}^{0:k})P(V^k | V^{k-1} G^k). \quad (7.2)$$

7.2.2.3 Formes paramétriques

$P(\mathcal{Z}^{0:k})$ représente l'information *a priori* sur les observations du capteur. Les variables de la conjonction $\mathcal{Z}^{0:k}$ ne seront utilisées qu'instanciées, la distribution $P(\mathcal{Z}^{0:k})$ peut donc être quelconque.

La variable V^{k-1} sera également instanciée, la distribution $P(V^{k-1})$ peut donc être quelconque.

La distribution $P(G^k | \mathcal{Z}^{0:k})$ représente la probabilité de la grille d'occupation sachant les différentes observations des capteurs depuis l'instant initial. Nous n'avons pas pour l'instant parlé de distribution de probabilité sur l'espace des grilles d'occupation possibles, mais de probabilité d'occupation pour chaque cellule de la grille. Il nous faut donc exprimer le lien entre ces distributions de probabilité. Une instance g^k de la variable G^k décrit une grille d'occupation possible, c'est à dire que pour chaque cellule X de la grille, elle dit si cette cellule est vide ou occupée. Autrement dit, elle donne à la variable E_X^k sa valeur e_X^k , qui vaut 0 ou 1. La valeur de g^k est donc en réalité la conjonction des valeurs données à l'état d'occupation de chaque cellule de la grille. Puisque l'état d'une cellule est indépendant de l'état des autres cellules (hypothèse de base des grilles d'occupation), nous pouvons écrire :

$$P(g^k | \mathcal{Z}^{0:k}) = \prod_{X^k=1}^N P([E_X^k = e_X^k] | X^k \ \mathcal{Z}^{0:k}). \quad (7.3)$$

Les valeurs de $P([E_X^k = e_X^k] | X^k \ \mathcal{Z}^{0:k})$ pour chaque cellule X^k sont bien sûr fournies par le résultat de la phase d'estimation à l'instant k de notre filtre d'occupation bayésien.

Il nous reste à fournir une forme paramétrique pour les distributions de probabilité $P(V^k | V^{k-1} G^k)$. La spécification de cette forme paramétrique est difficile car très subjective. C'est en effet le programmeur qui en considérant les différentes cellules occupées de la grille G^k estimera le danger lié à la situation de conduite et choisira le comportement du cycab : freinage (et quelle puissance de freinage) ou accélération.

Dans notre implantation, nous avons choisi de passer par l'intermédiaire du « Closest Point of Approach (CPA) ». Si l'on considère deux objets en mouvement, le CPA est une prédiction de la position relative de ces deux objets lorsque la distance qui les sépare est minimale. On considère pour cela les vitesses des deux objets constantes. Nous considérons également le temps qui nous sépare de cette distance minimale, que nous notons TCPA². Les notions de CPA et TCPA sont souvent utilisées en évitement de collision, particulièrement pour la navigation maritime [Fou03]. Par exemple, la société *Raymarine*³ commercialise un système d'évitement de collision basé sur le CPA et le TCPA, appelé *Marpa*. Nous présentons en annexe A la méthode de calcul du CPA et du TCPA.

Pour chaque cellule occupée de la grille G^k , nous calculons donc le CPA et le TCPA associé. Ceci permet d'estimer de manière intuitive le niveau de danger que représente l'objet (supposé) occupant cette cellule :

- si le TCPA est négatif, l'objet s'éloigne du cycab, la situation n'est donc pas dangereuse, nous pouvons choisir d'accélérer dans ce cas ;

²pour *Time to Closest Point of Approach*

³<http://www.raymarine.com>

- sinon plus le TCPA est faible et le CPA proche, plus le niveau de danger est élevé. Il faut également considérer les spécificités des situations de conduite et accorder un certain niveau de confiance aux autres acteurs de la circulation. Par exemple, pour une voiture circulant à contresens du cycab, et donc placée sur la voie de gauche, le CPA est faible (*i.e.* $< 3m$) mais situé sur le coté. Il ne faut bien sûr pas freiner (et surtout pas piler) chaque fois qu'un véhicule croise le cycab.

La forme paramétrique associée à $P(V^k | V^{k-1} G^k)$ est donc spécifiée « à la main » par le programmeur.

7.2.2.4 Question

Le but de ce programme est d'estimer la nouvelle vitesse consigne à appliquer au cycab. Nous posons donc la question :

$$P(V^k | v^{k-1} z^{0:k}) \quad (7.4)$$

Nous n'instancions donc pas la valeur de la grille G^k parmi les variables connues. Cela traduit le fait que nous ne prenons toujours pas de décision sur l'éventuelle présence ou absence d'objet dans l'environnement du cycab. L'inférence sur cette question induit donc une somme sur toutes les configurations possibles de la grille d'occupation :

$$\begin{aligned} P(V^k | v^{k-1} z^{0:k}) &= \frac{\sum_{G^k} P(z^{0:k})P(v^{k-1})P(G^k | z^{0:k})P(V^k | v^{k-1} G^k)}{\sum_{V^k} \sum_{G^k} P(z^{0:k})P(v^{k-1})P(G^k | z^{0:k})P(V^k | v^{k-1} G^k)} \\ &= \frac{\sum_{G^k} P(G^k | z^{0:k})P(V^k | v^{k-1} G^k)}{\sum_{V^k} \sum_{G^k} P(G^k | z^{0:k})P(V^k | v^{k-1} G^k)} \\ &= \frac{1}{\alpha} \sum_{G^k} P(G^k | z^{0:k})P(V^k | v^{k-1} G^k), \end{aligned} \quad (7.5)$$

où α est une constante de normalisation.

Dans cette inférence, les connaissances *a priori* sur V^{k-1} et $Z^{0:k}$ disparaissent puisque ces variables sont instanciées.

La figure 7.2 résume le programme bayésien que nous venons de détailler.

7.2.2.5 En pratique : résolution de l'inférence et prise de décision

Estimer la distribution de probabilité sur V^k nécessite donc une sommation sur l'ensemble des grilles d'occupation possibles, donc sur 2^N , N étant le nombre de cellules de la grille.

Dans le cas de l'application avec le cycab, notre grille comporte environ 64000 cellules. Effectuer une sommation sur 2^{64000} valeurs pour contrôler le cycab est impossible. Nous allons donc devoir calculer une valeur approchée de la distribution de probabilité $P(V^k | v^{k-1} z^{0:k})$.

