

Par **Julien MORAS**

Grilles de perception évidentielles pour la navigation robotique en milieu urbain

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 17 janvier 2013
Spécialité : Technologies de l'Information et des Systèmes

D2057

Thèse de Doctorat de l'Université de Technologie
de Compiègne.

Champ disciplinaire : Technologies de
l'Information et des Systèmes

Grilles de perception évidentielles pour la navigation robotique en milieu urbain

Julien Moras

Soutenue publiquement le 17 janvier 2013
à l'Université de Technologie de Compiègne

Devant le jury composé de :

Rapporteurs	Paul Checchin	Institut Pascal - Clermont-Ferrand
	Christian Laugier	INRIA - Grenoble
Président	Thierry Denoeux	UTC - HEUDIASYC - Compiègne
Examineurs	Michel Devy	LAAS - Toulouse
	Nicolas Paparoditis	IGN - Saint-Mandé
Directeurs de thèse	Véronique Cherfaoui	UTC - HEUDIASYC - Compiègne
	Philippe Bonnifait	UTC - HEUDIASYC - Compiègne



Table des matières

Résumé	3
1. Introduction générale	7
1.1. Contexte et intention de l'étude	7
1.2. Éléments clefs pour la navigation des véhicules intelligents	9
1.2.1. Localisation	9
1.2.2. Perception de l'environnement	10
1.3. Modèle de représentation géométrique de l'environnement	12
1.3.1. Méthodes "basées primitive"	12
1.3.2. Méthodes "basées grille"	13
1.3.3. Représentation hybride	14
1.4. Problématique de la thèse	14
1.5. Organisation du manuscrit	17
2. Grilles métriques spatio-référencées pour la perception robotique	19
2.1. Introduction	19
2.2. Grilles Métriques	20
2.2.1. Approximation discrète d'une fonction continue 2D	20
2.2.2. Référencement spatial des grilles métriques	25
2.3. Grille d'occupation pour la perception	30
2.3.1. Grille d'occupation	30
2.3.2. Perception avec des grilles	33
2.4. Construction par fusion temporelle d'une grille	35
2.4.1. Principe	35
2.4.2. Prédiction et rémanence de l'information dans les grilles	36
2.4.3. Correction de la grille et détection des cellules mobiles	36
2.4.4. Architecture du système en fonction du repère de référence	37
2.5. Conclusion	39
3. Formalismes de gestion d'incertitude pour les grilles de perception	41
3.1. Introduction	41
3.2. Grille d'accumulation	42
3.2.1. Principe	42
3.2.2. Prise de décision	42
3.2.3. Fusion temporelle dans les grilles d'accumulation	43

3.3.	Grille d'occupation probabiliste	46
3.3.1.	Principe	46
3.3.2.	Grille d'occupation en environnement statique	47
3.3.3.	Règle de décision	48
3.3.4.	Fusion de grilles probabilistes	49
3.3.5.	Grille d'occupation en environnement dynamique	51
3.4.	Grille d'occupation évidentielle	54
3.4.1.	Principe	54
3.4.2.	Fonction de décision	55
3.4.3.	Grille évidentielle en environnement dynamique	58
3.4.4.	Opérateurs de combinaison	60
3.4.5.	Entropie et spécificité d'une fonction de masse	66
3.5.	Traitements particuliers possibles dans les grilles évidentielles	68
3.5.1.	Représentation de l'ignorance	68
3.5.2.	Identification du conflit à partir de $m(\emptyset)$	69
3.5.3.	Comparatif des règles de combinaison	71
3.5.4.	Étude de la dynamique de mise à jour	74
3.6.	Comparaison des cadres probabiliste et évidentiel	77
3.6.1.	Représentation de l'ignorance	78
3.6.2.	Masse conflictuelle	81
3.6.3.	Comparaison des comportements de mise à jour	82
3.6.4.	Choix du mode de décision	82
3.7.	Conclusion	86
4.	Construction d'une <i>ScanGrid</i> à partir d'un lidar	87
4.1.	Introduction	87
4.2.	Télémetre laser à balayage	88
4.2.1.	Modèle de faisceau laser et mesure	89
4.2.2.	Amélioration du lidar pour les applications robotiques	91
4.2.3.	Lidar utilisé	94
4.3.	Modèle de capteur lidar	95
4.3.1.	Lien entre mesure et occupation	95
4.3.2.	Principes pour l'élaboration de modèles inverses de capteur	96
4.3.3.	Détection du plan du sol	98
4.3.4.	Modèles inverses de capteur sur grille polaire	99
4.4.	Conclusion	112
5.	Fusion séquentielle multi-grilles et mise en œuvre expérimentale	113
5.1.	Introduction	113
5.2.	Fusion séquentielle halo-centrée	114
5.2.1.	Rappel du principe	114
5.2.2.	Système expérimental	115
5.2.3.	Injection d'information cartographique	118
5.2.4.	Implémentation	122

5.2.5.	Résultats de la méthode d'accumulation	124
5.2.6.	Résultats de la méthode évidentielle	129
5.3.	Fusion séquentielle égo-centrée	136
5.3.1.	Rappel du principe	136
5.3.2.	Présentation de la plate-forme expérimentale	136
5.3.3.	Implémentation	140
5.3.4.	Méthode de validation expérimentale	143
5.3.5.	Résultats	145
5.4.	Conclusion	149
6.	Conclusion et perspectives	151
6.1.	Conclusion générale	151
6.2.	Perspectives	153
A.	Opérations sur les grilles	157
A.1.	Transformation spatiale	157
A.1.1.	Principe du traitement de l'antécédent	157
A.1.2.	Interpolation multivariée	158
A.1.3.	Erreurs d'interpolation successives	161
A.1.4.	Ré-échantillonnage de grille	162
A.1.5.	Transformation solide	163
A.2.	Transformation intrinsèque	163
A.2.1.	Changement de repère	164
A.2.2.	Transformation polaire / cartésienne	164
A.2.3.	Agrandissement et troncature	165
A.3.	Transformation de contenu	165
A.3.1.	Normalisation	166
A.4.	Combinaison	166
A.4.1.	Fusion statique de deux grilles	166
A.4.2.	Inter-corrélation	166
B.	Affaiblissement contextuel	169
C.	Détection des points du sol	175
	Bibliographie	177
	Nomenclature	185

Table des figures

1.1.	Exemple d'une scène urbaine typique	15
1.2.	Contrainte d'espace libre	16
1.3.	Contrainte de permission	16
1.4.	Espace navigable provenant de la fusion des contraintes \mathcal{F} et \mathcal{A}	17
2.1.	Approximation d'une fonction 2D	20
2.2.	Plan image pour un modèle de camera projective	21
2.3.	Échantillonnage d'une fonction suivant un maillage 2D	22
2.4.	Différence d'approximation d'une fonction en fonction du pas d'échantillonnage	23
2.5.	Phénomène de repliement spectral sur une image	24
2.6.	Modélisation des barreaux d'une barrière en vue de dessus à l'aide d'une grille	24
2.7.	Repère mobile dans un monde plan	26
2.8.	Définitions des repères.	27
2.9.	Correspondance entre un repère habituel d'une matrice image et le repère \mathcal{R}_M de la grille.	29
2.10.	Exemples de grilles d'occupation	31
2.11.	Interactions du système de fusion.	33
2.12.	Approche ScanGrid	34
2.13.	Principe prédicteur - correcteur	35
2.14.	Approche par mise à jour séquentielle	35
2.15.	Grille halo-centrée	37
2.16.	Architecture du système de fusion halo-centrée	37
2.17.	Grille égo-centrée	38
2.18.	Architecture du système de fusion égo-centrée	39
2.19.	Méthode égo-centrée par fusion complète	39
3.1.	Fusion de grilles d'accumulation	43
3.2.	Mise à jour de la MapGrid par accumulation	44
3.3.	Processus d'accumulation d'une cellule en fonction des observations	45
3.4.	Profil d'allocation 1D issu d'un capteur parfait	48
3.5.	Notion de vitesses discrètes	52
3.6.	Structure estimateur - prédicteur du BOF	53
3.7.	Utilisation du conflit pour caractériser le sens de déplacement d'un mobile	71

3.8. Simulation d'évolution des fonctions de masse issue de la mise à jour temporelle par fusion en utilisant différentes règles	74
3.9. Simulation d'évolution des fonctions de masse avec affaiblissement . .	76
3.10. Comparaison des transferts de croyance	77
3.11. Initialisation d'une grille d'occupation	78
3.12. Effet de l'affaiblissement	79
3.13. Fusion de deux cellules en conflit	81
3.14. Comparaison du comportement de différentes fusions	83
4.1. Principe de mesure lidar	88
4.2. Modèle de faisceau gaussien	90
4.3. Signal de retour d'un lidar multi-échos	91
4.4. Balayage angulaire du laser utilisant un miroir rotatif	92
4.5. Télémètre laser 4 nappes	93
4.6. Exemple de données brutes issues de lidars	93
4.7. Résolution angulaire de l'Alasca XT	94
4.8. Modèle de capteur 1D	95
4.9. Modèle de capteur ultra-sonore 2D	96
4.10. Exemple de construction d'un secteur angulaire à partir de trois échos	98
4.11. Modèle de capteur 2D	98
4.12. Exemple de construction d'un secteur angulaire contenant trois échos dont deux tapant le sol	99
4.13. Modèle de capteur par accumulation	100
4.14. Modèle de capteur évidentiel simplifié	101
4.15. Profils du modèle de capteur pour deux mesures élémentaires	102
4.16. Résultat de la fusion des deux bba avec une fusion conjonctive	104
4.17. Résultat de la construction de m_f à partir des deux bba	105
4.18. Profil des bba élémentaires prenant en compte les effets de masquages après affaiblissement	106
4.19. Résultat de la fusion des deux bba prenant en compte les effets de masquage	107
4.20. Modèle de capteur évidentiel gaussien obtenu par fusion de chaque écho	111
5.1. Principe de l'approche de la fusion halo-centrée	114
5.2. Véhicule utilisé pour l'enregistrement du jeu de données	115
5.3. Trajectoire du véhicule lors de l'enregistrement du jeu de données. .	116
5.4. Synchronisation des données.	117
5.5. Représentation dans un moteur en rendu temps réel de la carte tex- turée et de la chaussée.	118
5.6. Représentation en fil de fer du maillage de la chaussée.	119
5.7. Représentation 3D de la Carte CityVIP	119
5.8. Représentation de la PriorGrid	120
5.9. Cadre de discernement raffiné	121
5.10. Architecture du système halo-centrée	122

5.11. Résultat de la cartographie utilisant le formalisme d'accumulation . . .	124
5.12. Illustration de l'intérêt du multi-nappe	125
5.13. Illustration de l'intérêt du multi-écho	126
5.14. Résultats obtenus lors de deux scénarios illustrant la détection d'objets en mouvement	128
5.15. Résultat de la cartographie utilisant le formalisme évidentiel	130
5.16. Résultat de la détection d'objets mobiles des méthodes d'accumulation et évidentielle au travers d'une séquence de quatre clichés	131
5.17. Résultat de détection des objets en mouvement par l'approche évidentielle	132
5.18. Comparaison des capacités de détection des objets mobiles des méthodes d'accumulation et évidentielle 1	133
5.19. Comparaison des capacités de détection des objets mobiles des méthodes d'accumulation et évidentielle 2	134
5.20. Résultat de fusion contextuelle	135
5.21. Architecture du système égo-centré	137
5.22. Le véhicule expérimental Carmen	138
5.23. Système de positionnement : Span CPT	139
5.24. Principaux outils de la plate-forme logicielle de PACPUS	140
5.25. Positionnement de la caméra virtuelle pour faire le rendu de la carte.	143
5.26. Reprojection de la carte dans une grille pour créer la PriorGrid	144
5.27. Projection de la grille dans l'image de la camera	145
5.28. Résultat de la construction de la CLD : un piéton traverse la route.	146
5.29. Résultat de la construction de la CLD : des véhicules sont garés sur les bords de la route.	147
5.30. Résultat de la construction de la CLD : détection du mouvement d'un cycliste	147
5.31. Résultat de la construction de la CLD : la détection des terre-pleins centraux est quasiment retirée de la grille par la contrainte de carte.	148
5.32. Résultat de la construction de la CLD : effet d'une erreur de localisation	149
5.33. Résultat de la construction de la CLD : effet d'un ralentisseur	149
A.1. Classification d'opérations classiques sur les grilles	157
A.2. Exemple de transformation géométrique	158
A.3. Interpolation d'une cellule	159
A.4. Interpolation bilinéaire de la valeur d'une cellule	160
A.5. Approximations dues à l'interpolation	162
A.6. Représentation de la grille polaire	165
A.7. Fusion cellule à cellule de deux grilles	167
B.1. Profils du modèle de capteur pour deux mesures élémentaires	170
B.2. Résultat de la construction de m_f à partir des deux bba	172

C.1. Détection du plan du sol 176

Remerciement

Tout au long des trois années et demi que j'ai passé au laboratoire Heudiasyc de l'UTC, j'ai rencontré de nombreuses personnes qui m'ont aidé, accompagné et avec lesquels j'ai passé de très bons moments. Aussi je souhaiterais remercier toutes les personnes qui ont pu contribuer directement ou indirectement à la réalisation de ces travaux.

Je souhaite remercier Ali Charrara, le directeur du laboratoire Heudiasyc, qui m'a accueilli au sein de son établissement dans les meilleures conditions possibles pour réaliser cette thèse.

Je remercie aussi tout particulièrement les membres du jury. Les rapporteurs M. Paul Checchin et M. Christian Laugier pour leur travail de relecture et leurs remarques pertinentes vis à vis de mon travail. Les examinateurs : Michel Devy pour son intérêt, Nicolas Paparoditis pour ses conseils et pour toute l'expertise que l'IGN a pu m'apporter dans le cadre du projet CityVIP et Thierry Denoueux pour tout ce qu'il a pu m'apprendre sur le formalisme des fonctions de croyance.

Merci à Véronique Cherfaoui et à Philippe Bonnifait, mes directeurs de thèse pour m'avoir toujours accompagné surtout dans les moments difficiles durant ces trois années. Ils m'ont permis de vraiment progresser au cours de ces années et ont montré leur confiance en m'impliquant directement dans plusieurs projets.

Je remercie mes amis et collègues du laboratoire Heudiasyc pour leur accueil, l'aide qu'ils ont su m'apporter durant ma thèse et surtout pour la convivialité ainsi que la bonne ambiance qui régnait au laboratoire. Je remercie particulièrement Gérald pour le temps qu'il a su me consacrer et pour son expertise.

Je souhaite encore remercier mes parents pour m'avoir toujours soutenu, ma grande sœur pour m'avoir particulièrement aidé lors de mon périple à Compiègne et mon petit frère.

Enfin je souhaite te remercier Jiqiong.

Résumé

Les travaux de recherche présentés dans cette thèse portent sur le problème de la perception de l'environnement en milieu urbain, complexe et dynamique et ce en présence de mesures extéroceptives bruitées et incomplètes obtenues à partir de capteurs embarqués. Le problème est formalisé sous l'angle de la fusion de données capteurs à l'aide d'une représentation spatiale de l'environnement. Ces travaux ont été réalisés pour la navigation autonome de véhicules intelligents dans le cadre du projet national ANR CityVIP.

Après avoir considéré les principaux formalismes de modélisation de l'incertitude, un système de fusion de grilles spatio-référencées gérant l'incertitude avec des fonctions de croyances est étudié. Ce système est notamment capable de fusionner les mesures d'un lidar multi-nappes et multi-échocs, obtenues à différents instants pour construire une carte locale dynamique sous la forme discrète d'une grille d'occupation évidentielle.

Le principal avantage des fonctions de croyance est de représenter de manière explicite l'ignorance et ne nécessite donc pas d'introduire d'information à priori non fondée. De plus, ce formalisme permet d'utiliser facilement l'information conflictuelle pour déterminer la dynamique de la scène comme par exemple les cellules en mouvement.

Le formalisme de grilles d'occupation évidentielles est présenté en détails et un modèle de capteur lidar multi-nappes et multi-échocs est ensuite proposé. Deux approches de fusion séquentielle multi-grilles sont étudiées selon les paradigmes halo-centré et égo-centré. Enfin, l'implémentation et les tests expérimentaux des approches sont décrits et l'injection d'informations géographiques connues a priori est étudiée.

La plupart des travaux présentés ont été implémentés en temps réel sur un véhicule du laboratoire et de nombreux tests en conditions réelles ont été réalisés avec une interface d'analyse de résultat utilisant une rétro-projection dans une image grand angle.

Les résultats ont été présentés dans 5 conférences internationales [Moras *et al.*, 2010, Moras *et al.*, 2011a, Moras *et al.*, 2011b, Moras *et al.*, 2012, Kurdej *et al.*, 2012] et le système expérimental a servi à la réalisation de démonstrations officielles dans le cadre du projet CityVIP à Paris et lors de la conférence IEEE *Intelligent Vehicles Symposium* 2011 en Allemagne.

Summary

The research presented in this thesis focuses on the problem of the perception of the urban environment which is complex and dynamic in the presence of noisy and incomplete exteroceptive measurements obtained from on-board sensors. The problem is formalized in terms of sensor data fusion with a spatial representation of the environment. This work has been carried out for the autonomous navigation of intelligent vehicles within the national project ANR CityVIP.

After having considered various formalisms to represent uncertainty, a fusion of spatio-referenced grids managing uncertainty with belief functions is studied. This system is capable of merging multi-layers and multi-echoes lidar measurements, obtained at different time indexes to build a dynamic local map as a discrete evidential occupancy grid.

The main advantages of belief functions are, firstly, to represent explicitly ignorance, which reduces the assumptions and therefore avoid introducing wrong a priori information and, secondly, to easily use conflicting information to determine the dynamics of the scene such as movements of the cells.

The formalism of evidential occupancy grids is then presented in details and two multi-layers and multi-echos lidar sensor models are proposed. The propagation of the information through geometrical transformations is formalized in a similar way of image transformation framework. Then, the implementation of the approach is described and the injection of prior geographic information is finally investigated.

Most of the works presented have been implemented in real time on a vehicle and many tests in real conditions have been realized.

The results of these researches were presented through five international conferences [Moras *et al.*, 2010, Moras *et al.*, 2011a, Moras *et al.*, 2011b, Moras *et al.*, 2012], [Kurdej *et al.*, 2012] and the experimental vehicle was presented at the official demonstration project CityVIP in Paris and at the IEEE Intelligent Vehicles Symposium 2011, in Germany.

1. Introduction générale

Sommaire

1.1. Contexte et intention de l'étude	7
1.2. Éléments clefs pour la navigation des véhicules intelligents	9
1.3. Modèle de représentation géométrique de l'environnement	12
1.4. Problématique de la thèse	14
1.5. Organisation du manuscrit	17

1.1. Contexte et intention de l'étude

Depuis plusieurs décennies, le parc automobile mondial ne cesse d'augmenter et on compte plus d'un milliard de véhicules recensés dans le monde en 2010. Depuis quelques années, une problématique autour de l'utilisation des véhicules pour la mobilité des personnes en milieu urbain apparaît. En effet, les grands centres urbains sont de plus en plus peuplés et les personnes ont besoin d'avoir une mobilité toujours accrue que les transports en commun ne parviennent pas à assurer. Cela engendre un ensemble de problèmes bien connus des grandes agglomérations mondiales :

- saturation des réseaux routiers actuels,
- saturation des parkings,
- augmentation du risque d'incident en agglomérations,
- augmentation de la pollution dans les grandes villes qui atteint des niveaux inquiétants.

L'automatisation des véhicules pourrait être une solution aux problèmes de transport rencontrés aujourd'hui dans les grandes villes en permettant :

- la réduction du nombre d'accidents (le facteur humain étant à l'origine de 90% des accidents),
- le partage de véhicules individuels qui pourraient se redéployer de façon autonome,
- la mise en place d'un système de navigation coopératif entre les véhicules autonomes permettrait de fluidifier le trafic.