Les méthodes d'intégration de Monte-Carlo permettent d'estimer la valeur de l'intégrale sur une variable A à partir d'un nombre fini de réalisations de cette variable. Ces réalisations sont choisies par tirage aléatoire sur une distribution de probabilité $P(A)$. Afin d'assurer un

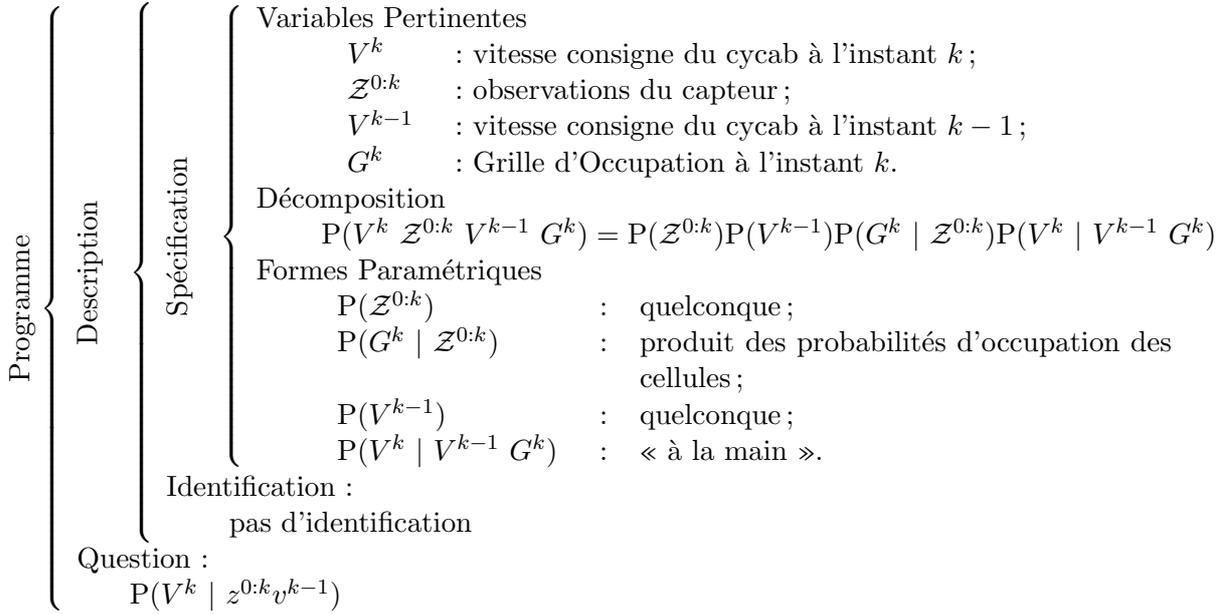


FIG. 7.2: Contrôle du Cycab : programme bayésien.

temps de calcul raisonnable, nous choisissons d'approximer l'intégrale sur G^k en utilisant une unique réalisation de G^k . Afin de minimiser l'erreur, nous choisissons comme réalisation la valeur g^{k*} qui maximise la distribution $P(G^k | z^{0:k})$. Ce maximum est particulièrement facile à calculer. En effet, par suite de l'expression (7.3), nous pouvons écrire :

$$\max_{G^k} P(G^k | z^{0:k}) = \prod_{X=1}^N \max_{E_X^k} P(E_X^k | X^k z^{0:k}). \quad (7.6)$$

La grille d'occupation la plus probable g^{k*} est donc déduite des probabilités d'occupation de chaque cellule :

- si $P([E_x = 1] | x z^{0:k}) > 0.5$, la cellule est considérée comme occupée ;
- si $P([E_x = 1] | x z^{0:k}) \leq 0.5$, la cellule est considérée comme vide.

C'est donc l'obligation d'un temps de calcul raisonnable qui nous oblige à prendre une décision sur l'occupation des cellules de notre grille. Cette décision pourrait être nuancée en considérant plusieurs valeurs de G^k pour calculer l'intégrale. Ceci obligerait à effectuer un tirage aléatoire cellule par cellule pour les considérer vides ou occupées.

En approximant la valeur de l'intégrale à partir de la seule valeur g^{k*} , le résultat de l'inférence (7.5) devient :

$$\begin{aligned} P(V^k | v^{k-1} z^{0:k}) &= \frac{P(g^{k*} | z^{0:k})P(V^k | v^{k-1} g^{k*})}{\sum_{V^k} P(g^{k*} | z^{0:k})P(V^k | v^{k-1} g^{k*})} \\ &= \frac{1}{\alpha} P(V^k | v^{k-1} g^{k*}) \end{aligned} \quad (7.7)$$

Une fois cette distribution de probabilité calculée, il nous reste à choisir la valeur de V^k à envoyer au contrôleur du Cycab. Le plus simple est de choisir la valeur de V^k qui maximise cette distribution de probabilité. En pratique, nous ne calculons donc que cette valeur, et non toute la distribution, ce qui permet de gagner du temps de calcul.

7.2.3 Résultats expérimentaux

7.2.3.1 Détails techniques

Les expériences dont nous présentons maintenant les résultats ont été menées sur le parking de l'INRIA Rhône-Alpes, au milieu des voitures du personnel.

Pour ces expériences, la vitesse du cycab est limitée à 2 m.s^{-1} . La grille comporte 4 dimensions, dans le détail :

- devant le cycab : $x_{min} = 0 \text{ m}$, $x_{max} = 10 \text{ m}$, pas de discrétisation : 50 cm ;
- sur les cotés : $y_{min} = -5 \text{ m}$, $y_{max} = 5 \text{ m}$, pas de discrétisation : 50 cm ;
- vitesse relative longitudinale : $vx_{min} = -3 \text{ m.s}^{-1}$, $vx_{max} = 1 \text{ m.s}^{-1}$, pas de discrétisation : 0.4 m.s^{-1} ;
- vitesse relative latérale : $vy_{min} = -3,2 \text{ m.s}^{-1}$, $vy_{max} = 3,2 \text{ m.s}^{-1}$, pas de discrétisation : 0.4 m.s^{-1} .

Ces plages de variations des variables ont été choisies pour être cohérentes avec la vitesse du cycab et des objets.

L'intégralité de cette grille est observée par le sick monté sur le cycab.

Notre grille comporte environ 64000 cellules, ce qui est déjà important. Grâce aux approximations sur la phase de prédiction de la grille et sur l'estimation de la vitesse consigne, l'ensemble des calculs peut être effectué dans un temps compatible avec le contrôle du cycab, c'est à dire de l'ordre du dixième de seconde.

Le modèle dynamique choisi pour la phase de prédiction est simple, il inclue les mouvements du cycab (simples puisque celui-ci va tout droit), et les mouvements des objets. Pour ces objets, nous considérons que leur vitesse reste quasiment constante entre deux pas de temps.

7.2.3.2 Premier exemple : un piéton imprudent traverse devant le cycab



FIG. 7.3: Un piéton imprudent traverse devant le cycab !

La figure 7.3 montre le scénario de ce premier exemple de contrôle du cycab. La voie est dégagée devant le cycab, qui peut donc accélérer jusqu'à atteindre sa vitesse maximale. Tout à coup un piéton « surgit » de derrière une voiture. Le cycab doit donc freiner afin

d'éviter la collision avec ce piéton. Lorsque celui-ci est passé, le cycab peut à nouveau accélérer.

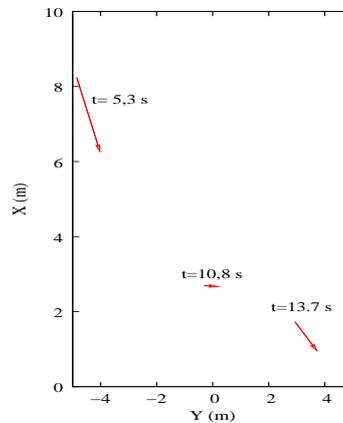


FIG. 7.4: Observations du capteur lors de l'évitement de piéton à trois instants différents.

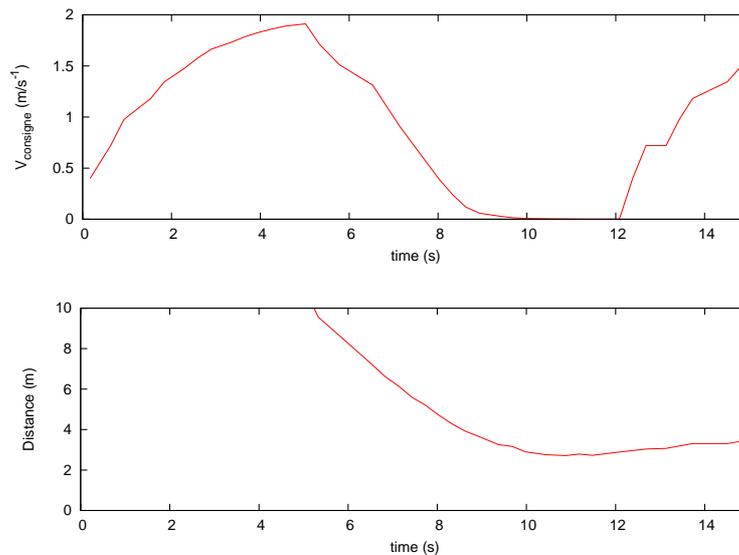


FIG. 7.5: En haut, profil de la vitesse consigne du Cycab lors de l'évitement du piéton. En bas, distance entre le piéton et le Cycab au cours de cette même expérience.

La figure 7.4 montre les observations retournées par le capteur à trois instants différents lors de cette expérience. Ces observations sont représentées sous forme de vecteur, l'origine du vecteur indiquant la position relative de l'objet détecté par rapport au cycab, la direction et longueur du vecteur indiquant la vitesse relative de cet objet par rapport au cycab. À l'instant $t = 5,3$ s, un objet arrivant droit sur le cycab est détecté. Il s'agit bien sûr du piéton. Comme celui-ci marche latéralement devant le cycab et que le cycab avance, la vitesse relative du piéton par rapport au cycab est diagonale. Le choc semble proche, donc

le cycab freine. À l'instant $t = 10,3 \text{ s}$, le piéton marche toujours devant le cycab, mais celui-ci s'est arrêté. La vitesse relative du piéton par rapport au cycab est donc horizontale. Enfin à $t = 13,7 \text{ s}$, le piéton est toujours devant le cycab, mais à droite de celui-ci. Le cycab a redémarré, la vitesse relative du piéton par rapport au cycab est donc à nouveau diagonale.

La figure 7.5 montre l'évolution parallèle de la vitesse du cycab (en haut) et de la distance relative piéton/cycab (en bas) pendant cette expérience. Le piéton est détecté pour la première fois à $t = 5,2 \text{ s}$. Avant cet instant, le cycab accélérât jusqu'à une vitesse proche de 2 m.s^{-1} . Compte tenu du danger que représente le piéton, il se met à freiner. La distance piéton/cycab diminue lors de ce freinage, jusqu'à environ 2 m lorsque le cycab s'arrête ($t = 9,6 \text{ s}$). Lorsque le piéton est passé devant le cycab, celui-ci ré-accélère ($t = 12 \text{ s}$). La distance piéton/cycab reste quasiment constante, mais le piéton ne représente plus un danger puisque la vitesse relative indique que le piéton s'éloigne du cycab.

7.2.3.3 Suivi d'un véhicule

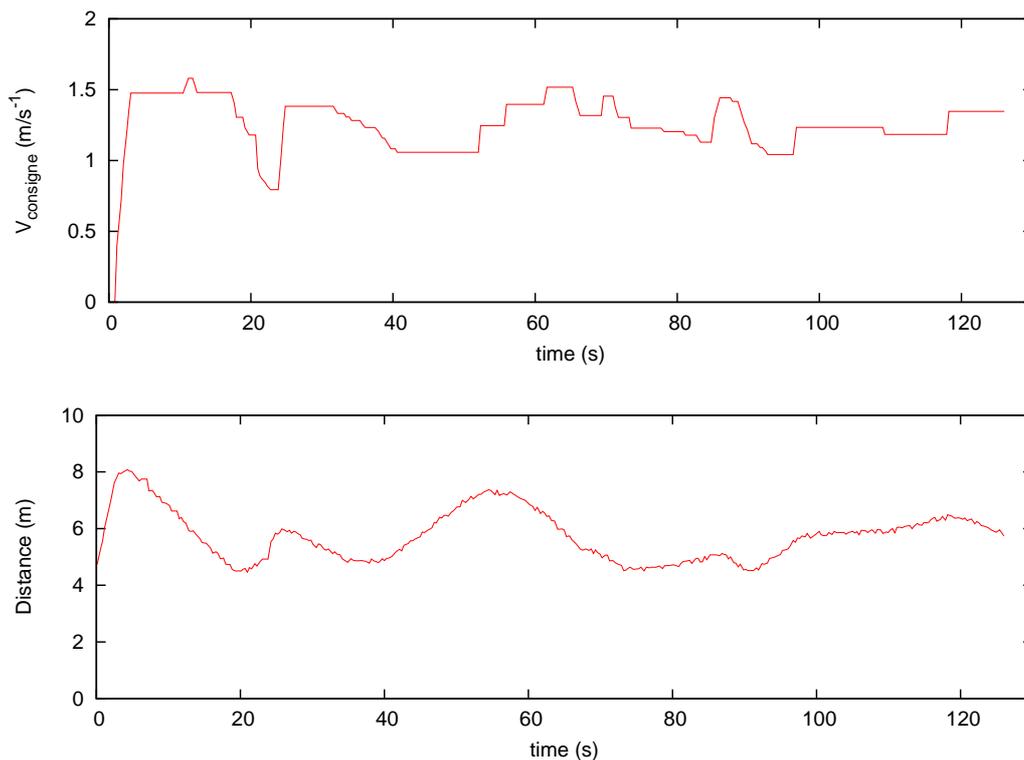


FIG. 7.6: En haut, profil de la vitesse consigne du Cycab lors d'un suivi de véhicule. En bas, distance entre le véhicule et le Cycab au cours de cette même expérience.