Parallèlement à cela, les systèmes informatiques embarqués ont continué leur évolution ces dernières années et les capacités de traitement en temps-réel deviennent significatives. De nouveaux capteurs, de moins en moins chers et de plus en plus

précis, sont apparus sur le marché. Les moyens de télécommunication sans fils sont maintenant disponibles quasiment partout (et notamment dans les grandes villes de France). L'automatisation des moyens de collecte et de traitement des données cartographiques, tout d'abord satellitaires et aériennes, et maintenant terrestre, a permis la création d'énormes bases de données cartographiques de plus en plus précises comme le Géoportail de l'IGN ou GoogleMap par exemple.

Ces nouvelles solutions sensorielles ont notamment permis lors du concours DARPA Urban challenge [Buehler *et al.*, 2009, Berlin, 2007, Khatib et Groen, 2009] de démontrer que des véhicules pouvaient réaliser des tâches de navigation dans des situations quasi-urbaines. Ceci est aujourd'hui mis en œuvre dans un véhicule autonome présenté par Google qui navigue en conditions réelles et sur des routes ouvertes de certains états américains. Cependant, ces systèmes reposent principalement sur l'utilisation d'un capteur lidar omnidirectionnel de haute résolution Velodyne. Son coût élevé et ses dimensions ne le rendent pas intégrable actuellement dans des véhicules de série.

Dans la continuité du projet ANR MobiVIP, le projet ANR CityVIP de 2008 à 2011 intitulé "*Développement sûr de véhicules individuels adaptés à l'environnement urbain*" a étudié une nouvelle modalité de transport en milieu urbain reposant sur les principales idées suivantes :

- l'utilisation de véhicules autonomes pour le transport individuel,
- aucune modification de l'infrastructure réelle d'évolution (notamment, sans ajout d'amers artificiels),
- l'utilisation de connaissances préalables renseignées dans une carte vectorielle précise enrichie appelée aussi "infrastructure virtuelle",
- utilisation de caméras intégrées dans les véhicules pour la localisation en temps réel,
- utilisation de capteurs lasers de type lidars intégrés dans le véhicule pour la détection, la localisation et l'évitement d'obstacles pare-choc par l'intermédiaire d'un mécanisme appelé "carte locale dynamique".

La mise en place de véhicules autonomes en milieu urbain implique donc que ces véhicules soient capables de naviguer de façon autonome dans un milieu complexe, dynamique et en présence de vulnérables comme des piétons. De façon à prendre en compte l'environnement dynamique avec lequel il est en interaction, le véhicule doit être doté de fonctionnalités avancées de perception.

Dans ce mémoire, nous nous intéressons ainsi aux techniques de perception embarquées sur les robots mobiles évoluant sur route connue de façon à créer en temps-réel une représentation de l'environnement utilisable par le robot qui lui permette de naviguer de façon sûre à partir d'une carte locale dynamique.

1.2. Éléments clés pour la navigation des véhicules intelligents

Nous considérerons un véhicule équipé de technologies intelligentes qui lui fournissent des capacités lui permettant la réalisation de façon autonome d'un ensemble de tâches avancées. Ces tâches peuvent être très variées (sécurité, aide à la conduite, conduite autonome, navigation coopérative, gestion du trafic, contrôle de la réglementation, ...). Parmi toutes ces tâches, la principale est la navigation autonome.

Le concept de navigation autonome est un point central en robotique mobile depuis de nombreuses années. Dans [Levitt et Lawton, 1990], la navigation autonome est définie comme suit.

Définition 1 (Navigation autonome). Il s'agit d'un procédé qui permet de répondre aux questions :

1. Où suis-je ?
2. Où sont les autres éléments de l'environnement par rapport à moi ?
3. Comment atteindre un objectif à partir de l'endroit où je me trouve ?

A partir de cette définition, nous remarquons que le véhicule autonome doit être capable de réaliser les tâches suivantes :

- Se localiser dans un repère de référence et suivre sa trajectoire prévue :
la localisation doit être intègre et suffisamment précise pour permettre au système de contrôle le suivi de la trajectoire de consigne avec une erreur acceptable.
- Détecter et localiser des éléments de l'environnement.
- Planifier sa trajectoire jusqu'à la destination :
Cette planification doit être réalisée en tenant compte de l'information connue de l'environnement (carte) et des capacités du véhicule, tout en minimisant un ou des critères particuliers comme la durée du trajet ou la consommation énergétique.
- Éviter les collisions :
L'évitement de collision est indispensable quand le véhicule autonome se déplace en milieu inconnu ou incertain. C'est le cas lorsque le véhicule a, au moment où il fait la planification, des informations incertaines ou incomplètes sur l'environnement, et lorsqu'il évolue dans un environnement dynamique. L'évitement de collision est alors un processus complexe, qui nécessite la planification en ligne d'une trajectoire (de manière locale ou globale) et avec une contrainte de réactivité suffisante.

Pour la réalisation de ces différentes tâches, le véhicule a besoin de connaître sa position et de caractériser son environnement. Le véhicule intelligent est donc confronté aux problèmes de localisation et de perception.

1.2.1. Localisation

La tâche de localisation sert à répondre à la question essentielle : *où suis-je ?*

Pour se localiser, le véhicule a deux moyens :

Il peut se localiser de façon relative, par rapport à ses positions précédentes (notamment par rapport à sa position initiale). Pour cela, il doit connaître son déplacement et dans ce cas on parle d'odométrie. Cette technique consiste à évaluer le déplacement du véhicule sur un intervalle de temps court à l'aide de capteurs embarqués. On peut reconstruire la trajectoire du véhicule par intégrations successives des déplacements élémentaires. Pour évaluer le déplacement, on peut soit :

- le mesurer directement à partir de capteurs odométriques,
- le reconstruire à partir de mesures sur l'accélération ou sur la vitesse (accéléromètres, gyromètres, radar Doppler,...) et d'un modèle dynamique,
- le déduire à partir de la perception d'amers de l'environnement (caméra, lidar, sonar).

Le véhicule peut aussi se localiser de façon absolue. Dans ce cas, un repère de référence est déterminé. Pour se localiser, le véhicule doit détecter un ensemble d'amers. Si la position de ces amers est connue dans le repère de référence, le véhicule peut alors en déduire sa pose dans ce repère. C'est le principe qui est utilisé par le système de positionnement GPS (Global Positioning System).

1.2.2. Perception de l'environnement

La perception de l'environnement est définie pour les êtres humains de la façon suivante :

Évènement cognitif dans lequel un stimulus ou un objet, présent dans l'environnement immédiat d'un individu, lui est représenté dans son activité psychologique interne, en principe de façon consciente ; fonction psychologique qui assure ces perceptions. (Dictionnaire Larousse 2012)

On peut définir la perception d'un système robotique de façon analogue :

C'est un processus complexe qui met en œuvre l'acquisition d'informations sur l'environnement à l'aide de capteurs extéroceptifs et l'analyse de ces données afin de construire une modélisation de l'environnement exploitable pour réaliser des tâches robotiques.

La perception sert principalement à résoudre 3 types de problèmes.

Ego-localisation Comme nous l'avons déjà remarqué dans la section 1.2.1, la perception permet de se localiser, en répondant à la question "Où est-ce-que je me situe dans mon environnement ?". Ce problème se base principalement sur la détection et la mesure d'amers dans le référentiel du véhicule dont la position absolue est connue. Avec suffisamment de mesures, ce type de localisation permet d'estimer la pose du véhicule à chaque instant indépendamment des instants précédents.

Dans le cas où on ne connaît pas la position des éléments détectés mais, s'ils sont fixes, on peut utiliser la perception de façon à construire une carte dans laquelle

on positionnera les éléments par rapport au véhicule de façon incrémentale et on utilisera cette carte pour se localiser. Ce problème est connue sous le nom de SLAM (Simultaneous Localization And Mapping).

Détection des obstacles, suivi et reconnaissance Le rôle principal de la perception est de détecter et de caractériser les obstacles dans l’environnement du véhicule. On peut décomposer cette action en deux tâches élémentaires : la détection et la localisation des objets de l’environnement.

La détection répond à la question “*Y a-t-il des éléments dans mon environnement ?*”. S’il y a des éléments, alors cette tâche est souvent complétée de la tâche de localisation de l’obstacle qui répond à la question “*Où sont ces éléments ?*”

Ces deux tâches sont à la base de tout système de perception. Elles nécessitent l’utilisation de capteurs extéroceptifs pour récupérer des informations sur l’environnement du robot. Le suivi dans le temps des objets mobiles est nécessaire pour établir une relation temporelle entre les détections du capteur et pour augmenter la robustesse de la détection face aux erreurs de mesure et aux masquages. De plus, le suivi de cibles va permettre d’estimer des paramètres dynamiques tels que la vitesse. Ce problème est généralement adressé sous le nom de MOT (*Moving Object Tracking*).

Enfin, une fois les objets suivis, on peut vouloir les classifier ou les identifier pour augmenter le niveau de connaissance sur ces objets comme dans [Himmelsbach *et al.*, 2009, Nashashibi *et al.*, 2008].

Détermination de l’espace libre De manière duale à la détection des obstacles, la perception est utile pour déterminer les zones libres c’est-à-dire les zones dans lesquelles le robot peut évoluer sans risque de collision. Les zones libres d’obstacles sont généralement déterminées de manière implicite c’est-à-dire déduites de la détection et du suivi des obstacles. Or, il est nécessaire de distinguer un espace réellement libre d’obstacle d’un espace qui n’a pas d’obstacle car il n’a pas été observé. Dans le domaine des véhicules intelligents, de nombreux algorithmes de génération de trajectoires reposent sur une représentation explicite de l’espace navigable (c.f. [Cherubini et Chaumette, 2011, Urmson *et al.*, 2007, Montemerlo *et al.*, 2008]). Il est donc nécessaire de déterminer et caractériser l’espace libre selon un schéma de perception intégrant les erreurs de mesures, les limitations des capteurs et les connaissances à priori afin de calculer l’espace navigable.

1.3. Modèle de représentation géométrique de l'environnement

Chacun des trois problèmes présentés nécessite une modélisation particulière de l'environnement. Bien que la plupart de ces représentations sont métriques, il en existe d'autres comme les représentations topologiques. Nous allons présenter ici les deux principaux modes de représentation géométrique et leur utilisation dans les travaux de perception existants.

1.3.1. Méthodes “basées primitive”

Les méthodes “basées primitive” utilisent des primitives géométriques pour représenter l'environnement. Le type de primitive dépend essentiellement de l'application visée, de l'environnement considéré, de la capacité calculatoire embarquée et de la précision nécessaire. Souvent, les primitives sont 2D et forment une représentation dans le plan horizontal (en vue de dessus). Les problèmes “basés primitive” reposent essentiellement sur un problème d'association de données pour permettre de propager de façon cohérente les primitives dans le temps.

Un grand nombre de systèmes de “Featured SLAM” [Montemerlo *et al.*, 2002] utilisent une représentation de l'environnement sous forme d'amers représentés par des points 2D. Dans ce type d'approche, l'environnement est représenté par un vecteur d'état qui contient les coordonnées des amers. Ce vecteur est filtré au cours du temps à l'aide d'un filtre de type Kalman (EKF, UKF) [Kalman, 1960, Julier et Uhlmann, 2004] ou de filtre particulaire [Ristic *et al.*, 2004]. La mise en correspondance entre les détections et les pistes propagées un problème se résume à un problème d'association des données qui est traité de différentes façons (NN, JPDA, MHT, etc...)[Bar-Shalom, 2000, Hähnel *et al.*, 2005]. Enfin, d'autres types de primitives peuvent aussi être utilisés pour la localisation. [Xu et Collins, 2010] utilise une représentation par segments alors que [Royer *et al.*, 2005] et [Rodriguez Florez *et al.*, 2009] utilisent des points d'intérêt dans l'image d'une caméra.

Beaucoup de travaux concernant la détection d'obstacles utilisent aussi des primitives 2D pour représenter l'environnement. Le type de primitives dépend généralement de l'application. Dans [Petrovskaya et Thrun, 2009, Fayad et Cherfaoui, 2007] où l'on cherche à détecter les véhicules, les objets dans sont représentés par des boîtes rectangulaires. Dans [Fayad et Cherfaoui, 2008], on utilise de simples points 2D associés à une ellipse d'incertitude pour représenter les piétons et [Gidel *et al.*, 2008] représente les piétons en utilisant des amas de points. Les primitives ainsi détectées sont ensuite suivies au cours du temps en utilisant des algorithmes de filtrage généralement construits autour du filtre de Kalman ou du filtre particulaire.

Comme nous l'avons précisé, le problème de SLAM considère que les amers sont fixes. Pour utiliser ce type d'approche en présence d'objets mobiles, les algorithmes

de SLAMMOT (par exemple [Zhao *et al.*, 2009]) propose résoudre le problème de MOT pour détecter les objets mobiles de l'environnement et ne pas utiliser ces objets comme amers de localisation.

Les avantages sont :

- une représentation simple,
- une propagation dans le temps facile,
- une prise en compte des objets mobiles,
- une faible consommation de mémoire.

Les inconvénients sont :

- une représentation non exhaustive et donc insuffisante pour faire de la navigation,
- une grande très sensibilité aux résultats de l'association.

1.3.2. Méthodes “basées grille”

D'autres approches utilisent une représentation par grille. Les grilles cartésiennes 2D utilisent un maillage structuré pour partitionner le plan horizontal de l'environnement en cellules. Chaque cellule correspond à une portion de surface du sol. Ainsi, la grille ressemble à une vue de dessus de la scène. L'état d'occupation de chaque cellule est évalué indépendamment. Généralement, les cellules sont carrées mais certains travaux [Herrmann *et al.*, 2010] ont considéré des géométries de grilles différentes.

Les premiers travaux sur les grilles d'occupation utilisaient une grille 2D pour construire et mettre à jour une carte de l'environnement considérée comme statique [Elfes, 1989c, Elfes, 1989a, Elfes, 1989b, Elfes, 1990]. Initialement limité par la complexité calculatoire, ce type d'approche est maintenant très utilisé pour la navigation :

- les techniques de SLAM sur grille se sont développées [Levinson et Thrun, 2010, Thrun *et al.*, 1998, Steux et El, 2010, Bourgault *et al.*, 2002]. Dans ce cas, on construit une grille fixe qui permet de corriger la position à partir de chaque nouvelle observation.
- les grilles peuvent être utilisées dans les problèmes de MOT. Dans [Coué *et al.*, 2006], une grille est liée au véhicule puis cette grille est mise à jour en calculant l'occupation mais aussi en calculant le vecteur vitesse de chaque cellule. A partir de l'occupation, de la vitesse et de la position des cellules, on peut alors les regrouper au sein d'objets.
- Enfin, d'autres travaux utilisent des grilles pour résoudre les deux problèmes simultanément (SLAMMOT) [Gate, 2009].

Pour prendre en compte la hauteur, certains travaux utilisent des grilles dites *2D et demi*. Ces grilles contiennent dans chaque cellule une information de hauteur maximale [Himmelsbach *et al.*, 2008]. Enfin d'autres approches utilisent des grilles 3D, la grille partitionne un volume et les cellules sont des cubes [Miyasaka *et al.*, 2009].

Pour optimiser ces approches, des grilles multi-échelles ont été développées. Ces grilles proposent d'implémenter une structure en arbre. Par exemple pour une grille

2D, on peut définir une grille Quadtree, dans laquelle chaque cellule peut être découpée en 4 sous cellules et ce de façon récursive. Grâce à ce procédé, les zones uniformes sont couvertes par de grosses cellules et les zones hétérogènes sont modélisées plus finement. L'intérêt immédiat est la réduction de l'espace mémoire occupé par la grille ainsi que la réduction de calculs pour les zones homogènes. [Xie *et al.*, 2010] utilisent la représentation multi-échelle pour faciliter la mise en correspondance de grille en procédant en cascade avec une résolution de plus en plus fine.

Les avantages sont :

- une représentation très complète que permet de faire de la navigation,
- pas d'hypothèse sur la géométrie de l'environnement que l'on observe.

Les inconvénients sont :

- la propagation dans le temps est complexe,
- la prise en compte des objets mobiles est difficile,
- l'approche est coûteuse en termes de calcul et de mémoire.

1.3.3. Représentation hybride

Considérant que chacune des deux représentations possède certains avantages et certains inconvénients, des travaux utilisent une double représentation comportant à la fois une représentation par primitives et une représentation par grille.

Plusieurs stratégies ont été proposées dans littérature. [Himmelsbach *et al.*, 2008] utilisent une grille $2D\frac{1}{2}$ pour construire une carte de l'environnement à partir de chaque scan du télémètre Velodyne. Puis à partir de cette grille, il crée une représentation des obstacles avec des boîtes en segmentant les groupes de cellules suffisamment hauts.

[Wurm *et al.*, 2007] proposent d'utiliser une représentation duale pour la navigation de robots en environnements ouverts. Il utilise une représentation par grille pour avoir une représentation complète à l'intérieur des bâtiments et une représentation par primitives pour l'extérieur qui est moins structuré et moins contraint.

1.4. Problématique de la thèse

Dans cette thèse, on considère un véhicule intelligent qui évolue en milieu urbain. Ce milieu, schématisé sur la figure 1.1, est dense, relativement structuré et dynamique.

Pour faire naviguer un véhicule autonome, il est indispensable de caractériser la navigabilité de son environnement de façon à pouvoir évaluer si une trajectoire est réalisable ([Delmas, 2011]). La navigabilité d'un point P de l'environnement peut être exprimée à partir d'un ensemble de contraintes comme l'occupation, la praticabilité ou encore l'autorisation. On peut séparer ces contraintes en deux groupes ; les contraintes dures, i.e. le véhicule ne peut physiquement pas traverser P et les

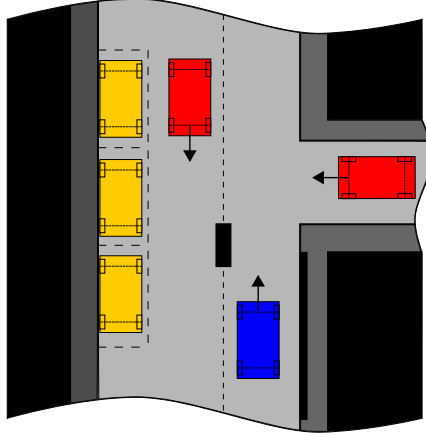


Figure 1.1.: Exemple d’une scène urbaine typique. L’ego-véhicule est en bleu foncé (en bas).

contraintes molles, i.e. on ne veut pas que le véhicule traverse P . Si l’environnement est dynamique, la navigabilité et les contraintes dépendent aussi du temps.

Afin de traiter de façon formelle ce problème, on définit dans un premier temps, à chaque instant t , un état de navigabilité binaire $\mathcal{N}(P, t)$ de tout point P de l’espace comme dans la définition 1.1. Cette fonction est définie pour un véhicule donné.

$$\begin{aligned} \mathcal{N}(P) &= 1, \text{ si } P \text{ est navigable} \\ \mathcal{N}(P) &= 0, \text{ si } P \text{ est non navigable} \end{aligned} \tag{1.1}$$

On peut de la même façon définir un état pour chacune des contraintes. Pour l’occupation, par exemple, on définit une fonction *free* $\mathcal{F}(P, t)$, voir équation 1.2. Cet état représente le fait qu’un point du plan est occupé ou non par un obstacle. La figure 1.2 illustre la fonction *free* estimée à partir des capteurs du véhicule.

$$\begin{aligned} \mathcal{F}(P) &= 1, \text{ si } P \text{ est libre} \\ \mathcal{F}(P) &= 0, \text{ si } P \text{ est occupé} \end{aligned} \tag{1.2}$$

On peut définir d’autres états binaires pour chaque contrainte \mathcal{P} (praticabilité) et \mathcal{A} (autorisation). La figure 1.3 illustre la contrainte de permission, définie par l’espace réglementaire. Cette contrainte est, par exemple, produite à partir d’une carte qui précise la surface de la chaussée et les voies autorisées.