Pour ce deuxième exemple, nous voulons illustrer les capacités du cycab à rester à distance de sécurité d'un véhicule situé devant lui. Nous rappelons que pour cette expérience, le cycab est en conduite automatique, et ne peut aller que tout droit, puisque le seul

contrôle est longitudinal.

Au début de cette expérience, le cycab est arrêté derrière un véhicule, à 5 mètres de celui-ci. Le véhicule précédant le cycab démarre pour atteindre une vitesse d'environ $1,5 \text{ m.s}^{-1}$. Le cycab démarre également et reste derrière le véhicule tout au long de l'expérience.

La figure 7.6 montre l'évolution parallèle de la vitesse du cycab (en haut) et de la distance relative véhicule/cycab (en bas) au cours de cette expérience de suivi. On remarque sur cette figure une certaine instabilité sur la vitesse du cycab et sur la distance véhicule/cycab. Cette instabilité est due à notre méthode de calcul de V^k (et particulièrement à la définition de la forme paramétrique associée à $P(V^k | V^{k-1} G^k)$), qui est un peu trop simpliste pour une tâche de suivi pure et l'asservissement de la vitesse du cycab sur la vitesse du véhicule le précédant. Cependant cette instabilité reste raisonnable, même si le confort du passager ne s'en trouve pas amélioré.

Malgré cette instabilité, le comportement du cycab est satisfaisant du point de vue de l'application. En effet, la distance entre le cycab et le véhicule n'est jamais inférieure à quatre mètres, ce qui est raisonnable pour une vitesse maximale du cycab égale à $1,5 \text{ m.s}^{-1}$.

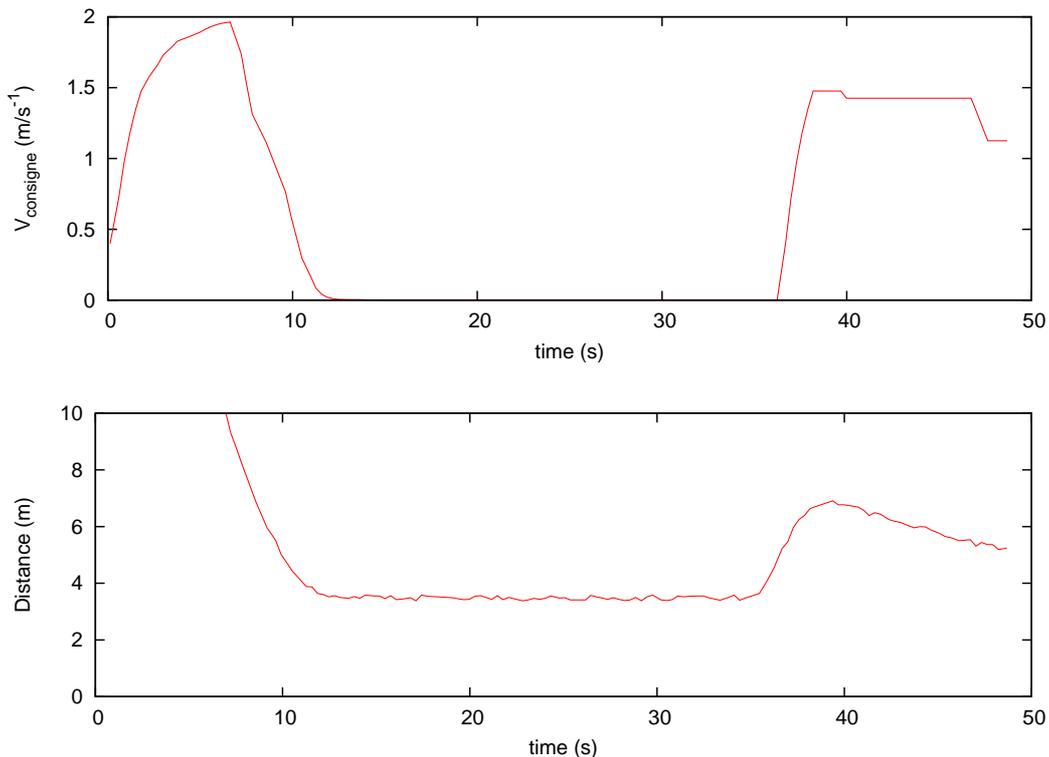


FIG. 7.7: En haut, profil de la vitesse consigne du Cycab lors d'un Stop & Go. En bas, distance entre le véhicule et le Cycab au cours de cette même expérience.

7.2.3.4 « Stop and Go »

Le « Stop and Go » est une aptitude importante pour la conduite en ville et est souvent citée en application possible de l'assistance à la conduite automobile. Le but de cette application est l'arrêt automatique du véhicule derrière un véhicule à l'arrêt, par exemple dans un embouteillage ou à un feu, et le redémarrage automatique lorsque le véhicule situé devant redémarre lui-même. Nous voulons montrer ici que ce genre de comportement peut être réalisé sur les bases de nos grilles d'occupation et de notre contrôle simpliste du cycab.

Au début de cette expérience, la voie est dégagée devant le cycab, qui peut donc accélérer et atteindre sa vitesse de croisière de 2 m.s^{-1} . Quelques dizaines de mètres devant lui se trouve un véhicule à l'arrêt. Le cycab stoppe quelques mètres derrière ce véhicule. Après quelques secondes, celui-ci redémarre et atteint une vitesse de $1,5 \text{ m.s}^{-1}$. Le cycab redémarre à sa suite.

La figure 7.7 montre l'évolution parallèle de la vitesse du cycab (en haut) et de la distance relative véhicule/cycab (en bas) au cours de cette expérience de Stop and Go. Lorsque le véhicule à l'arrêt est détecté (environ $t = 8 \text{ s}$), le cycab freine jusqu'à atteindre une vitesse nulle. Il se trouve alors environ 4 mètres derrière le véhicule arrêté. À $t = 35 \text{ s}$, le véhicule arrêté redémarre. La distance entre le véhicule et l'objet augmente, donc le cycab peut lui même redémarrer. On remarque que le cycab démarre moins violemment que le véhicule situé devant lui, puisque la distance continue d'augmenter quelques secondes avant d'atteindre un maximum puis de diminuer.

7.2.3.5 Occultation

Dans cette exemple, nous faisons intervenir une occultation entre les objets pour comparer l'estimation statique de la grille d'occupation et le filtre d'occupation bayésien.

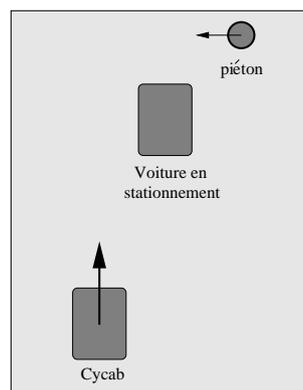
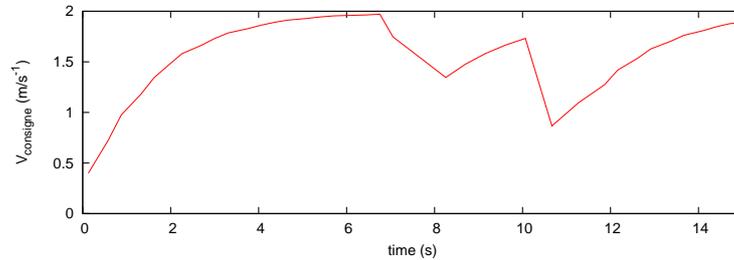


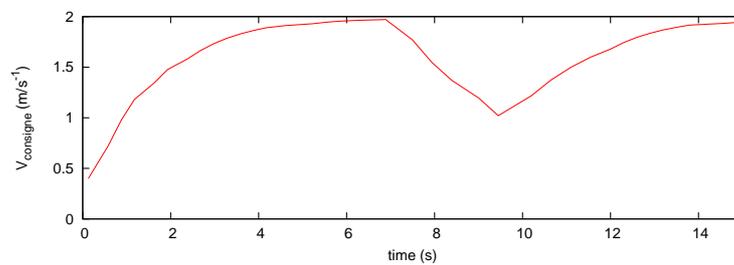
FIG. 7.8: Exemple de situation incluant une occultation : description du scénario.

La figure 7.8 montre le scénario de cet exemple. Le cycab avance en ligne droite à une vitesse de 2 m.s^{-1} . Un piéton arrive par la droite vers le cycab. Visible au début de l'expérimentation, ce piéton est temporairement masqué par une voiture garée le long de

la voie. Le risque est donc que le piéton ressurgisse de derrière cette voiture juste devant le cycab.



(a) Estimation statique des grilles.



(b) Filtre d'occupation.

FIG. 7.9: Profils de vitesse du cycab lors de l'évitement de piéton incluant une occultation. En haut, résultat obtenu avec une estimation statique de la grille. En bas, résultat obtenu avec le filtre d'occupation.

La figure 7.9(a) présente le profil de vitesse du cycab lorsque l'estimation de la grille d'occupation est faite de manière statique, c'est à dire sans tenir compte de l'historique des observations. La première observation du piéton a lieu pour $t = 7 s$. Le cycab commence alors à ralentir. Entre les instants $t = 8.2 s$ et $t = 10 s$, le piéton est masqué par la voiture, il n'est donc pas détecté par le capteur. Le cycab ré-accélère. À $t = 10 s$, le piéton débouche devant le cycab de derrière la voiture, Le cycab freine alors fortement pour laisser passer le piéton. Une fois celui-ci passé, il peut ré-accélérer.

La figure 7.9(b) présente le profil de vitesse du cycab pour le même scénario, l'estimation de la grille d'occupation incluant cette fois-ci le filtrage. L'historique des observations capteurs est donc pris en compte. Jusqu'à l'instant $t = 8.2s$, ce profil est similaire à celui de la figure 7.9(a) : quand le piéton est détecté, le cycab ralentit. Entre les instants $t = 8.2 s$ et $t = 10 s$, le piéton est également masqué par la voiture. Cette fois-ci, grâce au filtrage, le cycab continue à freiner, une zone dangereuse de forte probabilité d'occupation étant toujours présente dans la grille. Quand l'objet est à nouveau détecté, il n'est plus dangereux, car loin devant avec une vitesse latérale suffisante pour passer devant le cycab. Celui-ci peut donc ré-accélérer.

Dans ce deuxième exemple, le comportement du cycab est à la fois plus sûr et plus confortable pour ces passagers. Plus sûr car le passage du piéton devant le véhicule étant anticipé, la distance qui sépare le cycab du piéton quand celui-ci débouche de derrière la voiture est supérieure. Il est également plus confortable car le freinage est plus doux que dans le cas d'une estimation statique des grilles. De plus, la petite phase d'accélération entre les instants $t = 8$ s et $t = 10$ s dans ce cas, accélération suivie d'un fort freinage, est particulièrement inconfortable.

7.3 Conclusion sur l'évitement de collision

Nous avons dans ce chapitre présenté une application d'évitement de collision basée sur notre méthode de représentation d'environnements dynamiques et encombrés : le filtre bayésien d'occupation. Cet évitement de collision a été implanté sur le véhicule de test de l'INRIA Rhône-Alpes, le cycab.

Le point remarquable de cet algorithme d'évitement de collision tient au fait qu'en théorie, aucune décision n'a à être prise avant le choix de la commande à envoyer au robot. En pratique, il est cependant nécessaire de travailler avec la grille la plus probable pour assurer un temps de calcul raisonnable de la commande. Cette forme de décision pourrait cependant être nuancée en travaillant non pas avec la grille la plus probable, mais avec un petit nombre de grilles tirées aléatoirement.

Par rapport à notre but initial, les résultats présentés dans ce chapitre sont satisfaisants. Nous avons montré qu'il est possible d'éviter des objets en mouvements (les piétons), ceux-ci pouvant être cachés temporairement à la vue du cycab. Nous avons également montré qu'il est possible de maintenir une distance de sécurité par rapport à un véhicule alors même que la notion d'objet est totalement absente de notre représentation de l'environnement.

L'utilisabilité du filtre d'occupation bayésien dans le cadre d'applications d'assistance à la conduite automatique est donc prouvée dans ce chapitre.

D'un point de vue purement « contrôle », cette application peut paraître peu satisfaisante. En effet, nous avons pu constater que les profils de vitesse du cycab ne sont pas très lisses, particulièrement dans le cas de suivi de véhicule. Ceci n'était pas notre but. En effet, nous avons présenté ici une application de conduite automatique, mais notre but à terme est l'assistance à la conduite. Dans notre vision personnelle de l'assistance, le système d'assistance ne doit pas prendre le contrôle total du véhicule, sauf peut être en cas de danger immédiat. Il doit se cantonner à rendre plus difficile certaines actions au conducteur. De cette manière, si le conducteur prend des décisions dangereuses, il le fait consciemment, et non par inattention. Dans cette optique, la programmation proscriptive [KPBM03] paraît être une approche très intéressante. Dans ce cadre, le but d'un programme n'est pas de calculer la vitesse à appliquer au cycab pour obtenir un comportement sûr, mais d'interdire les commandes pour lesquelles le danger est trop important.

Chapitre 8

Perspectives et conclusion

8.1 Conclusion

Nous avons présenté dans ce document une représentation d'environnements dynamiques et encombrés, le cadre d'application étant le design d'applications d'assistance à la conduite automobile en milieu urbain. L'originalité de cette méthode tient au fait qu'elle ne prend aucune décision intermédiaire, contrairement aux méthodes basées sur une poursuite multipiste. Elle ne pâtit donc pas des pertes d'information dues à ces décisions et contourne ainsi une des difficultés principale : l'appariement piste-observation.