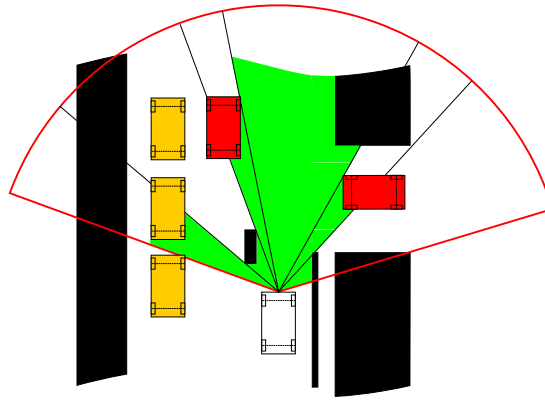


Figure 1.2.: Contrainte d'espace libre : en vert, l'espace libre détecté par le véhicule.

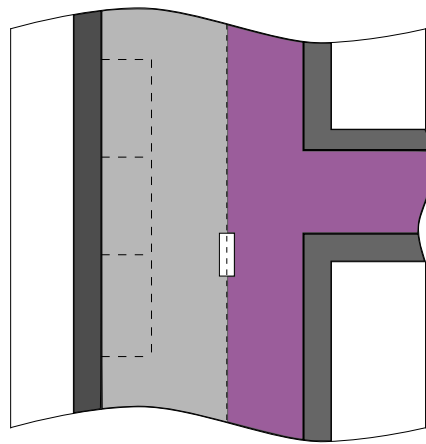


Figure 1.3.: Contrainte de permission : chaussée (gris) + espace réglementaire (violet).

On peut calculer la navigabilité d'un point comme l'intersection de toutes les contraintes (voir équation 1.3), pour finalement calculer l'espace navigable comme sur la figure 1.4 qui correspond à l'expression logique :

$$\mathcal{N} = \mathcal{F} \text{ and } \mathcal{P} \text{ and } \mathcal{A} \quad (1.3)$$

Les états \mathcal{F} et \mathcal{N} sont, par exemple, reliés par les implications logiques suivantes :

$$\bar{\mathcal{F}} \Rightarrow \bar{\mathcal{N}} \text{ et } \mathcal{N} \Rightarrow \mathcal{F}$$

En d'autres termes, si le point P n'est pas libre, cela implique que le robot ne peut pas traverser P et par réciproque, pour que l'on puisse le traverser, il faut que le point en

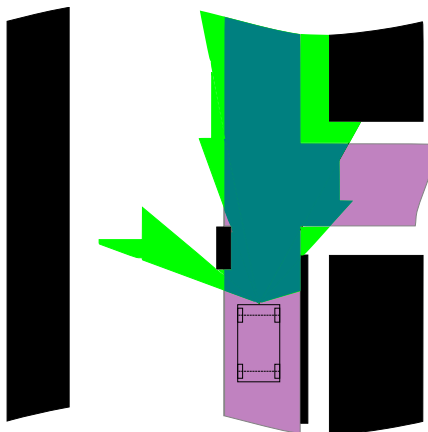


Figure 1.4.: Espace navigable (vert foncé) provenant de la fusion des contraintes \mathcal{F} et \mathcal{A} .

question soit libre. C'est pour cela que l'on s'intéresse aux contraintes d'occupation. La détermination de l'espace libre est alors le point essentiel pour la détermination de l'espace navigable.

Dans la suite de cette thèse, on s'intéressera à l'estimation et la représentation de la fonction d'occupation $\mathcal{O}(P, t)$ qui est l'inverse de $\mathcal{F}(P, t)$, comme c'est l'usage en robotique.

$$\mathcal{O} = \bar{\mathcal{F}}.$$

On aborde donc le problème de la perception de l'environnement en milieu urbain, complexe et dynamique et ce en présence de mesures extéroceptives bruitées et incomplètes obtenues à partir de capteurs embarqués. Le problème est formalisé sous l'angle de la fusion de données capteurs à l'aide d'une représentation spatiale de l'environnement. On proposera une méthode "basée grille" pour construire une carte locale dynamique de l'environnement du véhicule qui soit capable de représenter l'espace libre, l'espace occupé et de détecter le mouvement dans la scène. Nous porterons une attention particulière à la prise en compte et à la gestion des incertitudes, ainsi qu'aux contraintes embarquées et temps réel liées à ce type d'applications.

1.5. Organisation du manuscrit

La suite du mémoire est organisée comme suit.

Dans le second chapitre, nous présentons et formalisons l'utilisation de grilles métriques pour la perception. Nous nous intéressons notamment au parallèle entre les grilles métriques et les images matricielles. Puis, nous proposons une architecture modulaire de construction de grille dont l'objectif est de déterminer et caractériser

l'espace libre. La modularité de cette architecture permet de mettre à jour la grille de perception par une fusion séquentielle soit dans un repère fixe soit dans le repère du véhicule.

Dans le troisième chapitre, nous présentons les formalismes d'accumulation et bayésien utilisés dans les grilles d'occupation pour représenter les incertitudes. Nous détaillons les méthodes de mise à jour et les méthodes de prise de décision. Nous proposons une approche évidentielle reposant sur l'utilisation de la théorie de Dempster-Shafer et nous comparons ce formalisme avec le cadre bayésien et les méthodes d'accumulation. Grâce à une hypothèse simplificatrice d'environnement statique, plusieurs opérateurs de fusion sont proposés, étudiés et comparés dans des scénarios dynamiques parfaitement contrôlés en simulation. Un mécanisme d'affaiblissement et l'analyse du conflit sont proposés pour gérer la dynamique de la scène. L'intérêt du cadre évidentiel pour la prise de décision et la gestion de différents types d'ignorance est illustré.

Dans le quatrième chapitre, plusieurs modèles de capteur sur grille pour télémètre laser à balayage multi-nappes multi-echos sont présentés et étudiés. En particulier, grâce à l'utilisation de coordonnées polaires qui permettent d'exprimer les mesures dans un repère adéquat, deux nouveaux modèles évidentiels sont développés.

Dans le cinquième chapitre, nous décrivons la mise en œuvre expérimentale de deux approches de fusion séquentielle. L'une est "halo-centrée" et l'autre "égo-centrée". Nous montrons les performances de ces approches évidentielles appliquées à des données expérimentales enregistrées et nous les confrontons à une approche d'accumulation utilisée comme expérience test pour la comparaison. Puis, nous détaillons l'implémentation en temps-réel sur un véhicule expérimental du laboratoire. Cette dernière permet de faire une validation qualitative des performances grâce à la re-projection de la carte locale dynamique dans l'image d'une caméra de contrôle dotée d'une optique grand angle.

Finalement, ce mémoire se termine par des conclusions et perspectives.

2. Grilles métriques spatio-référencées pour la perception robotique

Sommaire

2.1. Introduction	19
2.2. Grilles Métriques	20
2.3. Grille d'occupation pour la perception	30
2.4. Construction par fusion temporelle d'une grille	35
2.5. Conclusion	39

2.1. Introduction

Pour assurer la tâche de navigation autonome, le robot doit connaître la transversalité et l'occupation de l'espace. Parmi les modèles de représentation de l'environnement, des approches discrètes sont couramment utilisées en robotique, notamment au travers de la représentation par "*grilles d'occupation*" (voir [Elfes, 1989c]). Inspiré du traitement d'image numérique, ce formalisme permet de représenter l'environnement sans faire d'hypothèse sur la géométrie des éléments présents.

Les grilles d'occupation reposent sur un formalisme de grilles métriques présentées dans ce chapitre. Nous insisterons sur la similarité avec le traitement d'images, ce dernier mettant en place un grand nombre d'outils performants à la fois algorithmiques, logiciels et matériels. A titre d'exemple on citera la bibliothèque de calcul *OpenCV* [Bradski et Kaehler, 2008]. Nous décrirons, ensuite, comment utiliser les grilles métriques 2D pour représenter l'occupation de l'environnement. Enfin, nous présenterons l'architecture d'un système de perception en environnement dynamique utilisant des grilles d'occupation, basé sur une méthode de mise à jour par fusion multi-grilles. Deux variantes sont développées : une halo-centrée et une autre égo-centrée.

2.2. Grilles Métriques

2.2.1. Approximation discrète d'une fonction continue 2D

Dans cette section, on considère un espace à 2 dimensions E associé à un repère $\mathcal{R}_0(0, \vec{x}, \vec{y})$. Soit une fonction continue $f(x, y) : E \rightarrow [0; 1]$ que l'on cherche à estimer, définie pour tout point $P(x_P, y_P)$ de E . Pour modéliser f , une des méthodes possibles consiste à l'approcher par des modèles continus paramétriques (du type base de fonctions). D'autres méthodes consistent en une approximation discrète en réalisant un échantillonnage sur E . Par exemple, la figure 2.1b présente une discrétisation de la fonction donnée dans la figure 2.1a.

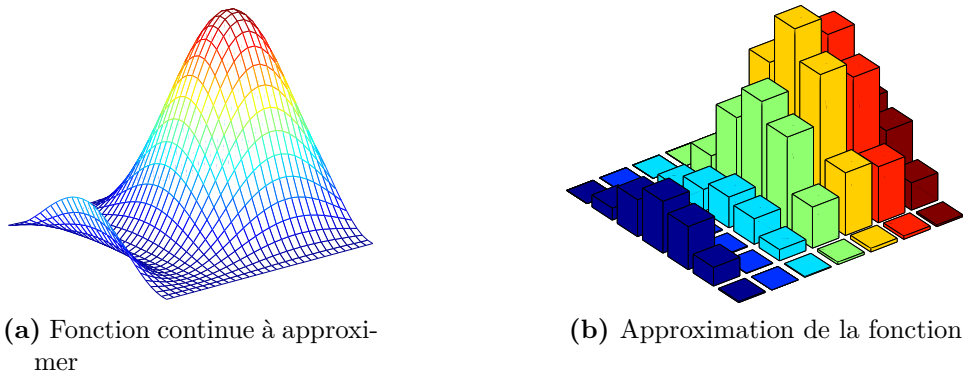


Figure 2.1.: Approximation d'une fonction 2D.

La théorie du traitement numérique d'image permet d'approximer une fonction *intensité*, sur généralement 1 à 4 canaux, à l'aide d'une grille constituée d'un ensemble de pixels. Ce paragraphe décrit comment utiliser ce cadre pour formaliser le problème d'approximation de fonctions 2D.

Discrétisation spatiale On considère la fonction f définie précédemment, dans le cas où E est le plan image d'une caméra dotée de son repère image $\mathcal{R}_I(I, \vec{u}, \vec{v})$, voir figure 2.2. L'opération de discrétisation spatiale qui permet de créer une image matricielle G consiste en un échantillonnage de f de façon régulière dans l'espace suivant les axes de \mathcal{R}_I .

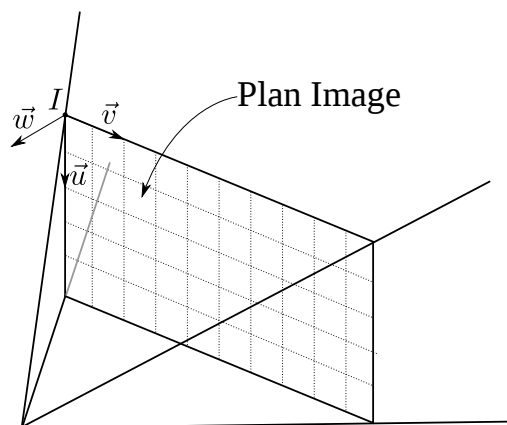


Figure 2.2.: Plan image pour un modèle de camera projective.

On peut considérer l'échantillonnage d'une fonction en un point P_e , de coordonnées (u_{P_e}, v_{P_e}) , comme le produit de la fonction à échantillonner et de la fonction de Dirac δ . Cela permet de définir la fonction f_e échantillonnée en ce point :

$$f_e(u, v) = f(u, v) \cdot \delta(u - u_{P_e}, v - v_{P_e}). \quad (2.1)$$

Où $\delta(x, y)$ est la fonction définie de la façon suivante :

$$\delta(x, y) = \begin{cases} +\infty & \text{si } (x, y) = (0, 0) \\ 0 & \text{sinon} \end{cases}, \quad (2.2)$$

telle que :

$$\int \int \delta(x, y) dx dy = 1. \quad (2.3)$$

En général, on utilise des périodes d'échantillonnage spatial Δ_u et Δ_v constantes. Dans ce cas, les points d'échantillonnage forment un maillage rectangulaire. Chaque point d'échantillonnage est repéré par ses coordonnées discrètes dans le maillage¹ (i, j) et ses coordonnées dans le repère \mathcal{R}_I . Si le point d'indice $(0, 0)$ a pour coordonnées $({}^0u, {}^0v)$, alors les coordonnées des points échantillonnage sont :

$$P_{ij} = \begin{bmatrix} i \cdot \Delta_u + {}^0u \\ j \cdot \Delta_v + {}^0v \end{bmatrix} \quad (2.4)$$

1. D'un point de vue de notation, on indicera en bas à droite les indices discrets.

Dans la suite du chapitre on considèrera que les coordonnées 0u et 0v sont nulles pour simplifier les expressions mathématiques.

$$P_{ij} = \begin{bmatrix} i \cdot \Delta_u \\ j \cdot \Delta_v \end{bmatrix}. \quad (2.5)$$

Le maillage \mathcal{M} est défini comme un peigne de Dirac à 2 dimensions :

$$\mathcal{M}(u, v) = \sum_i \sum_j \delta(u - i \cdot \Delta_u, v - j \cdot \Delta_v). \quad (2.6)$$

Dans ce cas, la fonction échantillonnée est définie de la façon suivante :

$$f_e(u, v) = f(u, v) \cdot \mathcal{M}(u, v). \quad (2.7)$$

La figure 2.3 montre un exemple d'échantillonnage de la fonction f .

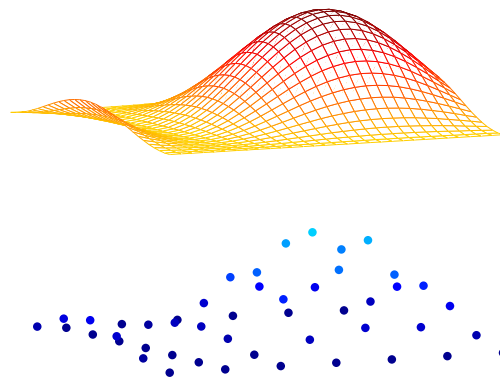


Figure 2.3.: Échantillonnage d'une fonction suivant un maillage 2D : la nappe en haut représente une fonction 2D, le nuage de points du bas représente l'échantillonnage de cette fonction par le maillage \mathcal{M} .

Pavage A partir des valeurs échantillonnées, on peut définir une approximation de f en tous points. Dans le cas de l'image numérique, on utilise une approximation constante par morceaux autour de chaque point d'échantillonnage. On obtient un pavage rectangulaire si on approxime la fonction en prenant en tout point la valeur échantillonnée la plus proche.

En traitement d'image, cet ensemble de pavés juxtaposés compose l'image numérique où chaque pavé est appelé *pixel*. Dans notre cadre de travail, l'ensemble des pavés compose une grille et chaque pavé est appelé *cellule*.

La fonction approximée f_a s'écrit de la façon suivante :

$$f_a(u, v) = f_e(u, v) * \Pi\left(\frac{u}{\Delta_u}, \frac{v}{\Delta_v}\right) \quad (2.8)$$

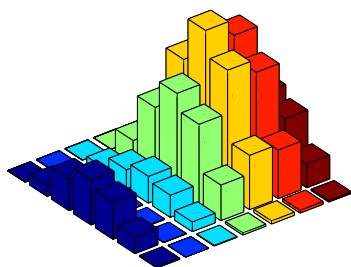
où $*$ est le produit de convolution définie comme :

$$f(x, y) * g(x, y) = \int \int f(x - u, y - v) \cdot g(u, v) du dv$$

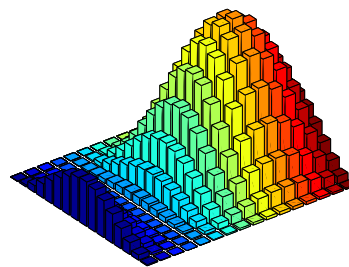
et où $\Pi(x, y)$ est la fonction porte définie comme :

$$\Pi(x, y) = \begin{cases} 1 & \text{si } (x, y) \in \left[-\frac{1}{2}, \frac{1}{2}\right] \times \left[-\frac{1}{2}, \frac{1}{2}\right] \\ 0 & \text{sinon} \end{cases}$$

Résolution Comme nous considérons un pavage jointif, les pas de discrétisation sont égaux aux dimensions d'une cellule. Plus les pas de discrétisation Δ_u et Δ_v sont petits, plus l'approximation de la fonction sera bonne (c.f. figure 2.4).



(a) Échantillonnage de f avec un pas de 5 unités



(b) Échantillonnage de f avec un pas de 2 unités

Figure 2.4.: Différence d'approximation d'une fonction en fonction du pas d'échantillonnage.

On définit la résolution r comme étant le nombre de pixels par unité de longueur. La résolution s'apparente à une fréquence spatiale que l'on peut exprimer en fonction des pas d'échantillonnage, $r = \frac{1}{\Delta}$.

Le théorème de Shannon-Nyquist [Shannon, 1948] fixe un cadre pour le choix de la résolution. En effet, si on considère f comme un signal bi-dimensionnel périodique, ce théorème précise qu'il n'y a pas de perte d'information durant l'opération d'échantillonnage si et seulement si la fréquence d'échantillonnage est au moins deux fois

supérieure à la fréquence maximale présente dans le signal f . Dans le cas contraire, il y a un phénomène de repliement spectral qui engendre une perte d'information. Ce phénomène est tout à fait visible en 2 dimensions comme le montre la figure 2.5.

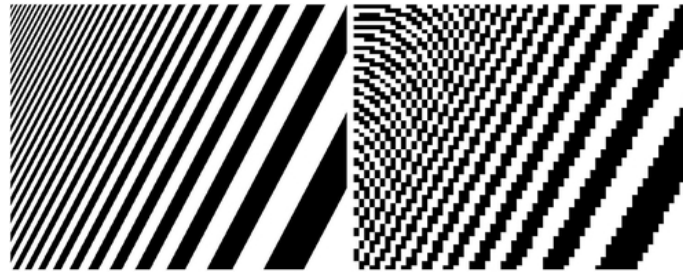


Figure 2.5.: Phénomène de repliement spectral sur une image : à gauche sans repliement, à droite avec un phénomène de repliement dû à un sous-échantillonnage, particulièrement visible dans le coin en haut à gauche.

Soit $f(x_1, x_2)$ un signal 2D et soit $F(f_1, f_2)$ sa transformée de Fourier. On considère que f est à bande limitée dans chacune de ses dimensions (en fréquence spatiale). Cela implique que F est à support borné. La résolution (r_u, r_v) doit être au moins égale à 2 fois (f_{1max}, f_{2max}) .

Étant donné que la fonction f évolue dans le temps et n'a pas de propriété de périodicité particulière, on va fixer les fréquences échantillonnages de façon à faire un compromis entre la quantité d'information et la fréquence maximale observable. Si le support spectral de F n'est pas borné, alors il est tronqué à partir d'une certaine valeur, ce qui implique une perte d'information. Comme le montre la figure 2.6, on sera par exemple limité pour observer les barreaux d'une barrière si on échantillonne avec une période spatiale plus grande que la dimension d'un barreau.

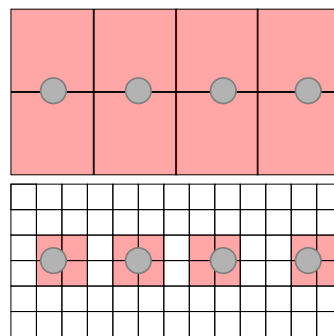


Figure 2.6.: Modélisation des barreaux d'une barrière en vue de dessus (cercle gris) à l'aide d'une grille, les cellules rouges et blanches indiquent respectivement la présence ou indiquent l'absence d'un barreau. La figure du haut montre que les barreaux ne sont pas identifiables sur la grille à cause d'une résolution trop faible. Sur la figure du bas, par contre, les barreaux sont identifiables car la résolution est suffisante.

Échantillonnage temporel Lorsque la fonction f n'est pas constante dans le temps, on réalise une approximation à différents instants d'échantillonnage (temporel). On considère alors que toutes les cellules sont échantillonnées simultanément (indépendance entre les échantillonnages en (u, v) et en t). De la même façon que pour l'échantillonnage spatial, la cadence de rafraîchissement temporel est déterminante par rapport à la dynamique de la fonction.

2.2.2. Référencement spatial des grilles métriques

Après avoir défini les notations et les concepts de base des grilles spatiales, nous décrivons les notions principales relatives aux grilles métriques qui seront nécessaires dans la suite de la thèse.

Repères Afin de faciliter l'expression et le calcul des opérations géométriques, nous utilisons le système de coordonnées homogènes. Les principaux repères attachés aux grilles et décrits dans ce paragraphe sont donnés dans ce système de coordonnées.

Système de coordonnées homogènes Le système de coordonnées homogènes permet de représenter toute application affine appliquée à un point par une seule matrice de transformation. Dans ce système, on ajoute une composante W non nulle aux autres coordonnées déjà présentes. On passe donc d'un espace vectoriel (EV) de dimension 2 à un EV de dimension 3 ou d'un EV de dimension 3 à un EV de dimension 4. Un point de coordonnées cartésiennes (x, y, z) aura plusieurs représentants en coordonnées homogènes, qui seront tous colinéaires entre eux. De manière générale, on préfère utiliser le représentant avec $W = 1$, car les coordonnées x, y, z sont alors identiques aux coordonnées cartésiennes.

- Un représentant en coordonnées homogènes d'un point (x, y, z) est alors $(x, y, z, 1)$.
- D'autre part, pour un point de coordonnées homogènes (x, y, z, W) , $W \neq 0$, on calcule les coordonnées cartésiennes correspondantes comme $\left(\frac{x}{W}, \frac{y}{W}, \frac{z}{W}\right)$.

Transformation homogène On considère une application affine f de $\mathbb{R}^3 \rightarrow \mathbb{R}^3$. Cette application est définie par sa matrice linéaire et son vecteur affine.

Pour tout point P de \mathbb{R}^3 , on applique la fonction f de la façon suivante :

$$f(P) = \begin{bmatrix} a_{x,x} & a_{x,y} & a_{x,z} \\ a_{y,x} & a_{y,y} & a_{y,z} \\ a_{z,x} & a_{z,y} & a_{z,z} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \quad (2.9)$$

Pour appliquer cette application en coordonnées homogènes, il faut construire la matrice de transformation homogène comme le montre l'équation 2.10.

$$f(P) = \begin{bmatrix} a_{x,x} & a_{x,y} & a_{x,z} & t_x \\ a_{y,x} & a_{y,y} & a_{y,z} & t_y \\ a_{z,x} & a_{z,y} & a_{z,z} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.10)$$

Tout ce qui a été défini ici pour un espace de dimension 3 est défini de façon similaire pour un espace de dimension 2. Dans la suite de ce manuscrit, les transformations géométriques seront réalisées par le biais de transformations homogènes.

Repères robotiques Considérons un robot mobile qui évolue à l'intérieur du domaine \mathcal{D} fini d'un monde plan (figure 2.7). Ce domaine est défini comme un espace euclidien à deux dimensions de \mathbb{E}^2 . Cet espace est muni du repère de référence \mathcal{R}_0 fixe.

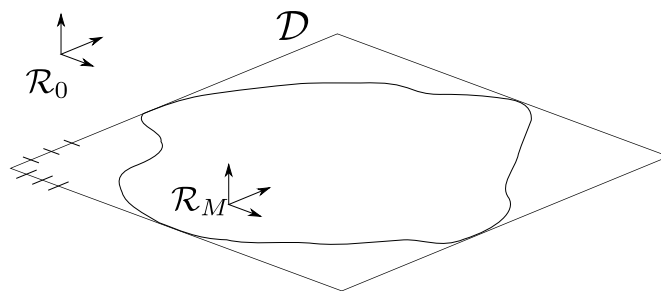


Figure 2.7.: Repère mobile dans un monde plan. La surface irrégulière matérialise le domaine d'évolution du robot inscrit dans le domaine \mathcal{D} .

Soit \mathcal{R}_M le repère attaché à un point M défini comme le centre du robot mobile. Ce repère sera utilisé pour référencer les grilles métriques par la suite.

En coordonnées homogènes, le point dans le repère \mathcal{R}_0 s'écrit² :

$${}^0M = \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix}$$

De la même manière, un point P , de l'espace \mathbb{E}^2 , a pour coordonnées (voir figure 2.8) :

2. Un exposant en haut à gauche indique le repère dans lequel sont exprimées les coordonnées.

$${}^0P = \begin{bmatrix} {}^0x_p \\ {}^0y_p \\ 1 \end{bmatrix} \quad {}^M P = \begin{bmatrix} {}^Mx_p \\ {}^My_p \\ 1 \end{bmatrix}$$

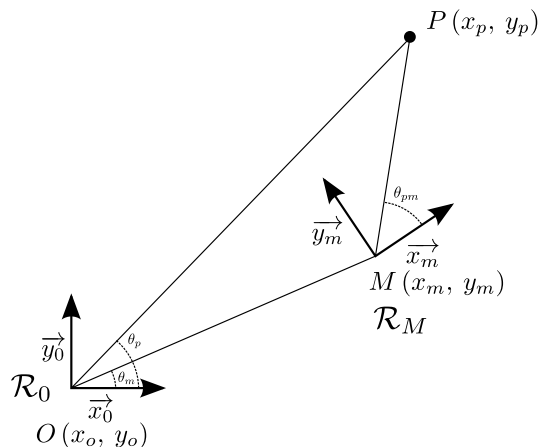


Figure 2.8.: Définitions des repères.

On définit la transformation homogène 0H_M de changement de repère entre \mathcal{R}_M et \mathcal{R}_0 décrite par l'équation 2.11.

$$\forall P \in \mathbb{E}^2, {}^0P = {}^0H_M \cdot {}^M P \quad (2.11)$$

avec :

$${}^0H_M = \begin{bmatrix} \cos({}^0\theta_m) & -\sin({}^0\theta_m) & {}^0x_m \\ \sin({}^0\theta_m) & \cos({}^0\theta_m) & {}^0y_m \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

Enfin, comme le robot mobile se déplace au cours du temps, le repère \mathcal{R}_M est un repère mobile qui évolue. En l'occurrence, les variables 0x_m , 0y_m et ${}^0\theta_m$ dépendent du temps.

Ces deux repères sont nécessaires pour la définition des grilles 2D spatio-référencées. Par la suite, nous définirons d'autres repères, notamment liés aux différents capteurs nous utiliserons alors des notations similaires.

Définition d'une grille de façon analogue au traitement d'image On considère un domaine rectangulaire $\mathcal{D} = [\underline{x} \ \bar{x}] \times [\underline{y} \ \bar{y}] \subset \mathbb{E}^2$, muni d'un repère de localisation \mathcal{R}_M qui permet de localiser toutes les cellules dans l'espace. Soit la fonction $f(x, y) : \mathcal{D} \rightarrow [0; 1]$ que l'on veut approximer par une méthode discrète. A partir du cadre précédent, on peut définir une grille³ G^A dans le but d'approximer f . Cette grille G^A est un partitionnement de \mathcal{D} en un ensemble de cellules G_{ij}^A rectangulaires de dimension $\Delta_x \times \Delta_y$ arrangées par convention de façon matricielle suivant les axes du repère \mathcal{R}_M :

$$G^A = \left\{ G_{ij}^A \mid \forall (i, j) \in \mathbb{Z}, \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} \in \mathcal{D} \right\} \quad (2.13)$$

où chaque G_{ij}^A correspond à la cellule d'indices (i, j) .

La grille G^A est définie par ses caractéristiques intrinsèques :

- un repère image \mathcal{R}_I qui permet d'indiquer les cellules comme indiqué dans la section 2.2.1,
- la position relative par rapport à \mathcal{R}_I du centre M du repère $\mathcal{R}_M (M, \vec{x}, \vec{y})$ de référencement spatial,
- un domaine spatial \mathcal{D}^A correspondant au rectangle de travail selon les axes du repère \mathcal{R}_M ,
- une résolution $\Delta^A (\Delta_u^A, \Delta_v^A)$ définie comme étant la dimension d'une cellule.

Le repère \mathcal{R}_M est par convention orienté comme indiqué sur la figure 2.9. Ainsi, si $({}^0x, {}^0y)$ sont les coordonnées dans \mathcal{R}_M du point I échantillonné d'indices discrets $(0, 0)$, chaque point P_{ij} en (i, j) du maillage peut être repéré soit dans \mathcal{R}_I soit dans \mathcal{R}_M en coordonnées homogènes :

$${}^I P_{ij} = \begin{bmatrix} i \cdot \Delta_u \\ j \cdot \Delta_v \\ 1 \end{bmatrix} \quad {}^M P_{ij} = \begin{bmatrix} {}^0x - i \cdot \Delta_u \\ {}^0y - j \cdot \Delta_v \\ 1 \end{bmatrix}.$$

3. On place en exposant en haut à droite la référence de la grille.

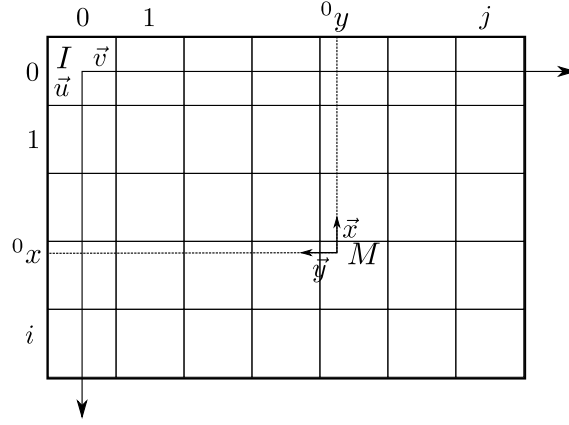


Figure 2.9.: Correspondance entre un repère habituel d'une matrice image et le repère \mathcal{R}_M de la grille.

Pour une cellule G_{ij}^A d'indices (i, j) de la grille A , on note

- C_{ij}^A le centre⁴,
- S_{ij}^A la surface, $S_{ij}^A = \left[C_{ij}^A \cdot x - \frac{\Delta_u^A}{2}, C_{ij}^A \cdot x + \frac{\Delta_u^A}{2} \right] \times \left[C_{ij}^A \cdot y - \frac{\Delta_v^A}{2}, C_{ij}^A \cdot y + \frac{\Delta_v^A}{2} \right]$,
- I_{ij}^A le contenu.

De même, pour la grille G^A , on note

- $C^A = \{C_{ij}^A\}$ l'ensemble de tous les centres,
- $S^A = \{S_{ij}^A\}$ l'ensemble de toutes les surfaces,
- $I^A = \{I_{ij}^A\}$ l'ensemble de tous les contenus.

Enfin, on définit la notion d'appartenance d'un point P à une cellule $P \in S_{ij}^A$ ou par abus de notation $P \in G_{ij}^A$ de la façon suivante :

$$\forall P = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{E}^2, P \in S_{ij}^A \iff \begin{matrix} {}^0x - (i + \frac{1}{2}) \cdot \Delta_u < x \leq {}^0x - (i - \frac{1}{2}) \cdot \Delta_u \\ {}^0y - (j + \frac{1}{2}) \cdot \Delta_v < y \leq {}^0y - (j - \frac{1}{2}) \cdot \Delta_v \end{matrix}, \quad (2.14)$$

ce qui peut aussi s'exprimer par

$$\forall P = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{E}^2, P \in G_{ij}^A \iff \begin{matrix} \text{round} \left(\begin{matrix} M \\ x \end{matrix} \right) = {}^0x - i \cdot \Delta_u \\ \text{round} \left(\begin{matrix} M \\ y \end{matrix} \right) = {}^0y - j \cdot \Delta_v \end{matrix}. \quad (2.15)$$

où $\text{round}()$ est la fonction arrondie à l'entier le plus proche.

Par extension des notations précédentes, on note pour un point P de coordonnées $\begin{bmatrix} x & y & 1 \end{bmatrix}^T$

4. Il ne faut pas confondre le centre C_{ij}^A avec la cellule G_{ij}^A

- $C^A(x, y)$ le centre de la cellule contenant P ,
- $S^A(x, y)$ la surface de la cellule contenant P ,
- $I^A(x, y)$ le contenu de la cellule contenant P .

Grille polaire On peut définir une grille dans un repère polaire $\mathcal{R}_O(O, \vec{r}, \vec{\theta})$. Dans ce cas, on discrétise toujours l'espace en utilisant des cellules rectangulaires, en distance suivant un axe et en angle suivant l'autre. On définit alors les paramètres de discrétisation Δ_r et Δ_θ .

$$\mathbf{I}_{P_{ij}} = \begin{bmatrix} i \cdot \Delta_r \\ j \cdot \Delta_\theta \end{bmatrix} \quad \mathbf{M}_{P_{ij}} = \begin{bmatrix} {}^0r - i \cdot \Delta_r \\ {}^0\theta - j \cdot \Delta_\theta \end{bmatrix}.$$

La grille polaire se représente de la même façon dans son repère polaire que les grilles cartésiennes : une cellule correspond à un pavé de $\Delta_r \times \Delta_\theta$.

En repère cartésien, chaque cellule correspond à une section d'anneau et la condition d'appartenance d'un point à la cellule de centre le plus proche n'est alors plus respectée.

Les grilles métriques spatio-référencées vont servir de support pour modéliser l'environnement perçu par le robot. Des opérations géométriques ou intrinsèques sont nécessaires pour propager l'information contenue dans ces grilles. Une description de ces transformations est donnée dans l'annexe ??.

2.3. Grille d'occupation pour la perception

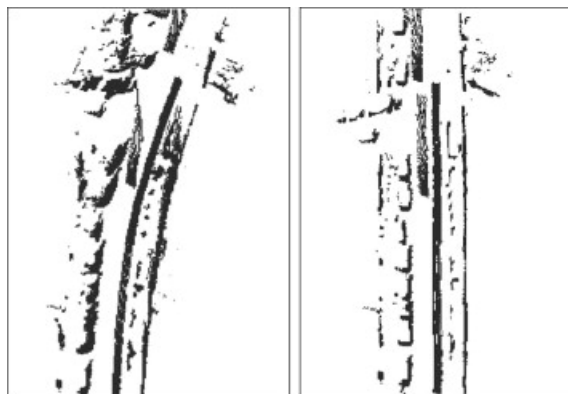
2.3.1. Grille d'occupation

La grille d'occupation est un moyen de modéliser l'occupation de l'environnement de façon discrète dans l'espace. Nous allons dans un premier temps considérer une grille d'occupation déterministe mais nous verrons par la suite que ce modèle est souvent couplé à un formalisme approprié pour prendre en compte les incertitudes.

On s'intéresse à représenter la fonction occupation de l'espace $\mathcal{O}(P, t)$ à un instant t . Cette fonction est une fonction spatiale à 2 dimensions référencée dans l'espace et qui évolue au cours du temps. Pour la représenter, nous allons utiliser le cadre des grilles métriques spatio-référencées.

On définit la grille d'occupation $G^\mathcal{O}$ de référentiel \mathcal{R} , de domaine \mathcal{D} et de résolution Δ , comme une grille dans laquelle la fonction estimée est la fonction d'occupation $\mathcal{O}(x, y, t)$. Chaque cellule $G_{ij}^\mathcal{O}$ de la grille contient une valeur binaire signifiant l'occupation de la cellule que l'on notera par extension $\mathcal{O}(G_{ij}^\mathcal{O}, t)$.

La figure 2.10 montre différents exemples de grilles d'occupation utilisées en robotique.



(a) Grille d'occupation issue de [Vu *et al.*, 2011]



(b) Grille d'occupation issue de [Thrun, 2000]

Figure 2.10.: Exemples de grilles d'occupation

Ce modèle est un modèle géométrique très efficace pour représenter l'environnement, car il ne fait pas d'hypothèse sur la forme des éléments qui composent l'environnement. Cependant, il pose deux principaux problèmes qu'il faut considérer. D'une part, la discrétisation engendre un problème d'approximation locale de la fonction d'occupation. D'autre part, la fonction d'occupation étant estimée, il faut prendre en compte les incertitudes induites par le processus d'estimation.

Occupation d'une cellule Si on utilise le modèle de grille pour approximer la fonction d'occupation, on considère l'occupation d'un ensemble fini de cellules au lieu de considérer l'occupation d'une infinité de points. Chaque cellule a une surface

qui est référencée dans le monde physique et on généralise l'information contenue⁵ dans une cellule à l'ensemble des points qu'elle contient. Pour cela, on définit une fonction F qui, à partir de l'ensemble des occupations de tous les points P d'une cellule de surface $S_{ij}^{\mathcal{O}}$, fournit une occupation de cette cellule, comme le montre l'équation 2.16 :

$$I_{ij}^{\mathcal{O}} = \mathcal{O}(G_{ij}^{\mathcal{O}}, t) = F\left(\left\{\mathcal{O}(P, t) \mid P \in S_{ij}^{\mathcal{O}}\right\}\right) \quad (2.16)$$

Chaque point a une valeur d'occupation qui peut être différente au sein de la même cellule. On peut se retrouver dans 3 cas possibles :

- la cellule est complètement vide : tous ses points sont libres $\forall P \in S_{ij}^{\mathcal{O}}, \mathcal{O}(P, t) = 0$. Dans ce cas, $F(\{0, \dots, 0\}) = 0$.
- la cellule est complètement occupée : tous ses points sont occupés $\forall P \in S_{ij}^{\mathcal{O}}, \mathcal{O}(P, t) = 1$. Dans ce cas, $F(\{1, \dots, 1\}) = 1$.
- la cellule est partiellement occupée : $\exists P_1, P_2 \in S_{ij}^{\mathcal{O}}, \mathcal{O}(P_1, t) = 0 \& \mathcal{O}(P_2, t) = 1$. On se trouve ici dans une situation qui ne nous permet pas de décrire la situation à l'aide d'une cellule sans faire une erreur d'approximation. On peut alors considérer plusieurs stratégies :
 - une cellule est libre si elle contient au moins un point libre, c'est-à-dire $\mathcal{O}(G_{ij}^{\mathcal{O}}) = 0 \Leftrightarrow \exists P \in S_{ij}^{\mathcal{O}}, \mathcal{O}(P) = 0$
 - une cellule est occupée si elle contient au moins un point occupé, c'est-à-dire $\mathcal{O}(G_{ij}^{\mathcal{O}}) = 1 \Leftrightarrow \exists P \in S_{ij}^{\mathcal{O}}, \mathcal{O}(P) = 1$
 - une cellule est occupée si elle contient plus de points occupés que de points libres, c'est-à-dire $\mathcal{O}(G_{ij}^{\mathcal{O}}) = 1 \Leftrightarrow \int_{S_{ij}^{\mathcal{O}}} \mathcal{O}(P) dP > \int_{S_{ij}^{\mathcal{F}}} \mathcal{F}(P) dP$.

Généralement, pour la navigation, on préfère utiliser la seconde stratégie qui est plus prudente, puisque l'on veut que le robot évolue uniquement dans les cellules entièrement libres. Néanmoins, cette stratégie restreint l'espace navigable d'autant plus que les cellules sont grandes.

Informations incertaines Le principe de discrétisation introduit des erreurs d'approximation spatiale de la fonction d'occupation (voir section précédente) mais ce n'est pas la seule source d'erreur.