Cette méthode est basée d'une part sur les grilles d'occupation, qui permettent de s'abstraire de la notion d'objet, et d'autre part sur les filtres bayésiens, qui permettent de prendre en compte la dynamique de la scène. Nous avons appelé le rapprochement de ces deux méthodes le *filtre d'occupation bayésien*.

De nombreux résultats expérimentaux ont permis d'illustrer que cette méthode de représentation de l'environnement est extrêmement riche. En effet, elle fournit de l'information non seulement sur les zones occupées de l'environnement, mais également sur les zones inoccupées et sur les zones cachées de l'environnement. Ceci permet notamment de gérer naturellement les occultations entre objets et les défaillances des capteurs, que ce soient des fausses alarmes ou des détections manquées.

Enfin, nous avons tenu à présenter une application d'évitement de collision basée sur le filtre d'occupation bayésien. Cette application est implantée sur le véhicule de démonstration de l'INRIA Rhône-Alpes, le cycab. Même si cette application est encore loin d'un système commercialisable d'assistance à la conduite automobile, elle permet d'illustrer la faisabilité et la pertinence de notre méthode. Faisabilité : alors que le coût calculatoire de notre méthode est à première vue énorme, le développement d'un algorithme d'approximation spécifique permet toutefois de contrôler le cycab en temps réel. Pertinence : le fait qu'un humain puisse interpréter une modélisation de l'environnement ne signifie pas que l'on puisse programmer un robot en s'appuyant sur cette modélisation.

8.2 Perspectives

grille (min ; max ; pas)	X (m)	0 ; 10 ; 0,5	0 ; 20 ; 0,5	0 ; 20 ; 0,5
	Y (m)	-5 ; 5 ; 0,5	-10 ; 10 ; 0,5	-10 ; 10 ; 0,5
	VX (m/s)	-3 ; 1 ; 0,4	-3 ; 1 ; 0,4	-6 ; 2 ; 0,4
	VY (m/s)	-3,2 ; 3,2 ; 0,4	-3,2 ; 3,2 ; 0,4	-6,4 ; 6,4 ; 0,4
nombre de cellules		64000	256000	1024000
temps de calcul (ms)		100	600	2400

TAB. 8.1 – Evolution du nombre de cellules dans la grille et du temps de calcul en fonction de l'espace représenté par la grille. Il suffit de doubler la taille de la grille selon chaque dimension pour que le nombre de cellules explose et que le temps de calcul devienne incompatible avec le contrôle d'un robot.

Le développement à considérer en premier lieu suite à ces travaux est bien entendu celui d'une « vraie » application d'assistance à la conduite, qui pourrait prendre la forme d'un évitement de collision. Pour ce développement, plusieurs problèmes sont à étudier :

- tout d'abord, la vitesse d'un véhicule automobile même en zone urbaine est supérieure à celle du cycab dans notre application. Afin d'opérer en toute sécurité, nous aurons donc besoin de connaître une zone plus grande de l'environnement, particulièrement devant le véhicule, et donc d'agrandir la grille d'occupation. Cette explosion du nombre de cellules va entraîner une explosion des besoins en puissance de calculs (voir Table 8.1). Afin de palier cette difficulté, il sera nécessaire de développer des algorithmes d'approximation très efficaces. Une idée pour limiter le nombre de cellules peut également être de travailler avec des grilles logarithmiques, ou de manière plus générale avec des grilles *multi-échelle*, en utilisant par exemple des techniques de quadtree ou d'ondelettes ;
- pour une application d'assistance, il est nécessaire de fusionner les ordres du conducteur et les ordres du système d'assistance. Nous l'avons déjà mentionné en conclusion du chapitre 7, la programmation proscriptive semble être une piste très intéressante ;
- l'utilisation d'un sick seul pour observer l'environnement est insuffisant. En particulier, le sick est un capteur plan. Il est donc incapable de détecter une moto couchée en travers de la route s'il est placé trop haut, ce qui peut être fort dommageable. Il est donc nécessaire d'augmenter les capacités de perception du véhicule. Le choix des capteurs doit être fait selon deux critères principaux : leur pertinence vis à vis de l'application d'assistance considérée et leur coût. Dans cette optique, les capteurs de vision semblent particulièrement adaptés, seuls [CCJM01] [BBFL] ou couplés avec des radars [Ste01]. La fusion des informations provenant de ces différents capteurs ne pose pas de problème théorique pour notre filtre d'occupation.

Bibliographie

- [AG92] M. A. Abidi and R. C. Gonzalez. *Data Fusion in Robotics and Machine Intelligence*. Academic Press, 1992.
- [AHM02] D. Arbuckle, A. Howard, and M. M. Matarić. Temporal occupancy grids : a method for classifying the spatio-temporal properties of the environment. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, (CH), Octobre 2002.
- [AM00] S.M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2) :325–343, 2000.
- [AMGC02] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filter for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 2002.
- [ATS01] K.O. Arras, N. Tomatis, and R. Siegwart. Multisensor on-the-fly localization : precision and reliability for applications. *Robotics and Autonomous Systems*, 44 :131–143, 2001.
- [BBC⁺95] J. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot Rhino. *Artificial Intelligence Magazine*, 16(1), 1995.
- [BBFL] M. Bertozzi, A. Broggi, A. Fascioli, and P. Lombardi. Vision-based pedestrian detection : will ants help ?
- [BBS96] W.D. Blair and Y. Bar-Shalom. Tracking maneuvering targets with multiple sensors : Does more data always mean better estimates? *IEEE Transactions on Aerospace and Electronic Systems*, 32(1), January 1996.
- [BBW95] A.L. Barker, D.E. Brown, and Martin W.N. Bayesian estimation and the Kalman filter. *Computers and Mathematics with Applications*, 30(10), 1995.
- [BDL⁺98a] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer, and K. Mekhnacha. Interprétation vs. description I : Proposition pour une théorie probabiliste des systèmes cognitifs sensori-moteurs. *Intellectica*, 26-27 :257–311, 1998.
- [BDL⁺98b] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer, and K. Mekhnacha. Interprétation vs. description II : Fondements mathématiques. *Intellectica*, 26-27 :313–336, 1998.