A cause des incomplétudes des mesures, la fonction d'occupation ne peut pas être estimée en tout point. De plus, aux endroits où elle est estimée, il faut considérer les incertitudes de mesure. Si on ne peut pas connaître avec certitude l'état d'une cellule $\mathcal{O}(G_{ij}^{\mathcal{O}}, t)$, on va alors estimer la confiance Bel que l'on a sur le fait que la cellule est dans un état donné, par exemple : $Bel(\mathcal{O}(G_{ij}^{\mathcal{O}}, t) = 1)$. Au final, la grille ne sera plus binaire, chaque cellule va contenir différents paramètres numériques qui permettent de traduire cette confiance. Une fonction de décision sera ensuite utilisée pour reconstruire une grille binaire. Différents formalismes existent pour représenter les données incertaines et les principaux seront détaillés dans le chapitre 3.

5. On rappelle que la notation $I_{ij}^{\mathcal{O}}$ désigne le contenu de la cellule d'indices (i, j) de la grille \mathcal{O} .

2.3.2. Perception avec des grilles

L'objectif est de construire une grille d'occupation à partir d'un ensemble de mesures incomplètes et incertaines dans un environnement inconnu et dynamique. Le robot est équipé de capteurs qui fournissent des séries de mesures extéroceptives sur son environnement au cours du temps, que l'on nommera de façon générale *scan* et sera noté Z_t . Ces scans peuvent être des ensembles de points, des images, ou d'autres types de données acquises dans un temps relativement petit par rapport à la dynamique. On considérera que les capteurs fonctionnent à une cadence fixe suffisamment rapide par rapport à la dynamique de la scène et on s'intéresse à construire une grille à chaque pas de temps où l'on disposera d'une mesure. De plus, on suppose qu'un système de positionnement est capable de fournir à chaque instant la pose du véhicule (locale ou globale) qui sera dénoté q_t . Enfin, si des cartes sont disponibles, on peut intégrer l'information cartographique dans le système de perception. La figure 2.11 montre l'architecture globale du système.

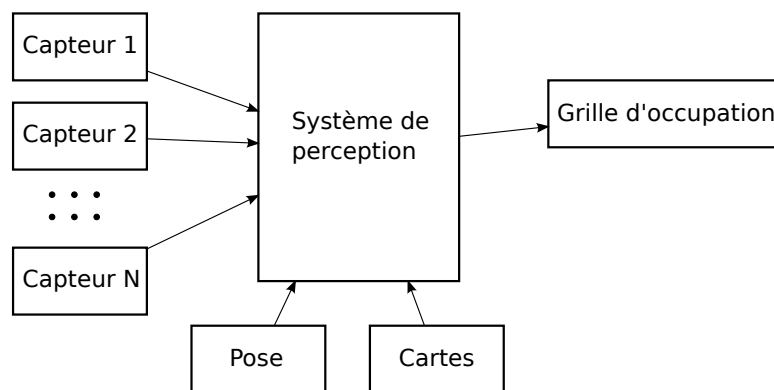


Figure 2.11.: Interactions du système de fusion.

Modèle de capteur Les différents capteurs extéroceptifs (caméra, RADAR, LIDAR, etc...) utilisés fournissent des informations sur l'espace autour du robot. Ces capteurs sont des dispositifs qui permettent de traduire des propriétés physiques de l'environnement en données exploitables. Comme on s'intéresse à l'occupation de l'espace, il faut être capable de traduire les grandeurs fournies par les capteurs en information sur la confiance dans l'occupation. Pour cela, on définit une fonction que l'on nommera le *modèle de capteur*. Le modèle peut être déterminé empiriquement (statistiquement) ou au travers d'un modèle physique.

Par exemple, si on considère un télémètre laser 1D :

– Modèle direct :

Sachant qu'il existe un obstacle à une distance d , quelle est la vraisemblance d'avoir un écho laser à une distance x , pour $x \in [0 + \infty]$? Avec ce genre de modèle, on peut prédire la mesure sachant l'état du système.

– Modèle inverse :

Sachant que le capteur a retourné un écho à une distance x , quelle est la confiance qu'il y ait un obstacle à une distance d , pour $d \in [0 + \infty]$? Dans certains cas, ce modèle peut être défini par l'inversion du modèle direct du capteur.

Lorsqu'on utilise des grilles pour représenter l'environnement, le modèle de capteur est une fonction qui permet d'établir la relation entre l'état des cellules et les mesures du capteur. Suivant la méthode utilisée pour construire la grille, méthode par fusion ou par inférence, il faut utiliser soit le modèle de capteur direct soit le modèle inverse. Il existe un grand nombre de modèles de capteur dans la littérature [Thrun, 2003, Elfes, 1989c, Durrant-whyte, 1988].

Dans notre cas, on mettra en place un modèle de capteur inverse sur grille pour faire le lien entre l'occupation d'une cellule et les données du scan.

Approche grille instantanée La solution la plus simple consiste à estimer une nouvelle grille d'occupation à chaque fois qu'un nouveau scan Z_t est disponible. La grille construite est nommée ScanGrid (SG). Comme le montre la figure 2.12, chaque SG est indépendante des autres SG .

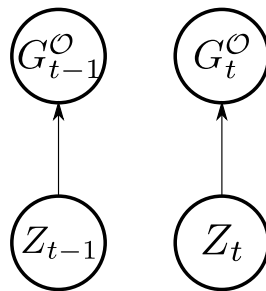


Figure 2.12.: Approche ScanGrid : la grille courante ne dépend que du scan courant.

Les inconvénients de cette approche sont :

- l'incomplétude de la grille,
- la forte sensibilité aux variations brusques et notamment au bruit de mesure,
- la confiance de la grille est difficile à quantifier (intégrité),
- la difficulté pour estimer la dynamique de la scène avec un capteur n'utilisant pas l'effet Doppler.
- la non prise en compte de plusieurs capteurs,

Certaines approches (par exemple [Himmelsbach *et al.*, 2008]) qui utilisent des capteurs très performants peuvent baser la perception sur une grille instantanée. Dans notre cas, à cause des limitations de notre capteur et des inconvénients cités ci-dessus, cette méthode n'est pas suffisante pour faire de la navigation autonome dans les conditions qui nous intéressent.

2.4. Construction par fusion temporelle d'une grille

Pour améliorer la perception, une solution est d'utiliser la grille précédente pour calculer la nouvelle grille. Cela permet de prendre en compte les redondances ou de détecter des conflits mais aussi d'avoir une information plus complète. Cela revient à réaliser un filtrage temporel séquentiel du type prédicteur - correcteur, comme le montre la figure 2.13.

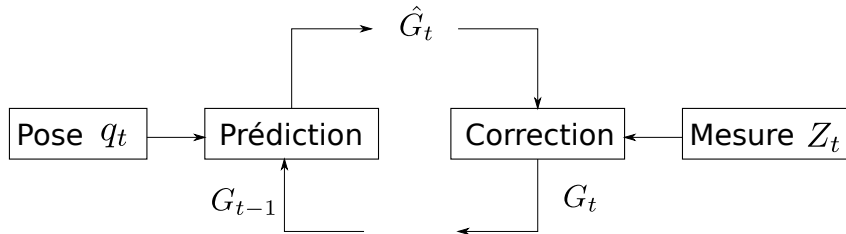


Figure 2.13.: Principe prédicteur - correcteur.

2.4.1. Principe

On construit une ScanGrid à partir du scan courant qui est ensuite fusionné dans une MapGrid (MG). Cette dernière est propagée au cours du temps. Ce schéma est décrit dans la figure 2.14.

La MapGrid est une grille qui sert de mémoire pour l'état des cellules entre les différents instants. Elle réalise donc d'une certaine façon une cartographie dynamique de l'environnement. Pour pouvoir fusionner la SG et la MG , il faut que les deux grilles soit cohérentes spatialement et temporellement. A chaque fois que l'on dispose d'une nouvelle SG , il faut faire une prédiction de la MG à la nouvelle position et au nouveau temps. D'autre part, la MG peut être exprimée soit dans un repère fixe lié au monde et on parle d'approche halo-centrée, soit dans un repère lié au véhicule et, dans ce cas, on parle d'approche égo-centrée.

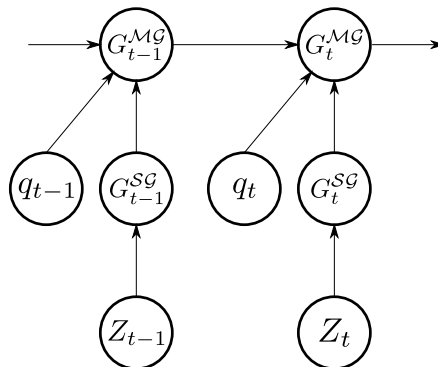


Figure 2.14.: Approche par mise à jour séquentielle.

2.4.2. Prédiction et rémanence de l'information dans les grilles

Quand une nouvelle mesure est disponible, une prédiction de la grille est réalisée au temps de la mesure. Puisque la scène est dynamique et que le véhicule se déplace, il faut être capable de réaliser une prédiction de la scène en tenant compte de tous ces éléments. Une mauvaise prédiction peut entraîner une inconsistance la *MG*.

La phase de prédiction se déroule en deux étapes : la compensation du mouvement propre du véhicule (égo-motion) et la prédiction de la dynamique de la scène.

La prédiction de la position du véhicule dépend du repère de référence et donc du mode halo-centré ou égo-centré. Pour prédire la grille, on doit connaître la pose du véhicule à l'instant de chaque scan. On utilise ici un système de positionnement précis pour que la cohérence spatiale soit respectée et on réalise un déplacement de grille pour compenser le mouvement.

Comme on observe un milieu dynamique, au bout d'un certain laps de temps, l'information de la MapGrid précédente n'est plus valable du fait que la scène a évolué entre temps. Il faut donc faire une prédiction de la grille. Cette prédiction peut se faire de deux façons.

- Si l'on estime les paramètres dynamiques de la scène, on peut prédire les déplacements dans la nouvelle grille. La détermination de ces paramètres passe généralement par une étape d'association de cellules entre la grille courante et la grille précédente. La précision de ce type de méthode dépend principalement de la qualité d'association et du modèle dynamique utilisé pour prédire la grille.
- A défaut de prédire la scène, une autre solution est de considérer la grille fixe et de prédire la confiance de la grille. Dans ce cas, on considère que plus le temps passe depuis la dernière correction, moins la grille est fiable. Cette solution permet si les capteurs sont suffisamment rapides par rapport à la dynamique de la scène de réaliser un filtrage temporel pour les zones auscultées. Les zones n'étant plus observées sont alors rapidement oubliées. C'est cette dernière approche qui a été retenue dans cette thèse.

2.4.3. Correction de la grille et détection des cellules mobiles

Une fois l'étape de prédiction réalisée, suit la phase de correction. Comme on a vu précédemment, on construit une *SG* à partir du scan courant que l'on transforme pour la rendre compatible avec la *MG*. Enfin, on réalise la fusion de ces deux grilles. L'opérateur de fusion dépend du formalisme de représentation des incertitudes et du contenu des cellules. Cette partie sera développée dans le chapitre 3.

2.4.4. Architecture du système en fonction du repère de référence

2.4.4.1. Approche halo-centrée

Une grille halo-centrée est liée à un point fixe de l'environnement : elle ne se déplace pas. Le robot mobile évolue dans la grille comme n'importe quel autre élément (voir figure 2.15). Cette configuration est plutôt utilisée lorsque l'on veut faire de la cartographie ou lorsqu'on développe des approches de type SLAM.

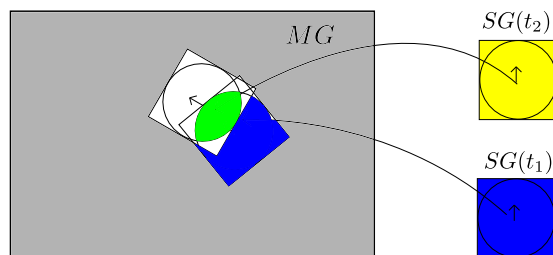


Figure 2.15.: Grille halo-centrée : la grille principale est fixe.

La figure 2.16 illustre cette approche. Une fois la SG construite, elle est transformée dans le repère de la MG . La MG de l'instant précédent est utilisée pour faire la prédiction, qui n'est ici qu'une prédiction de dynamique qui ne concerne pas l'ego-motion. Puis, les deux grilles sont fusionnées afin d'intégrer l'information de la nouvelle SG . Finalement, on peut, si nécessaire, extraire une portion de la MG autour du véhicule et la retransformer dans le repère véhicule pour la navigation.

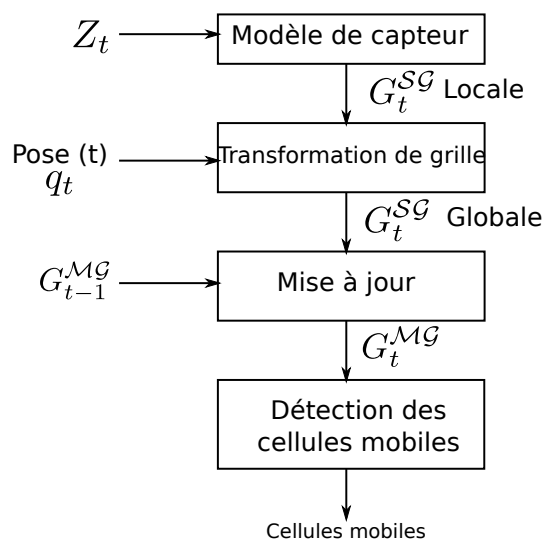


Figure 2.16.: Architecture du système de fusion halo-centrée.

L'avantage de cette méthode est qu'elle ne nécessite pas beaucoup d'opérations sur les grilles : une seule transformation (et donc interpolation) est nécessaire pour chaque SG . D'autre part, on peut obtenir une carte de l'environnement globale à l'issue d'un déplacement.

Cependant, cette méthode a plusieurs inconvénients. La MG est de dimension finie ce qui limite la zone d'évolution du robot. Pour remédier à cela, on peut mettre en place une allocation dynamique de la taille et de la position de la grille fixe autour du robot mobile, mais cela complexifie le système. Enfin, pour réaliser la tâche de navigation, il est généralement nécessaire d'avoir la grille dans un repère local, il faut donc effectuer une transformation pour repasser dans le repère véhicule à chaque instant si nécessaire.

2.4.4.2. Approche égo-centrée

Dans le cas égo-centré, la grille filtrée (appelée précédemment MG) est mobile et liée au robot (et en général centrée sur lui), cf. figure 2.17. Dans ce cas, on l'appellera carte locale dynamique (CLD). Cette grille se déplace avec le robot, ce qui nécessite de la recalculer en intégralité à chaque déplacement du robot. Cette grille a l'avantage de couvrir toujours la même zone autour du robot et ne limite pas son domaine d'évolution.

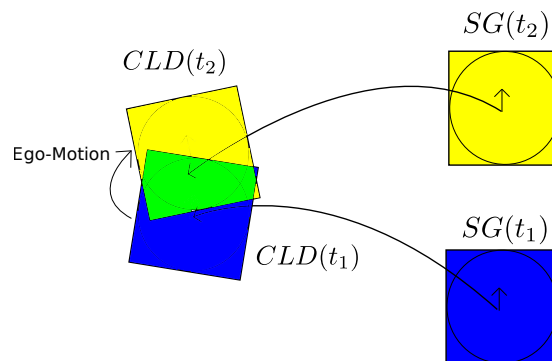


Figure 2.17.: Grille égo-centrée : la grille principale est centrée sur le robot et se déplace avec celui-ci. La zone jaune correspond à la partie superposée entre la grille courante et la grille précédente, la zone verte est une nouvelle zone qui entre dans la grille, la zone bleue quitte la grille et est oubliée.

La figure 2.18 montre l'architecture du système égo-centré. Dans ce cas, c'est la CLD qui subit la transformation spatiale pour se repositionner dans la repère courant du véhicule. Les cellules qui n'avaient pas d'antécédents dans la précédente CLD , car en dehors de la grille, sont affectées à l'inconnu. La CLD est ensuite atténuée par prédiction. Puis, on peut faire la fusion des deux grilles dans le repère véhicule.

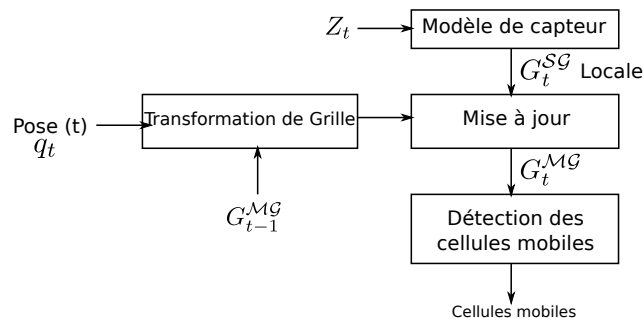


Figure 2.18.: Architecture du système de fusion égo-centrée.

L'inconvénient de cette méthode réside dans le fait que la mise à jour de la grille nécessite de faire une interpolation de la *CLD* et ceci de façon séquentielle à chaque pas de temps. Ces interpolations provoquent une perte d'information dans la grille comme décrit dans l'annexe ???. Pour éviter ce phénomène, une solution consisterait à conserver un historique des *SG* précédentes de façon à refaire une fusion complète dans la *CLD* à chaque pas de temps (voir figure 2.19). Mais ce type d'approche aurait un coût calculatoire et un besoin en mémoire importants.

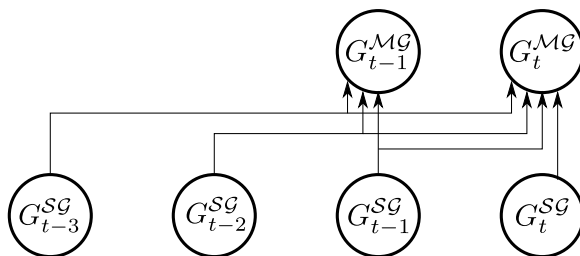


Figure 2.19.: Méthode égo-centrée par fusion complète.

2.5. Conclusion

Dans ce chapitre, nous avons introduit un cadre de traitement de grille 2D basé sur celui du traitement d'image numérique pour traiter le problème d'estimation de fonction définie sur un espace 2D spatio-référencé. Ce cadre a été utilisé pour la perception de l'espace navigable d'un véhicule en milieu urbain. Cette méthode repose sur l'utilisation du formalisme de grille métrique pour construire des grilles d'occupation. L'architecture de mise à jour de la grille de perception suit un schéma prédicteur - correcteur et nécessite d'appliquer les outils de transformation et de fusion de grille. Selon le repère de référence, les cas halo-centré avec une grille fixe et égo-centré avec une grille mobile qui se déplace avec le véhicule ont tous deux été étudiés.

3. Formalismes de gestion d'incertitude pour les grilles de perception

Sommaire

3.1. Introduction	41
3.2. Grille d'accumulation	42
3.3. Grille d'occupation probabiliste	46
3.4. Grille d'occupation évidentielle	54
3.5. Traitements particuliers possibles dans les grilles évidentielles	68
3.6. Comparaison des cadres probabiliste et évidentiel	77
3.7. Conclusion	86

3.1. Introduction

Dans les chapitres précédents, nous avons introduit le principe de grille d'occupation pour la navigation. Ces grilles permettent d'estimer l'occupation de l'espace avec certaines incertitudes. La façon de représenter ces incertitudes a d'importantes conséquences sur la manière de traiter les informations contenues dans la grille et ensuite de raisonner sur ces données. Il faudra, in fine, être capable de prendre une décision et fournir pour chaque cellule une information binaire, occupé ou libre.

Dans cette partie, nous allons introduire trois formalismes :

- le formalisme d'accumulation, très simple, inspiré du principe du vote,
- le formalisme bayésien repose sur la théorie des probabilités et c'est formalisme le plus utilisé en robotique,
- le cadre évidentiel, qui découle de la théorie de Dempster-Shafer et du Transferable Belief Model de Smets (TBM) [Dempster, 1968, Shafer, 1976, Smets et Kennes, 1994] qui est une généralisation des probabilités.