- [Bil98] J.A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-021, University of Berkeley, 1998.
- [BP00] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 2000.
- [BSF88] Y. Bar-Shalom and T.E. Fortman. *Tracking and Data Association*. Academic Press, 1988.
- [BSL95] Y. Bar-Shalom and X. Li. *Multitarget Multisensor Tracking : Principles and Techniques*. YBS Publishing, 1995.
- [BST74] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. In *Proceedings of the 4th Symposium on nonlinear estimation theory and its applications*, 1974.
- [BST75] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11, 1975.
- [CB01] C. Coué and P. Bessière. Chasing an elusive target with a mobile robot. In *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, Hawai (HI), 2001.
- [CCJM01] X. Clady, F. Collange, F. Jurie, and P. Martinet. Object tracking with a pan-tilt-zoom camera : application to car driving assistance. In *Proceedings of IEEE International Conference on Robotics and Automation*, Seoul (Korea), mai 2001.
- [Col02] F.. Colas. Modélisation analytique et bayésienne des neurones d'orientation de la tête. Rapport de DEA, Institut National Polytechnique de Grenoble (INPG), Grenoble, (FR), Juin 2002.
- [Coo90] G. Cooper. The computational complexity of probabilistic inference using bayesian belief network. *Artificial Intelligence*, 42(2-3), 1990.
- [Cox46] R.T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14, 1946.
- [Cox61] R.T. Cox. *The Algebra of Probable Inference*. The John Hopkins Press, Baltimore, 1961.
- [Cro89] J. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Scottsdale (Az), mai 1989.
- [DGA00] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3), 2000.
- [Dia99] J. Diard. Apprentissage hiérarchique bayésien. Rapport de DEA, Université Joseph Fourier, Grenoble, France, Juin 1999.
- [Dia03] Julien Diard. *La carte bayséienne – Un modèle probabiliste hiérarchique pour la navigation en robotique mobile*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France, Janvier 2003.

- [DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39 :1–38, 1977.
- [Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer, Special Issue on Autonomous Intelligent Machines*, pages 46–57, Juin 1989.
- [Elf92] A. Elfes. Multi-source spatial data fusion using bayesian theory. In M. A. Abidi and R. C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*. Academic Press, 1992.
- [FBSS83] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE-8, 1983.
- [Fou03] S. Fournier. Détection de collision et d'échouage dans la plate-forme TRANS. GTAS'03 10èmes journées du groupe de travail Animation et Simulation, Brest, Juin 2003.
- [Fox01] D. Fox. KLD-sampling : Adaptive particle filters. In MIT Press, editor, *Advances in Neural Information Processing Systems 14*, 2001.
- [Gew96] J. Geweke. Monte Carlo simulation and numerical integration. In H. Amman, D. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 13, pages 731–800. Elsevier North-Holland, Amsterdam, 1996.
- [GLJ97] H. Gauvrit, J.P. Le Cadre, and C Jauffret. A formulation of multitarget tracking as an incomplete data problem. *IEEE Trans. on Aerospace and Electronic Systems*, 33(4), 1997.
- [Gor97] N. Gordon. A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1), 1997.
- [GR03] R. García-Ramírez. *Programmation Bayésienne des Bras Manipulateurs*. Thèse de doctorat, Institut National Polytechnique de Grenoble (INPG), Grenoble (FR), Mai 2003.
- [HBS89] A. Houles and Y. Bar-Shalom. Multisensor tracking of a manoeuvring target in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 25, 1989.
- [HL97] D. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1), January 1997.
- [HLCP02] C. Hue, J-P. Le Cadre, and P. Pérez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50(2), 2002.
- [HS98] J. W. Harris and H. Stocker. *Handbook of Mathematics and Computational Science*. Springer, 1998.
- [Hue03] C. Hue. *Méthodes Séquentielles de Mont Carlo pour le filtrage non linéaire multi-objets dans un environnement bruité. Applications au pistage multi-cibles*

- et à la trajectographie d'entités dans des séquences d'images 2D*. PhD thesis, Université de Rennes I, Janvier 2003.
- [Jay03] E. T. Jaynes. *Probability Theory : the Logic of Science*. Cambridge University Press, 2003.
- [Jaz70] A. H. Jazwinsky. *Stochastic Processes and Filtering Theory*. New York : Academic Press, 1970.
- [JUDW95] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc of the American Control Conference*, 1995.
- [JUDW00] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new method for the transformation of means and covariances in filters and estimators. *IEEE Trans. on Automatic Control*, 45(3), 2000.
- [Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 35, Mars 1960.
- [KL88] T. Kurien and M.E. Liggins. Report-to-track assignment in multisensor multitarget tracking. In *27th Conference on Decision and Control*, décembre 1988.
- [KPBM03] C. Koike, C. Pradalier, P. Bessière, and M. Mazer. Obstacle avoidance and proscriptive bayesian programming. In *Workshop on Reasoning with Uncertainty in Robotics, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco (ME), Août 2003.
- [LBDM04] O. Lebeltel, P. Bessière, J. Diard, and E. Mazer. Bayesian robot programming. *Autonomous Robots*, 16 :49–79, 2004.
- [LBS96] R. Li and Y. Bar-Shalom. Tracking clutter with nearest neighbor filters : Analysis and performance. *IEEE Transactions on Aerospace and Electronic Systems*, 32(3), 1996.
- [Leb99] O. Lebeltel. *Programmation Bayésienne des Robots*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France, Septembre 1999.
- [Mar47] J.I. Marcum. A statistical theory of targets detection by pulsed radar. Research Memorandum RM-745, The Rand Corporation, Santa-Monica, CA, 1947. repris sans les details dans blackman.
- [MD03] A. Mittal and L.S. Davis. M₂ Tracker : a multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3), 2003.
- [Mek99] K. Mekhnacha. *Méthodes probabilistes bayésiennes pour la prise en compte des incertitudes géométriques : application à la CAO-robotique*. Thèse de doctorat, Institut National Polytechnique de Grenoble (INPG), Grenoble (FR), juillet 1999.
- [MES94] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Trans. on Automatic Control*, 39(6), June 1994.

- [MMB01] K. Mekhnacha, E. Mazer, and P. Bessière. The design and implementation of a bayesian CAD modeler for robotic applications. *Advanced Robotics*, 15(1) :45–70, 2001.
- [MMDB00] G. Marsden, M. Mc Donald, and M. BrackStone. Toward an understanding of adaptive cruise control. *Journal of Transportation Research Part C : Emerging Technologies*, 9(5), juin 200.
- [Mor88] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2) :61–74, 1988.
- [MR87] G. L. Miller and Wagner E. R. An optical rangefinder for autonomous robot cart navigation. In *Proc. of the 1987 SPIE/IECON*, 1987.
- [Nea93] R. M. Neal. Probabilistic inference using Markov Chain Monte Carlo methods. Research Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.
- [OF02] M. Orton and W. Fitzgerald. A bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, 50(2), 2002.
- [PO93] L.Y. Pao and S.D. O’Neil. Multisensor fusion algorithm for tracking. In *Proceedings of the American Control Conference*, Juin 1993.
- [Ras02] R. Raspail. Apprentissage bayésien par imitation. Rapport de DEA, Université Joseph Fourier, Grenoble, France, Juin 2002.
- [SBC99] L.D. Stone, C.A. Barlow, and T.L Corwin. *Bayesian Multiple Target Tracking*. Artech House, 1999.
- [SBFC01] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving target with a mobile robot using particle filters and statistical data association. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Hawaii, HI (US), oct-nov 2001.
- [Sea71] R.G. Sea. An efficient suboptimal decision procedure for associating sensor data with stored tracks in real-time surveillance systems. In *Proceedings of the Conference on Decision and Control*, Miami Beach (FL), décembre 1971.
- [Sha93] E. Shahbazian. Implementation strategies for central-level multi-hypothesis tracking fusion with multiple dissimilar sensors. In *Signal Processing, Sensor Fusion, and Target Recognition II, Proc SPIE*, 1993. demandé à la doc le 06-06-03.
- [Sin73] R.G. Singer, R.A. and Sea. New results in oimizing surveillance system tracking and data correlation performance in dense multitarget environments. *IEEE Transactions on Automatic Control*, 18 :571–582, décembre 1973.
- [SL94] R.L. Streit and T.E. Luginbuhl. Maximum likelihood methof for probabilistic multi-hypothesis tracking. In *SPIE International Symposium, Signal and Data Processing of Small Targets*, volume 2235, Orlando (FL), Avril 1994.

- [SL95] R.L. Streit and T.E. Luginbuhl. Probabilistic multi-hypothesis tracking. Technical Report 10,428, Naval Undersea Warfare Center Division Newport, 1995.
- [Ste01] B. Steux. *RT Maps, un environnement logiciel dédié à la conception d'applications embarquées temps-réel. Utilisation pour la détection automatique de véhicules par fusion radar/vision*. Thèse de doctorat, École des Mines de Paris, 2001.
- [Swe54] P. Swerling. Probability of detection for fluctuating targets. Research Memorandum RM-1217, The Rand Corporation, Santa-Monica, CA, 1954. repris sans les détails dans blackman.
- [TBB⁺00] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotic Research*, 19(11), 2000.
- [Thr98] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1), 1998.
- [Thr01] S. Thrun. Learning occupancy grids with forward models. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Hawaii, HI (US), oct-nov 2001.
- [Thr02] S. Thrun. Robotic mapping : A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [WB] G. Welch and G. Bishop. An introduction to the Kalman filter. available at <http://www.cs.unc.edu/~welch/kalman/index.html>.
- [WDH⁺00] G.R. Widmann, M.K. Daniels, L. Hamilton, L. Humm, B. Riley, D.E. Schiffman, J.K. an Schnelker, and W. Wishon. Comparison of lidar-based and radar-based adaptive cruise control systems. SAE Technical paper series 2000-01-0345, SAE, 2000.
- [YL97] B. Yamauchi and P. Langley. Place recognition in dynamic environments. *Journal of Robotic System*, 14(2), 1997.

Annexe A

Closest Point of Approach

Le « Closest Point of Approach » désigne la position pour laquelle deux objets en mouvement sont à distance minimale. Ce point est souvent utilisé pour l'évitement de collision.

Nous considérons que les objets se déplacent à vitesse constante, en direction et en norme. Les objets se déplacent donc le long de droites. Cependant, la distance minimale entre les deux objets n'est pas égale à la distance minimale entre les deux droites support, puisque la distance entre les points doit être calculée à un instant t identique pour les deux objets. Ainsi en dimension 2, même si les deux droites support se croisent, les objets peuvent rester loin l'un de l'autre.

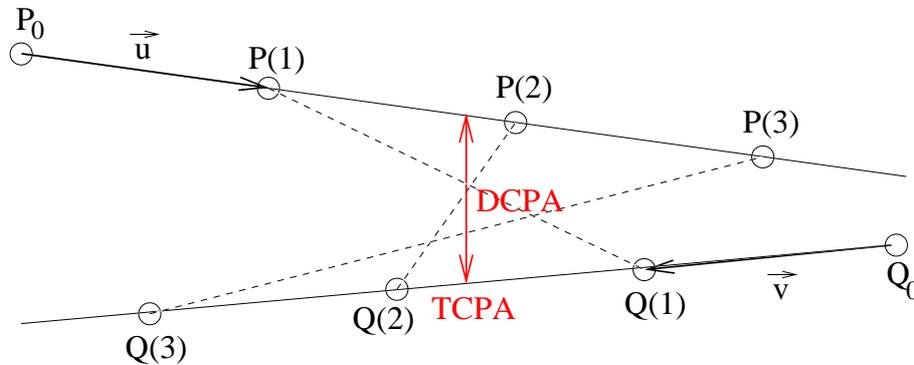


FIG. A.1: CPA pour deux objets mobiles dans un plan à vitesse constante.

Considérons deux points mobiles dont les positions au temps t sont données respectivement par $P(t)$ et $Q(t)$. Les positions à l'instant initial sont notées P_0 et Q_0 . Les vecteurs vitesses, supposés constants au cours du temps, des deux objets sont \vec{u} et \vec{v} . La figure A.1 résume cette situation. Les positions des objets au cours du temps sont données par les équations :

$$P(t) = P_0 + t\vec{u} ; \quad (\text{A.1})$$

$$Q(t) = Q_0 + t\vec{v}. \quad (\text{A.2})$$

De ces deux équations, on déduit la distance entre les deux objets à l'instant t :

$$d(t) = |P(t) - Q(t)| \quad (\text{A.3})$$

$$= |\vec{d}_0 + t(\vec{u} - \vec{v})|, \quad (\text{A.4})$$

avec $\vec{d}_0 = P_0 - Q_0$.

Pour estimer la distance minimale entre les objets, on cherche à minimiser le carré de cette distance, que nous notons $D(t)$:

$$D(t) = d(t)^2 \quad (\text{A.5})$$

$$= (\vec{u} - \vec{v}) \cdot (\vec{u} - \vec{v})t^2 + 2\vec{d}_0 \cdot (\vec{u} - \vec{v})t + d_0^2. \quad (\text{A.6})$$

$D(t)$ est minimale quand sa dérivée par rapport au temps s'annule :

$$D'(t_{CPA}) = 2(\vec{u} - \vec{v}) \cdot (\vec{u} - \vec{v})t_{CPA} + 2\vec{d}_0 \cdot (\vec{u} - \vec{v}) \quad (\text{A.7})$$

$$= 0. \quad (\text{A.8})$$

Ce qui donne :

$$t_{CPA} = \frac{-\vec{d}_0 \cdot (\vec{u} - \vec{v})}{|\vec{u} - \vec{v}|^2}, \quad (\text{A.9})$$

si $|\vec{u} - \vec{v}| \neq 0$. Si $|\vec{u} - \vec{v}| = 0$, les objets se déplacent à même vitesse suivant des droites parallèles. La distance entre les objets est donc constante, et on peut fixer $t_{CPA} = 0$.

Si $t_{CPA} < 0$, le CPA a déjà eu lieu dans le passé, et les deux objets s'éloignent l'un de l'autre.

La distance minimale est bien sûr donnée par :

$$d_{CPA} = d(t_{CPA}). \quad (\text{A.10})$$