C'est ce dernier formalisme que nous avons choisi dans cette thèse pour modéliser et gérer les incertitudes. Les raisons de ce choix seront exposées au travers d'un descriptif détaillé de ce formalisme et de ses outils ainsi qu'au travers d'une comparaison avec le cadre probabiliste. Nous verrons en particulier que ce formalisme

est doté d'un ensemble d'opérateurs de fusion très divers qui possèdent chacun leurs intérêts.

3.2. Grille d'accumulation

3.2.1. Principe

Inspiré du principe du vote, les grilles d'accumulation utilisent un principe fréquentiste : plus une cellule est vue comme occupée, plus elle a de chances d'être occupée. Bien qu'il soit assez peu utilisé, on peut trouver des contributions originales utilisant ce formalisme. Dans [Borenstein et Koren, 1991], les grilles d'accumulation sont utilisées pour la navigation expérimentale d'un robot d'intérieur à l'aide d'un sonar. Dans [Xie *et al.*, 2010], les grilles d'accumulation sont utilisées pour réaliser la cartographie d'un environnement extérieur en utilisant un télémètre laser à balayage. La localisation est assurée en ligne par une mise en correspondance des grilles en utilisant des grilles multi-résolutions.

Dans cette approche, le contenu I_{ij} de chaque cellule est un niveau d'accumulation appelé Al (Accumulation level). Cet Al est un entier qui représente le nombre d'occurrences où la cellule a été observée comme occupée. Par exemple, pour une cellule G_{ij} d'une grille d'occupation statique, l' Al peut être calculé en sommant les N mesures¹ P_k dont les coordonnées appartiennent à la cellule :

$$I_{ij} = Al_{ij} = \sum_{k=1}^N \mathcal{O}(G_{ij}|P_k)$$

$$\text{où, } \mathcal{O}(G_{ij}|P_k) = \begin{cases} 1 & \text{si } P_k \in S_{ij} \\ 0 & \text{sinon} \end{cases} \quad (3.1)$$

Dans cette expression, $\mathcal{O}(G_{ij}|P_k)$ correspond à la fonction d'occupation conditionnée par la mesure P_k .

3.2.2. Prise de décision

Cette méthode utilise la redondance d'information pour tenter d'estimer l'état O . Pour chaque cellule, on prend une décision en comparant le niveau d'accumulation à un seuil th .

$$\begin{cases} \text{si } Al_{ij} < th & , \text{ alors } G_{ij}^{\mathcal{O}} \text{ est libre } \Leftrightarrow \mathcal{O}(G_{ij}^{\mathcal{O}}) = 0 \\ \text{si } Al_{ij} \geq th & , \text{ alors } G_{ij}^{\mathcal{O}} \text{ est occupée, } \Leftrightarrow \mathcal{O}(G_{ij}^{\mathcal{O}}) = 1 \end{cases}$$

L'idée de base est intéressante mais la méthode possède trois inconvénients majeurs :

1. Ici, on considère que les mesures sont des points et on rappelle que S_{ij} correspond à la surface de la cellule G_{ij}

1. le score d'une cellule dépend non seulement de son état (libre ou occupé) mais aussi du nombre de fois où celle-ci a été observée. Al est donc difficile à interpréter. Une cellule avec un score faible peut correspondre à une cellule **vide** à cause des bruits de mesure ou à une cellule **occupée** mais très peu observée,
2. le principe de vote pour un seul état permet seulement la modélisation des environnements statiques,
3. la valeur du seuil est difficile à mettre en place compte-tenu de la disparité des valeurs possibles.

Nous proposons dans le paragraphe suivant un système de fusion dynamique qui tente de corriger ces inconvénients.

3.2.3. Fusion temporelle dans les grilles d'accumulation

Le principe de mise à jour présenté dans la section 2.4 et illustré dans la figure 3.1 a été mise en place avec le formalisme d'accumulation. Plusieurs améliorations ont été apportées pour permettre de considérer l'environnement dynamique et pour faciliter la décision.

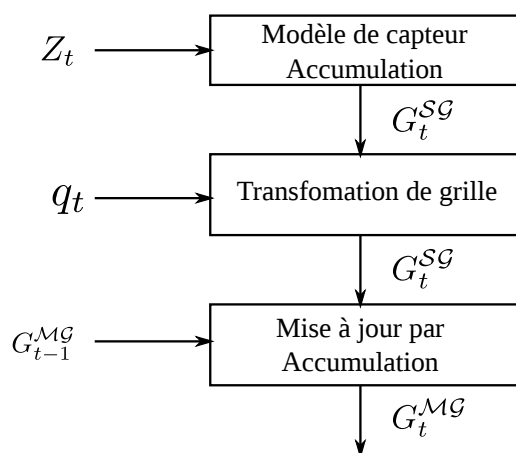


Figure 3.1.: Fusion de grilles d'accumulation, où Z_t correspond aux mesures, q_t est la pose du véhicule, G_{t-1}^{MG} est la MapGrid à l'instant précédent et G_t^{SG} est la ScanGrid courante.

Le modèle de capteur d'accumulation permet de construire la ScanGrid G^{SG} . Celle-ci est construite à partir des mesures selon la règle suivante :

$$Al^{SG} = \begin{cases} N & \text{si la cellule est occupée par } N \text{ mesures} \\ -M & \text{si la cellule est libre d'après } M \text{ mesures} \\ 0 & \text{si l'état de la cellule est inconnu} \end{cases}$$

La décrémentation du score d'une cellule observée libre permet de prendre en compte la dynamique de l'environnement.

La mise à jour de grille décrite dans la figure 3.2 s'effectue simplement en sommant les valeurs d'accumulation de la ScanGrid dans la MapGrid. On notera l'introduction des deux coefficients K_+ et K_- qui permettent d'attribuer des vitesses d'accumulation différentes pour l'incrémentaion et la décrémentation.

$$Al_{ij,t}^{MG} = Al_{ij,t-1}^{MG} + \begin{cases} K_+ \cdot Al_{ij,t}^{SG} & Al_{ij,t}^{SG} > 0 \\ K_- \cdot Al_{ij,t}^{SG} & Al_{ij,t}^{SG} \leq 0 \end{cases} \quad (3.2)$$

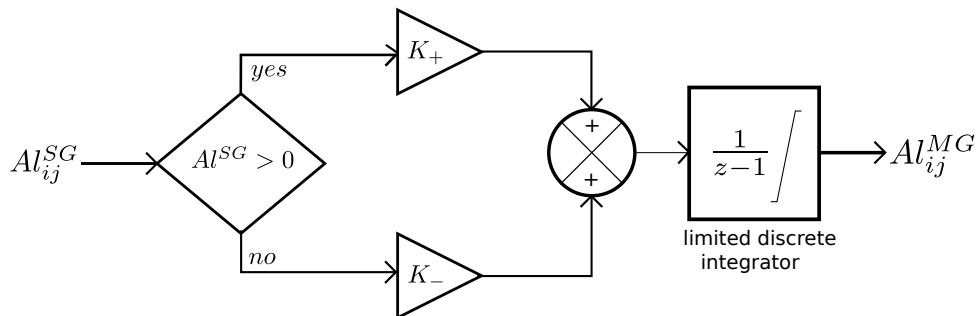


Figure 3.2.: Mise à jour de la MapGrid par accumulation

Nous proposons de contraindre le niveau d'accumulation dans un intervalle fixé $[Al_{min} Al_{max}]$. Cela permet de limiter les effets du nombre d'observations pour une cellule, ce qui peut se produire si l'ego-véhicule est arrêté. Cela permet aussi d'initialiser le système avec une valeur d'accumulation intermédiaire qui correspond alors à l'ignorance. La figure 3.3 montre le comportement de cette méthode de fusion. On observe que le temps de retour au niveau minimum d'accumulation pour la cellule libre est réduit grâce aux améliorations proposées.

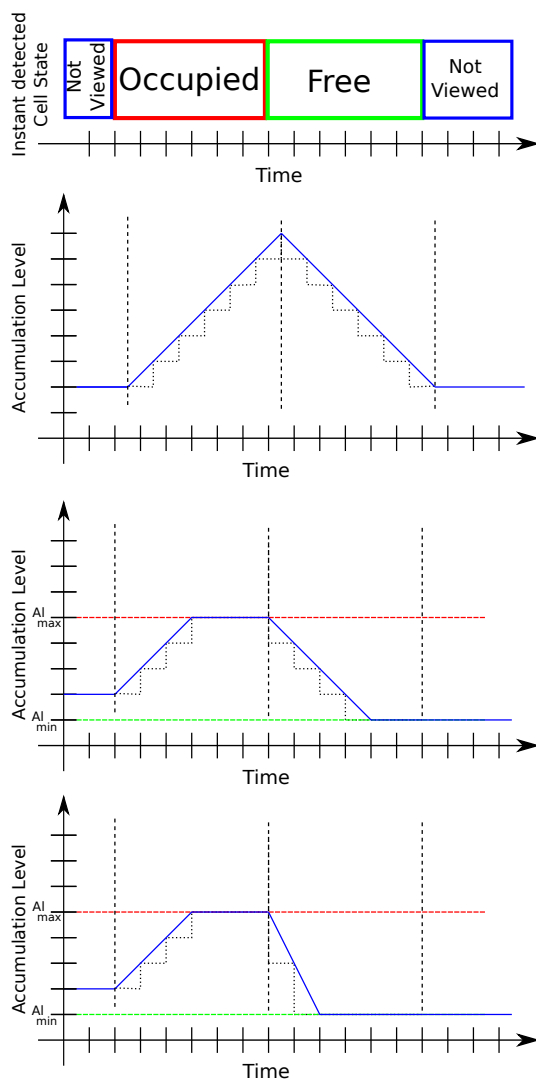


Figure 3.3.: Processus d'accumulation d'une cellule en fonction des observations. La figure du haut indique l'état observé pour la cellule à chaque instant (SG) avec $N = M = 1$. Les trois graphiques suivants indiquent l'évolution de Al dans la cellule respectivement avec (i) un processus d'incrément / décrémentation, (ii) un processus d'incrément / décrémentation borné et (iii) un processus d'incrément / décrémentation non symétrique borné.

Finalement, la décision se fait en utilisant deux seuils distincts, th_f et th_o , vérifiant :

$$th_f < th_o.$$

La décision est réalisée selon la règle énoncée ci-dessous :

$$\mathcal{O}(G_{ij}^{\mathcal{O}}) = \begin{cases} 1 & \text{si } Al_{ij} > th_o \\ 0 & \text{si } Al_{ij} < th_f \\ \text{non définie} & \text{si } th_f \leq Al_{ij} \leq th_o \end{cases}$$

Cette règle est plus prudente dans les cas ambigus et elle ne définit pas la fonction d'occupation dans ce cas. Les seuils sont cependant difficiles à choisir et sont, généralement, fixés de façon empirique.

3.3. Grille d'occupation probabiliste

3.3.1. Principe

Le formalisme probabiliste a été le premier formalisme de gestion des incertitudes utilisé dans les grilles d'occupation. Différentes formulations existent, utilisant soit un modèle de capteur direct soit un modèle inverse ou considérant l'environnement statique ou dynamique. Dans ce type d'approche, on utilise le principe d'inférence bayésienne pour estimer la probabilité d'occupation des cellules. Ceci est basé sur le théorème de Bayes [Bayes et Price, 1763] :

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

L'utilisation de ce formalisme avec une structure de grille a été proposée par Elfes dans [Elfes, 1989c, Elfes, 1989a]. On définit l'état d'une cellule parmi deux possibilités exclusives, occupée O ou libre F . Chaque cellule de la grille d'occupation $G_{ij}^{\mathcal{O}}$ contient une probabilité d'occupation : $I_{ij}^{\mathcal{O}} = P(\mathcal{O}(G_{ij}^{\mathcal{O}}) = 1) = P_{ij}(O)$. On peut aussi définir la probabilité que la cellule ne soit pas occupée $P_{ij}(F)$. Ces probabilités vérifient $\forall G_{ij}^{\mathcal{O}} \in G^{\mathcal{O}}, P(O) + P(F) = 1$. Dans cette approche, on fait l'hypothèse que toutes les cellules sont indépendantes entre elles.

3.3.2. Grille d'occupation en environnement statique

Pour chaque cellule², on cherche la probabilité d'occupation conditionnelle $P(O|r_{1:t})$, où $r_{1:t} = \{r_1, r_2, \dots, r_t\}$ représente l'ensemble des mesures obtenues jusqu'à cet instant t .

D'après le théorème de Bayes :

$$P(O|r_t, r_{1:t-1}) = \frac{p(r_t|O, r_{1:t-1}) \cdot P(O|r_{1:t-1})}{p(r_t|r_{1:t-1})} \quad (3.3)$$

où $p(r_t|O, r_{1:t-1})$ représente la vraisemblance de la mesure sachant l'occupation de la cellule comme on l'a défini dans le chapitre 2 comme le modèle de capteur direct.

Dans le cas où les erreurs de mesure sont indépendantes entre elles au cours du temps (par exemple si le bruit de mesure est blanc), on a :

$$p(r_t|O, r_{1:t-1}) = p(r_t|O) \quad (3.4)$$

En utilisant la marginalisation du dénominateur :

$$p(r_t|r_{1:t-1}) = p(r_t, O|r_{1:t-1}) + p(r_t, F|r_{1:t-1}) \quad (3.5)$$

d'où en utilisant l'équation 3.4 :

$$p(r_t|r_{1:t-1}) = p(r_t|O) P(O|r_{1:t-1}) + p(r_t|F) P(F|r_{1:t-1}) \quad (3.6)$$

Au final, on obtient l'équation 3.7 qui permet, à chaque fois qu'une nouvelle mesure r_t est disponible, de faire la mise à jour de la grille :

$$P(O|r_{1:t}) = \frac{p(r_t|O) \cdot P(O|r_{1:t-1})}{p(r_t|O) P(O|r_{1:t-1}) + p(r_t|F) P(F|r_{1:t-1})} \quad (3.7)$$

Cette mise à jour consiste à calculer la probabilité à posteriori $P(O|r_{1:t})$, à partir de la probabilité à priori $P(O|r_{1:t-1})$ et de la probabilité conditionnelle $p(r_t|O)$.

Dans son approche, Elfes considère que le modèle du bruit de mesure $p(r_t|z_t)$ correspondant à la vraisemblance entre la valeur mesurée par le capteur r sachant la vraie valeur z à l'instant t . Pour calculer le terme $p(r_t|O)$ de l'équation de mise à jour 3.7, à partir de $p(r_t|z_t)$, on utilise le théorème de Kolmogorov [Elfes, 1989b, Berger, 1985] sur les densités de probabilité de la mesure conditionnée pour toutes les configurations de grille possibles telles que la cellule soit occupée. Ce traitement est assez lourd en terme de calculs mais peut être calculé en prétraitement.

2. Pour simplifier les notations, on omet les indices ij .

Suivant la complexité du modèle de capteur, on peut calculer une solution analytique ou numérique. Dans le cas d'un capteur parfait $p(r|z) = \delta(r - z)$, une solution analytique simple existe. La figure 3.4 présente un profil d'allocation $P(O|r)$ pour une grille 1D construite à partir d'un capteur parfait. Les cellules situées avant l'obstacle ont une probabilité d'occupation nulle, la cellule située au niveau de l'obstacle à une probabilité égale à 1 et les cellule dernières étant non observées sont à 0.5.

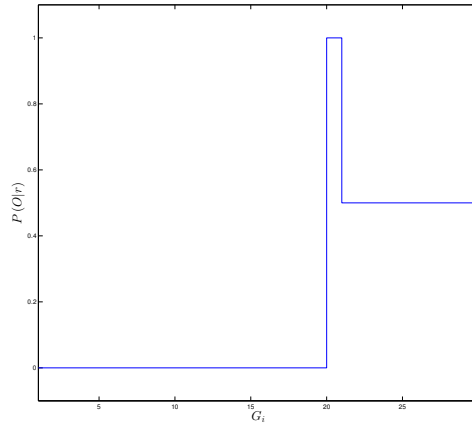


Figure 3.4.: Profil d'allocation 1D issu d'un capteur de distance parfait. Un obstacle est située dans la cellule G_{20} .

3.3.3. Règle de décision

La règle de décision élémentaire est la suivante :

$$\mathcal{O}(G_{ij}^{\mathcal{O}}) = \begin{cases} 1 & \text{si } P_{ij}(O) > P_{ij}(F) \\ 0 & \text{si } P_{ij}(O) < P_{ij}(F) \\ \text{non définie} & \text{si } P_{ij}(O) = P_{ij}(F) \end{cases}$$

Cette règle a l'inconvénient de ne pas définir l'occupation dans le cas où les probabilités sont égales.

Si on veut un niveau de confiance plus élevé, on peut utiliser deux seuils th_o et th_f sur les probabilités, comme on le fait pour l'accumulation.

$$\mathcal{O}(G_{ij}^{\mathcal{O}}) = \begin{cases} 1 & \text{si } P_{ij}(O) > th_o \\ 0 & \text{si } P_{ij}(F) > th_f \\ \text{non définie} & \text{sinon} \end{cases}$$

3.3.4. Fusion de grilles probabilistes

L'approche de mise à jour par fusion proposée dans le chapitre 2.4 est basée sur la fusion de grilles. Pour fusionner directement deux grilles indépendantes, Elfes propose d'utiliser la règle 3.8, à partir de l'équation 3.7 et du principe du sondage d'opinions indépendantes (*opinion independent poll*). Dans cette équation, $P_1 = P(O|r_{1:t}^1)$ et $P_2 = P(O|r_{1:t}^2)$ représentent respectivement la probabilité d'occupation de G_{ij}^1 construite à partir de l'ensemble des mesure $r_{1:t}^1$ et de G_{ij}^2 à partir de $r_{1:t}^2$. On notera par la suite \odot cette opérateur de combinaison.

$$P_{1\odot 2} = \frac{P_1 \cdot P_2}{P_1 \cdot P_2 + (1 - P_1) \cdot (1 - P_2)} \quad (3.8)$$

Démonstration. En utilisant la règle de Bayes, on a :

$$P(O|r_{1:t}^1, r_{1:t}^2) = \frac{P(r_t^1, r_t^2 | O, r_{1:t-1}^1, r_{1:t-1}^2) \cdot P(O|r_{1:t-1}^1, r_{1:t-1}^2)}{P(r_t^1, r_t^2 | r_{1:t-1}^1, r_{1:t-1}^2)} \quad (3.9)$$

Que l'on réécrit comme :

$$P(O|r_{1:t}^1, r_{1:t}^2) = P(r_t^1, r_t^2 | O, r_{1:t-1}^1, r_{1:t-1}^2) \cdot \frac{P(O|r_{1:t-1}^1, r_{1:t-1}^2)}{P(r_t^1, r_t^2 | r_{1:t-1}^1, r_{1:t-1}^2)} \quad (3.10)$$

En utilisant l'indépendance des mesures, on peut écrire :

$$P(O|r_{1:t}^1, r_{1:t}^2) = P(r_t^1 | O, r_{1:t-1}^1) \cdot P(r_t^2 | O, r_{1:t-1}^2) \cdot \frac{P(O|r_{1:t-1}^1, r_{1:t-1}^2)}{P(r_t^1, r_t^2 | r_{1:t-1}^1, r_{1:t-1}^2)} \quad (3.11)$$

Puis, on utilise de nouveau le théorème de Bayes sur $P(r_t^1 | O, r_{1:t-1}^1)$ et $P(r_t^2 | O, r_{1:t-1}^2)$, pour obtenir :

$$P(O|r_{1:t}^1, r_{1:t}^2) = \frac{P(O|r_t^1, r_{1:t-1}^1) \cdot P(r_t^1 | r_{1:t-1}^1)}{P(O|r_{1:t-1}^1)} \cdot \frac{P(O|r_t^2, r_{1:t-1}^2) \cdot P(r_t^2 | r_{1:t-1}^2)}{P(O|r_{1:t-1}^2)} \cdot \frac{P(O|r_{1:t-1}^1, r_{1:t-1}^2)}{P(r_t^1, r_t^2 | r_{1:t-1}^1, r_{1:t-1}^2)} \quad (3.12)$$

On réorganise les termes :

$$P(O|r_{1:t}^1, r_{1:t}^2) = P(O|r_{1:t}^1) \cdot P(O|r_{1:t}^2) \cdot \frac{P(O|r_{1:t-1}^1, r_{1:t-1}^2)}{P(O|r_{1:t-1}^1) \cdot P(O|r_{1:t-1}^2)} \cdot \frac{P(r_t^1|r_{1:t-1}^1) \cdot P(r_t^2|r_{1:t-1}^2)}{P(r_t^1, r_t^2|r_{1:t-1}^1, r_{1:t-1}^2)} \quad (3.13)$$

On arrive à une équation similaire si on considère l'état F :

$$P(F|r_{1:t}^1, r_{1:t}^2) = P(F|r_{1:t}^1) \cdot P(F|r_{1:t}^2) \cdot \frac{P(F|r_{1:t-1}^1, r_{1:t-1}^2)}{P(F|r_{1:t-1}^1) \cdot P(F|r_{1:t-1}^2)} \cdot \frac{P(r_t^1|r_{1:t-1}^1) \cdot P(r_t^2|r_{1:t-1}^2)}{P(r_t^1, r_t^2|r_{1:t-1}^1, r_{1:t-1}^2)} \quad (3.14)$$

Comme la somme des probabilités de libre et d'occupé égale 1, on a :

$$P(O|r_{1:t}^1, r_{1:t}^2) + P(F|r_{1:t}^1, r_{1:t}^2) = 1 \quad (3.15)$$

Dans le cas où l'on n'a pas d'à priori (on fusionne sans considérer le résultat de la fusion précédente), $P(O|r_{1:t-1}^1, r_{1:t-1}^2) = P(O|r_{1:t-1}^1) = P(O|r_{1:t-1}^2) = 0.5$ (de même pour les probabilités de libre), on a alors l'égalité suivante :

$$\frac{P(O|r_{1:t-1}^1, r_{1:t-1}^2)}{P(O|r_{1:t-1}^1) \cdot P(O|r_{1:t-1}^2)} = \frac{P(F|r_{1:t-1}^1, r_{1:t-1}^2)}{P(F|r_{1:t-1}^1) \cdot P(F|r_{1:t-1}^2)} = 0.25 \quad (3.16)$$

On pose α :

$$\alpha = 0.25 \cdot \frac{P(r_t^1|r_{1:t-1}^1) \cdot P(r_t^2|r_{1:t-1}^2)}{P(r_t^1, r_t^2|r_{1:t-1}^1, r_{1:t-1}^2)} \quad (3.17)$$

En remplaçant α dans 3.13 et 3.14 :

$$P(O|r_{1:t}^1, r_{1:t}^2) = P(O|r_{1:t}^1) \cdot P(O|r_{1:t}^2) \cdot \alpha \quad (3.18)$$

$$P(F|r_{1:t}^1, r_{1:t}^2) = P(F|r_{1:t}^1) \cdot P(F|r_{1:t}^2) \cdot \alpha \quad (3.19)$$

L'équation 3.15 devient :

$$1 = \alpha \cdot \left(P(O|r_{1:t}^1) \cdot P(O|r_{1:t}^2) + P(F|r_{1:t}^1) \cdot P(F|r_{1:t}^2) \right) \quad (3.20)$$

Ce qui permet de déduire le coefficient de normalisation α :

$$\alpha = \frac{1}{P(O|r_{1:t}^1) \cdot P(O|r_{1:t}^2) + P(F|r_{1:t}^1) \cdot P(F|r_{1:t}^2)} \quad (3.21)$$

d'où comme $P(F|r_{1:t}^1) = 1 - P(O|r_{1:t}^1)$ et $P(F|r_{1:t}^2) = 1 - P(O|r_{1:t}^2)$:

$$P(O|r_{1:t}^1, r_{1:t}^2) = \frac{P(O|r_{1:t}^1) \cdot P(O|r_{1:t}^2)}{P(O|r_{1:t}^1) \cdot P(O|r_{1:t}^2) + (1 - P(O|r_{1:t}^1)) \cdot (1 - P(O|r_{1:t}^2))} \quad (3.22)$$

□

A chaque scan, une SG est construite en utilisant l'équation 3.7 à partir d'une grille inconnue, donc initialisée pour chaque cellule de la façon suivante :

$$P_{ij}^{SG}(O) = P_{ij}^{SG}(F) = 0.5$$

La MG est initialisée de la même façon, puis pour chaque cellule, elle est propagée par mise à jour en utilisant l'équation 3.8 et la SG créée à chaque pas de temps de la manière suivante :

$$P_t^{MG} = P_{t-1}^{MG} \odot P_t^{SG} = \frac{P_{t-1}^{MG} \cdot P_t^{SG}}{P_{t-1}^{MG} \cdot P_t^{SG} + (1 - P_{t-1}^{MG}) \cdot (1 - P_t^{SG})}$$

3.3.5. Grille d'occupation en environnement dynamique

Pour prendre en compte la dynamique de l'environnement, certaines méthodes sont capables de traiter des problèmes dynamiques. Le *Bayesian Occupancy Filter (BOF)* développé par l'équipe eMotion de l'INRIA-Grenoble [Tay *et al.*, 2008, Coué *et al.*, 2006] propose d'utiliser une structure Estimateur - Prédicteur pour la mise à jour de la grille en suivant une approche bayésienne. Pour cela, cet algorithme estime non

seulement l'occupation des cellules mais aussi leur vitesse pour pouvoir réaliser la prédiction pour chaque cellule.

Dans cette approche, pour chaque cellule³ $c = G_{ij}^c$, les variables estimées sont O_t^c , l'occupation au temps t de la cellule c et A_t^c , l'antécédent au temps t de la cellule c .

La vitesse d'une cellule c prend valeur dans un ensemble discret et fini de vitesses possibles $V = \{V_1, V_2, \dots, V_n\}$, comme indiqué sur la figure 3.5. A chaque vitesse discrète V_k correspond une cellule C_j appartenant au voisinage de c , de telle sorte que la cellule C_j se déplace en c si elle se déplace avec une vitesse V_k durant un pas temporel. On dit alors que C_j est l'antécédent de c ($A_t^c = C_j$).

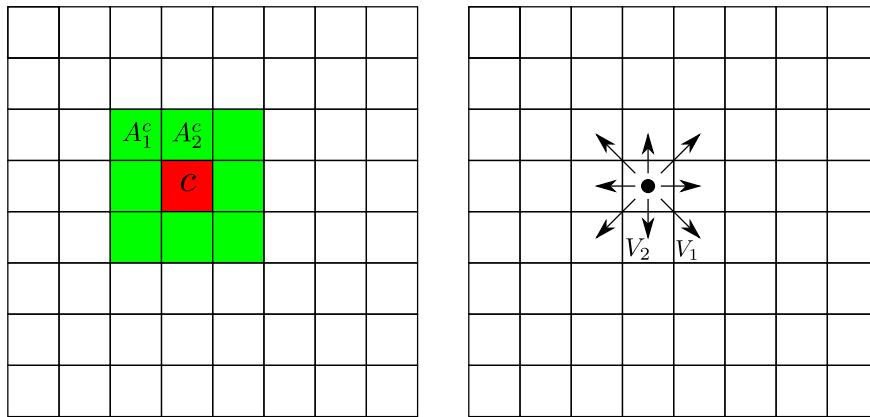


Figure 3.5.: Notion de vitesses discrètes : sur la grille à gauche, chaque cellule verte est un antécédent possible de la cellule rouge et on lui associe alors la vitesse, représentée sur la grille à droite, qui permettrait de passer de cette cellule à la cellule rouge.

L'objectif du filtre est de calculer la probabilité à posteriori $P(A_t^c, O_t^c | Z_t)$ ce qui peut être implémenté selon un schéma estimateur - prédicteur décrit par le schéma de la figure 3.6.

Estimation On peut exprimer cette probabilité en utilisant le théorème de Bayes à partir de la probabilité à priori $P(A_t^c, O_t^c)$, de la mesure Z_t et du modèle de capteur $P(Z_t | A_t^c, O_t^c)$:

$$P(A_t^c, O_t^c | Z_t) = \frac{P(A_t^c, O_t^c) \cdot P(Z_t | A_t^c, O_t^c)}{P(Z_t)} \quad (3.23)$$

D'où on déduit :

3. Dans cette section, on notera une cellule c afin de simplifier les notations

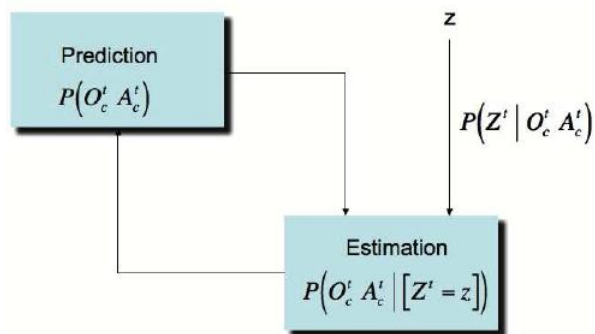


Figure 3.6.: Structure estimateur - prédicteur du BOF

$$P(A_t^c, O_t^c | Z_t) \propto P(A_t^c, O_t^c) \cdot P(Z_t | A_t^c, O_t^c)$$

Enfin on normalise en utilisant la condition :

$$\sum_{O_t^c} \sum_{A_t^c} P(A_t^c, O_t^c | Z_t) = 1$$

Dans l'équation 3.23, la probabilité $P(A_t^c, O_t^c)$ est une probabilité à priori qui est calculée pendant l'étape de prédiction. La probabilité $P(Z_t | A_t^c, O_t^c)$ correspond au modèle de capteur direct, qui est ici un modèle de capteur en occupation et en vitesse. Néanmoins, si le capteur ne fournit pas d'information relative à la vitesse, on peut utiliser un modèle de capteur qui fournit des informations uniquement sur l'occupation $P(Z_t | O_t^c)$.

Prédiction On utilise un modèle de prédiction de O_t^c et A_t^c défini à partir d'un ensemble d'hypothèses (notamment l'hypothèse de vitesse constante) qui ne dépend que de A_{t-1}^c . En supposant que O_t^c et A_t^c sont indépendants entre eux et dépendent uniquement de A_{t-1}^c , on peut exprimer la probabilité jointe $P(A_{t-1}^c, A_t^c, O_t^c)$ de la façon suivante :

$$P(A_{t-1}^c, A_t^c, O_t^c) = P(A_t^c | A_{t-1}^c) \cdot P(O_t^c | A_{t-1}^c) \cdot P(A_{t-1}^c)$$

La probabilité $P(A_t^c, O_t^c)$ est obtenue par marginalisation de $P(A_{t-1}^c, A_t^c, O_t^c)$ suivant A_{t-1}^c :

$$P(A_t^c, O_t^c) = \sum_{A_{t-1}^c} P(A_{t-1}^c, A_t^c, O_t^c) = \sum_{A_{t-1}^c} P(A_t^c | A_{t-1}^c) \cdot P(O_t^c | A_{t-1}^c) \cdot P(A_{t-1}^c) \quad (3.24)$$

Dans cette équation, $P(A_{t-1}^c)$ correspond à l'estimation de la vitesse à l'instant précédent. $P(O_t^c|A_{t-1}^c)$ correspond à la propagation de l'occupation des cellules antécédentes. Il en est de même pour $P(A_t^c|A_{t-1}^c)$ qui correspond à la propagation des vitesses. Ces propagations sont réalisées en faisant l'hypothèse de vitesse constante.

Le BOF est un outil très puissant qui permet de propager la grille même si on ne dispose pas de mesure pendant une courte période. En contre partie, cette méthode a quelques inconvénients :

- C'est une méthode qui a un coût calculatoire élevé car elle calcule toutes les hypothèses possibles suivant la zone de voisinage considérée pour l'ensemble des vitesses.
- Comme il faut normaliser dans l'étape d'estimation, il faut alors calculer entièrement la probabilité jointe $P(A_t^c, O_t^c)$. Donc la vitesse (ou l'antécédent) est estimée, même lorsque les cellules sont vides. Cela engendre de nombreux calculs qui n'apportent pas d'information sur l'état de la cellule.

On trouve de nombreux travaux et résultats basés sur cette approche comme la fusion multi-capteurs [Paromtchik *et al.*, 2011], la segmentation de la grille d'occupation en utilisant les critères des positions et de vitesses des cellules [Kamel Mekhnacha, Yong Mao, David R. ou l'implémentation temps réel utilisant le calcul sur GPU [Ferreira *et al.*, 2010]. D'autre part, ce type d'approche est de plus en plus utilisé dans le milieu robotique, par exemple dans [Gate, 2009] une méthode similaire est développée.

3.4. Grille d'occupation évidentielle

Le formalisme évidentiel n'est pas beaucoup utilisé en robotique, principalement, à cause de sa complexité calculatoire mais on peut noter quelques travaux sur les grilles évidentielles. [Pagac *et al.*, 1998] s'intéresse à la construction de grilles d'occupation statique à partir de capteurs ultrasons. [Aitken, 2005] quand à lui proposent une analyse détaillée de l'approche évidentielle à partir de simulations.

3.4.1. Principe

Dans le cas de grilles d'occupation évidentielles, l'incertitude est modélisée en utilisant le formalisme des fonctions de croyance. Le cadre de discernement Ω est défini tel que $\Omega = \{F, O\}$, le référentiel de définition est $2^\Omega = \{ \emptyset, F, O, \Omega \}$. On considère que le cadre de discernement est exhaustif. Il existe plusieurs représentations des fonctions de croyance (mass, belief, plausibility, communality). Toutes ces représentations sont équivalentes et nous utilisons la fonction de masse⁴ m^Ω telle

4. On ne répètera pas le cadre de discernement Ω dans la notation des masses.

que :

$$\begin{aligned} m &: 2^\Omega \rightarrow [0, 1] \\ \sum_{A \in 2^\Omega} m(A) &= 1. \end{aligned} \tag{3.25}$$

La fonction de masse d'une cellule est un vecteur contenant 4 masses qui représentent les parts de croyance sur chaque élément de 2^Ω :

$$I_{ij}^O = \{ m(\emptyset) \quad m(F) \quad m(O) \quad m(\Omega) \}$$

Chaque masse correspond au niveau de croyance que la cellule soit dans l'état correspondant. Une cellule est initialisée avec une fonction de masse appelée **bba** (basic belief assignment).

Dans les exemples numériques, on notera les fonctions de masse sous forme d'une matrice 2x4. Sur la première ligne on rappellera le sous ensemble considéré et sur la seconde ligne la masse de ce sous-ensemble :

$$m = \begin{bmatrix} \emptyset & F & O & \Omega \\ m(\emptyset) & m(F) & m(O) & m(\Omega) \end{bmatrix} \tag{3.26}$$

Par exemple, une fonction de masse $\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 1 & 0 & 0 \end{bmatrix}$ indique une cellule libre avec certitude, alors que $\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 1 & 0 \end{bmatrix}$ indique une cellule occupée avec certitude.

L'état $\Omega = \{F, O\}$ représente de manière explicite l'ignorance. La fonction de masse $\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}$ indique donc l'ignorance totale.

L'état \emptyset représente l'ensemble vide. Une masse sur ce sous-ensemble provient principalement de la fusion de deux masses conflictuelles et signifie qu'aucun état compatible n'est possible. Hormis dans certains cas où on utilise la fusion disjonctive, une bba ne contient pas de masse sur l'ensemble vide.

3.4.2. Fonction de décision

Dans les grilles incertaines, on ne modélise plus l'état d'occupation de la cellule directement mais on estime une autre fonction qui modélise la confiance. Cependant, les tâches qui dépendent de la perception ont souvent besoin d'un état binaire : il faut revenir à la définition de la fonction d'occupation : occupée ou libre.

Pour ce faire, le formalisme des fonctions de croyance dispose d'un grand nombre d'outils, notamment au travers du Transferable Belief Model (TBM) décrit par Smets dans [Smets et Kennes, 1994, Smets, 1990].

Probabilité pignistique La probabilité pignistique $BetP$ est souvent la première règle utilisée comme règle de décision. Cette transformation permet de transformer les fonctions de masse en probabilités sur des singletons. Pour ce faire, les masses sont normalisées de telle manière que $m(\emptyset) = 0$, puis la masse de chaque sous ensemble est répartie équitablement sur chaque singleton qui le compose, comme le montre l'équation 3.27, où l'opérateur $|X|$ représente le cardinal de l'ensemble X .

$$\forall A \in \Omega, BetP(A) = \sum_{X \in 2^\Omega} \frac{|X \cap A|}{|X|} \cdot \frac{m(X)}{1 - m(\emptyset)} \quad (3.27)$$

Exemple 1 :

$$\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0.6 & 0.4 \end{bmatrix}$$

$$BetP(F) = 0.2 \quad BetP(O) = 0.8$$

Exemple 2 :

$$\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$BetP(F) = 0.5 \quad BetP(O) = 0.5$$

Exemple 3 :

$$\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.3 & 0.5 & 0.2 \end{bmatrix}$$

$$BetP(F) = 0.4 \quad BetP(O) = 0.6$$

À partir de la probabilité pignistique, on peut utiliser les techniques de décision probabilistes classiques, voir section 3.3.3.

Bornes supérieures et inférieures Une autre solution est d'utiliser les degrés de crédibilité ou de plausibilité définis dans le cadre du TBM.

La fonction de crédibilité bel indique le niveau de certitude pour un état. Elle est calculée comme :

$$bel(A) = \sum_{\emptyset \neq X \subseteq A} m(X) \quad (3.28)$$

La fonction de plausibilité pl indique le niveau de plausibilité pour un état, elle est

calculée comme :

$$pl(A) = \sum_{X \cap A \neq \emptyset} m(X) \quad (3.29)$$

Ces deux fonctions sont des bornes supérieures et inférieures de la croyance de chaque état. Ces fonctions vérifient :

$$pl(A) = bel(\Omega) - bel(\bar{A}) = 1 - bel(\bar{A}) \quad (3.30)$$

Les exemples ci dessous permettent de comparer les résultats des fonctions bel , pl et $BetP$:

Exemple 1 :

$$\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0.6 & 0.4 \end{bmatrix}$$

$$\begin{aligned} bel(F) &= 0 & bel(O) &= 0.6 \\ pl(F) &= 0.4 & pl(O) &= 1 \end{aligned}$$

$$BetP(F) = 0.2 \quad BetP(O) = 0.8$$

Exemple 2 :

$$\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} bel(F) &= 0 & bel(O) &= 0 \\ pl(F) &= 1 & pl(O) &= 1 \end{aligned}$$

$$BetP(F) = 0.5 \quad BetP(O) = 0.5$$

Exemple 3 :

$$\begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.3 & 0.5 & 0.2 \end{bmatrix}$$

$$\begin{aligned} bel(F) &= 0.3 & bel(O) &= 0.5 \\ pl(F) &= 0.5 & pl(O) &= 0.7 \end{aligned}$$

$$BetP(F) = 0.4 \quad BetP(O) = 0.6$$

On remarque que dans chaque exemple la probabilité pignistique est bien encadrée par la crédibilité et la plausibilité. De plus, on remarque que, dans notre cas, la probabilité pignistique se trouve au centre de l'intervalle $[bel(A) \quad pl(A)]$. La taille de cet intervalle est aussi relié à l'ignorance. A partir de ces deux valeurs et suivant les contraintes de l'application, on peut définir la fonction \mathcal{O} . Par exemple, une

approche de navigation prudente restreindra la navigation dans les zones libres avec un niveau de confiance élevé, c'est-à-dire quand $bel(F) = 1 - pl(O)$ est supérieure à un seuil de confiance donné.

Les fonctions bel et pl sont une représentation de l'information équivalente à la fonction de masse m et on peut reconstruire une fonction de masse à partir des croyances ou des plausibilités. On peut noter qu'il existe deux autres opérateurs : la communalité q et l'implacabilité b , qui permettent aussi la représentation de l'information.

3.4.3. Grille évidentielle en environnement dynamique

Correction par inférence et prédiction A l'instar des méthodes d'inférence bayésienne, les méthodes évidentielles peuvent aussi grâce au Generalized Bayesian Theorem (GBT) [Smets, 1993] résoudre des problèmes posés sous forme de réseaux ou graphes [Shenoy, 1989].

Par exemple, on considère le problème dans lequel interviennent deux variables binaires x et θ . On peut définir une fonction de masse pour ces deux variables respectivement sur X et Θ . Si on dispose d'informations relationnelles entre x et θ , par exemple les plausibilités conditionnelles :

$pl(x \theta)$	$pl(\bar{x} \theta)$	$pl(x \cup \bar{x} \theta)$
$pl(x \bar{\theta})$	$pl(\bar{x} \bar{\theta})$	$pl(x \cup \bar{x} \bar{\theta})$

On peut propager une information à priori sur x : $m^X(x)$, $m^X(\bar{x})$, $m^X(x \cup \bar{x})$ pour inférer une information sur θ de la façon suivante⁵ :

$$\begin{aligned}
 pl(\theta) &= m(x) pl(x|\theta) + m(\bar{x}) pl(\bar{x}|\theta) + m(x \cup \bar{x}) pl(x \cup \bar{x}|\theta) \\
 pl(\bar{\theta}) &= m(x) pl(x|\bar{\theta}) + m(\bar{x}) pl(\bar{x}|\bar{\theta}) + m(x \cup \bar{x}) pl(x \cup \bar{x}|\bar{\theta}) \\
 pl(\theta \cup \bar{\theta}) &= m(x) (1 - (1 - pl(x|\theta))(1 - pl(x|\bar{\theta}))) + \\
 &\quad m(\bar{x}) (1 - (1 - pl(\bar{x}|\theta))(1 - pl(\bar{x}|\bar{\theta}))) + \\
 &\quad m(x \cup \bar{x}) (1 - (1 - pl(x \cup \bar{x}|\theta))(1 - pl(x \cup \bar{x}|\bar{\theta})))
 \end{aligned} \tag{3.31}$$

Ainsi grâce au GBT, on peut inférer les informations de la même façon que dans le cadre bayésien. On peut donc en théorie résoudre l'inférence de grille d'occupation évidentielle de la même façon qu'avec le formalisme probabiliste, en utilisant comme dans le BOF (voir section 3.3.5) un modèle de capteur direct $m(r|A)$, $A \in \Omega$.

Cependant, l'utilisation de GBT dans notre approche pose deux problèmes : la création de masse pour les modèles continus de capteur et la complexité calculatoire qui augmente de façon exponentielle avec le nombre d'états considérés. En effet, dans

5. L'inférence inverse se résout de façon symétrique

la majorité des cas, les fonctions de croyance sont utilisées dans un cadre discret. Il existe certains travaux qui proposent des extensions au cadre continu comme l'approximation des densités de probabilité continues par des fonctions de masse. Dans [Dubois et Prade, 1990], par exemple, on approxime des densités en utilisant des fonctions de masse consonantes inspirées des approches de logique floue. Hélas, si on veut une bonne approximation de la densité avec ce type d'approche, alors la taille du cadre de discernement devient tel qu'il est actuellement impossible de l'implémenter en temps réel.

D'autre part, pour réaliser une prédiction similaire à celle du BOF, il faudrait augmenter le cadre de discernement avec les vitesses. Par exemple, dans les cas du BOF avec un voisinage d'une cellule, on considèrerait 9 antécédents possibles et donc 2^9 masses à calculer en plus du cadre actuel. Ceci est largement incompatible avec les capacités de calcul actuelles.

Stratégie proposée : fusion et auscultation rapide Pour réduire la complexité algorithmique et permettre une implémentation temps réel, on propose d'utiliser une approche utilisant des modèles inverses de capteur. Ainsi, l'inférence peut être remplacée par une étape de fusion. Ceci permet de rendre possible l'exécution en temps réel de l'estimation d'une grille évidentielle. On actualise la grille en appliquant une mise à jour de type fusion de grilles, similaire à celle de la partie 3.3.4 :

$$m_t^{MG} = m_{t-1}^{MG} \otimes m_t^{SG} \quad (3.32)$$

où \otimes correspond à un opérateur de fusion générique qui représente indifféremment un opérateur présenté dans la section suivante.

N'estimant pas les paramètres dynamiques de la scène, on ne peut réaliser une prédiction correcte de la scène et on va donc faire l'hypothèse de monde statique. Cependant, en considérant la fréquence du capteur grande devant la dynamique de la scène, alors l'erreur de prédiction est faible car l'environnement est presque fixe entre deux scans. Si des cellules ne sont plus observées, l'hypothèse d'auscultation rapide faite ci-dessus n'est plus possible. Pour prendre en compte ce phénomène, on met en place une fonction d'oubli. On verra par la suite que la théorie des fonctions de croyance permet de réaliser facilement un tel affaiblissement. Cette fonction transfère une part α de l'information spécifique vers de l'ignorance. Cette atténuation est réalisée en suivant un modèle de durée de vie exponentiel décroissant décrit dans l'équation 3.33. La constante de temps τ détermine la vitesse d'oubli, l'écart de temps entre le dernier scan et le temps courant est noté Δt .

$$\alpha = 1 - e^{-\frac{\Delta t}{\tau}} \quad (3.33)$$

3.4.4. Opérateurs de combinaison

Pour fusionner les fonctions de masse, il existe plusieurs opérateurs. Chaque opérateur a des propriétés particulières et s'utilise dans certains cas. Nous allons présenter ici les principaux opérateurs.

Fusion conjonctive L'opérateur de fusion conjonctive permet de fusionner deux fonctions de masse indépendantes définies sur le même cadre de discernement. Cet opérateur de fusion fait l'hypothèse que toutes les sources sont fiables. Il permet d'obtenir une fonction de masse plus informative que les deux sources. Il est très similaire à l'opérateur intersection (\cap) dans la théorie des ensembles.

Pour réaliser la fusion de deux masses avec cet opérateur, on utilise la formule :

$$\forall A \in 2^\Omega, m_1 \odot_2 (A) = \sum_{B \cap C = A | B, C \subseteq \Omega} m_1(B) \cdot m_2(C) \tag{3.34}$$

Cet opérateur est commutatif et associatif. L'élément neutre est Ω et l'élément absorbant est \emptyset .

Si on fusionne deux fonctions de masse m_1 et m_2 , la masse résultante est une somme de produit de différents éléments de m_1 et de m_2 . Le tableau 3.1 indique l'ensemble d'arrivée du produit de masses au cours de la fusion, il est déduit à partir de l'opérateur \cap .

m_1	m_2	\emptyset	F	O	Ω
\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
F	\emptyset	\emptyset	F	\emptyset	F
O	\emptyset	\emptyset	\emptyset	O	O
Ω	\emptyset	\emptyset	F	O	Ω

Table 3.1.: Tableau d'attribution des masses de la fusion conjonctive

Exemple 1 :

Si on considère par exemple les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.8 & 0 & 0.2 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.7 & 0 & 0.3 \end{bmatrix}.$$

Le tableau 3.2 détaille le calcul de la fusion de m_1 et m_2 .

La fonction de masse résultante $m_1 \odot_2$ est :

m_2	\emptyset	F	O	Ω
m_1	0	0.7	0	0.3
\emptyset	0	0	0	0
F	0	0.56	0	0.24
O	0	0	0	0
Ω	0	0.14	0	0.06

Table 3.2.: Exemple d'attribution des masses de la fusion conjonctive

$$m_1 \oplus_2 (\emptyset) = 0$$

$$m_1 \oplus_2 (F) = 0.56 + 0.24 + 0.14 = 0.94$$

$$m_1 \oplus_2 (O) = 0$$

$$m_1 \oplus_2 (\Omega) = 0.06$$

Exemple 2 :

Si on considère, par exemple, les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.4 & 0 & 0.6 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.1 & 0 & 0.9 \end{bmatrix} \text{ alors } m_1 \oplus_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.46 & 0 & 0.54 \end{bmatrix}.$$

Exemple 3 :

Si on considère, par exemple, les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix} \text{ alors } m_1 \oplus_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0.2 & 0.3 & 0.2 & 0.3 \end{bmatrix}.$$

Fusion disjonctive L'opérateur de fusion disjonctive permet de fusionner deux fonctions de masse indépendantes définies sur le même cadre de discernement. Cet opérateur de fusion est utilisé lorsqu'au moins une source est fiable mais sans savoir laquelle. Il permet d'obtenir une fonction de masse compatible avec les deux sources. A l'instar de l'opérateur conjonctif, il est très similaire à l'opérateur union (\cup).

Pour réaliser la fusion de deux masses avec cet opérateur, on utilise la formule :

$$\forall A \in 2^\Omega, m_1 \oplus_2 (A) = \sum_{B \cup C = A | B, C \subseteq \Omega} m_1(B) \cdot m_2(C) \quad (3.35)$$

Cet opérateur est commutatif et associatif. L'élément neutre est \emptyset et l'élément absorbant est Ω .

Si on fusionne deux fonctions de masse m_1 et m_2 , la masse résultante est une somme de produits de différents éléments de m_1 et de m_2 . Le tableau 3.1 indique l'ensemble d'arrivée du produit de masses au cours de la fusion. Il est déduit à partir de l'opérateur \cup .

m_1	m_2	\emptyset	F	O	Ω
\emptyset	\emptyset	F	O	Ω	
F	F	Ω	O	Ω	
O	O	Ω	O	Ω	
Ω	Ω	Ω	Ω	Ω	

Table 3.3.: Tableau d'attribution des masses de la fusion disjonctive

Exemple 1 :

Si on considère, par exemple, les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.8 & 0 & 0.2 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.7 & 0 & 0.3 \end{bmatrix}.$$

Le tableau 3.4 détaille le calcul de la fusion.

m_1	m_2	\emptyset	F	O	Ω
\emptyset	0	0	0.7	0	0.3
\emptyset	0	\emptyset	F	O	Ω
F	0.8	0	0	0	0
F	0.8	F	F	Ω	Ω
		0	0.56	0	0.24
O	0	O	Ω	O	Ω
		0	0	0	0
Ω	0.2	Ω	Ω	Ω	Ω
		0	0.214	0	0.06

Table 3.4.: Exemple d'attribution des masses de la fusion disjonctive

La fonction de masse résultante est :

$$m_1 \oplus_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.56 & 0 & 0.14 + 0.24 + 0.06 = 0.44 \end{bmatrix}.$$

Exemple 2 :

Si on considère, par exemple, les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.4 & 0 & 0.6 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.1 & 0 & 0.9 \end{bmatrix} \text{ alors } m_1 \oplus_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.04 & 0 & 0.96 \end{bmatrix}.$$

Exemple 3 :

Si on considère, par exemple, les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix} \text{ alors } m_1 \oplus_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Fusion de Dempster L'opérateur de fusion de Dempster est un opérateur introduit par Dempster dans [Dempster, 1968]. C'est l'opérateur de référence qui est utilisé dans le domaine des fonction de croyance. La règle de fusion de Dempster est réalisée en deux étapes : la fusion conjonctive suivie de la normalisation de la masse conflictuelle $m(\emptyset)$.

$$\begin{cases} m_{1 \oplus_2}(A) = \frac{\sum_{B \cap C = A | B, C \subseteq \Omega} m_1(B) \cdot m_2(C)}{1 - m_{1 \oplus_2}(\emptyset)} = \frac{m_1 \oplus_2(A)}{1 - m_{1 \oplus_2}(\emptyset)} \\ m_{1 \oplus_2}(\emptyset) = 0 \end{cases} \quad (3.36)$$

Cet opérateur a de bonnes propriétés mathématiques, puisqu'il est associatif et commutatif. De plus, il ne génère pas de masse sur \emptyset ce qui est en général préférable lors que l'on veut utiliser le résultat pour faire une prise de décision ou pour propager l'information. Cette règle peut produire des résultats contre-intuitifs dans certains cas mis en évidence dans [Zadeh, 1984], qui sont principalement liés à une mauvaise interprétation (probabiliste) du cadre des fonction de croyance.

S'il n'y a pas de conflit alors les résultats sont les mêmes que la fusion conjonctive, ce qui est le cas dans les exemples 1 et 2.

Exemple 3 :

Si on considère, par exemple, les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix} \text{ alors } m_{1 \oplus_2} = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.375 & 0.25 & 0.375 \end{bmatrix}.$$

Dans ce cas là, la masse sur l'ensemble vide va se répartir sur chaque état présent en proportion de sa masse.

Fusion de Yager La règle de Yager [Yager, 1987] est aussi basée sur la règle de combinaison conjonctive. A l'instar de la fusion disjonctive, elle est utilisée lorsqu'une source n'est pas fiable. Elle permet de lever le problème du conflit en transférant la masse de conflit sur Ω . Le problème de cette règle est qu'elle perd la propriété d'associativité :

$$(m_1 \oplus_y m_2) \oplus_y m_3 \neq m_1 \oplus_y (m_2 \oplus_y m_3).$$

Le tableau 3.5 détaille le calcul de la fusion.

$m_1 \backslash m_2$	\emptyset	F	O	Ω
\emptyset	Ω	Ω	Ω	Ω
F	Ω	F	Ω	F
O	Ω	Ω	O	O
Ω	Ω	F	O	Ω

Table 3.5.: Tableau d'attribution des masses de la fusion de Yager

De même que pour la règle de Dempster, les résultats des exemples 1 et 2 sont les mêmes que pour la fusion conjonctive.

Exemple 3 :

Si on considère, par exemple, les deux fonctions de masse :

$$m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix} \text{ et } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix}.$$

$$m_1 \circledast m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.3 & 0.2 & 0.4 \cdot 0.5 + 0.6 \cdot 0.5 = 0.5 \end{bmatrix}$$

Fusion prudente L'opérateur de fusion prudente [Denoeux, 2006] doit être utilisé lorsque les sources ne sont pas indépendantes. Il ne prend donc pas en compte les redondances d'information, car il est basé sur le principe de l'engagement minimal :

Least commitment principle: "When several belief functions are compatible with a set of constraints, the least informative according to some informational ordering (if it exists) should be selected".

Cet opérateur est associatif, commutatif mais surtout idempotent :

$$(m_1 \circledast m_1) = m_1$$

L'élément neutre est Ω .

Le calcul de la fusion n'est pas aussi simple que dans les cas précédents i.e. plus coûteux au niveau calculatoire.

On considère que l'on veut fusionner deux fonctions de masse m_1 et m_2 .

1. Pour chaque fonction de masse initiale, il faut calculer les poids w calculés à partir de la communalité q définie dans l'équation 3.37.

$$q(A) = \sum_{B \supseteq A} m(B). \quad (3.37)$$

Dans notre cas (fonction de masse répartie sur F , O et Ω), cela donne :

$$q_i = \begin{bmatrix} \emptyset & F & O & \Omega \\ 1 & m_i(L) + m_i(\Omega) & m_i(O) + m_i(\Omega) & m_i(\Omega) \end{bmatrix}.$$

Puis, il faut calculer des poids w à partir de 3.38 :

$$w_i(A) = \prod_{B \supseteq A} q(B)^{(-1)^{|B|-|A|+1}} \quad (3.38)$$

ce qui donne :

$$w_i = \begin{bmatrix} \emptyset & F & O & \Omega \\ 1 & \frac{m_i(\Omega)}{m_i(L)+m_i(\Omega)} & \frac{m_i(\Omega)}{m_i(O)+m_i(\Omega)} & \end{bmatrix}.$$

On remarque que Ω n'a pas de poids.

2. Pour chaque état, on conserve le poids minimal d'après le *principe d'engagement minimal* [Smets, 1993] où \wedge représente l'opérateur *min*.

$$w_f = w_1(F) \wedge w_2(F)$$

$$w_o = w_1(O) \wedge w_2(O)$$

$$w_\emptyset = 1$$

3. Pour chaque état, on construit une fonction de masse simple :

$$m_f = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 1 - w_f & 0 & w_f \end{bmatrix}$$

$$m_o = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 1 - w_o & w_o \end{bmatrix}$$

$$m_\emptyset = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Enfin, on fusionne les masses simples en utilisant la règle de Dempster.

$$m_1 \circledast m_2 = m_f \oplus m_o \oplus m_\emptyset$$

Exemple 1 :

Si on considère, par exemple, les deux fonctions de masse : $m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.8 & 0 & 0.2 \end{bmatrix}$

et $m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.7 & 0 & 0.3 \end{bmatrix}$.

On calcule les poids :

$$w_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 1 & 0.2 & 1 & \end{bmatrix} \text{ et } w_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 1 & 0.3 & 1 & \end{bmatrix}.$$

On utilise le principe d'engagement minimal :

$$w_1(F) \wedge w_2(F) = 0.2 \quad w_1(O) \wedge w_2(O) = 1 \quad w_1(\emptyset) \wedge w_2(\emptyset) = 1$$

d'où :

$$m_f = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.8 & 0 & 0.2 \end{bmatrix}, m_o = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ et } m_\emptyset = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

La fonction de masse résultante $m_1 \circledast m_2 = m_f \circledast m_o \circledast m_\emptyset$ est :

$$m_1 \circledast m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.8 & 0 & 0.2 \end{bmatrix}.$$

Comme la masse m_1 n'apporte pas de nouvelle information, elle n'a aucun effet.

3.4.5. Entropie et spécificité d'une fonction de masse

L'entropie et la spécificité sont deux mesures des fonctions de masse introduites par Yager dans [Yager, 2008].

Entropie En probabilité, l'entropie de Shannon [Shannon, 1948] est définie, pour une variable aléatoire discrète, comme :

$$H_P = - \sum_A P(A) \cdot \ln(P(A))$$

De façon équivalente, on définit l'entropie d'une fonction de masse de la façon suivante :

$$E_m = - \sum_{A \subseteq \Omega} m(A) \cdot \ln(pl(A))$$

Dans le cas qui nous intéresse, l'entropie permet de caractériser une incompatibilité dans les hypothèses supportées par la fonction de masse. Dans le cas d'une fonction de masse simple, c'est-à-dire qui ne compte qu'un seul élément focal, l'entropie est nulle $E_m = 0$.

Avec le cadre de discernement $\Omega = \{F, O\}$, on peut démontrer que E_m admet un maximum $E_m = \ln(2)$ si $m(F) = m(O) = 0.5$. D'autre part, comme E_m est positive, on peut définir l'encadrement suivant :

$$0 \leq E_m \leq \ln(2)$$

Spécificité La spécificité d'une fonction de masse (qui n'a pas d'équivalent en probabilité) est définie comme :

$$S_m = \sum_{\substack{A \subseteq \Omega \\ A \neq \emptyset}} \frac{m(A)}{\text{card}(A)}$$

On peut démontrer que la spécificité évolue dans l'intervalle :

$$\frac{1}{\text{card}(\Omega)} \leq S_m \leq 1$$

Ce qui donne dans le cas où $\Omega = \{F, O\}$:

$$\frac{1}{2} \leq S_m \leq 1$$

Plus S_m est élevée, plus la fonction de masse est dite spécifique. Pour avoir une fonction très spécifique, il faut que $S_m \rightarrow 1$ ce qui implique que $m(O) + m(F) \rightarrow 1$. Dans le cas où on considère uniquement des masses simples, on a alors $m(O) \rightarrow 1$ ou $m(F) \rightarrow 1$. Au contraire, pour avoir une fonction peu spécifique, il faut que $S_m \rightarrow \frac{1}{2}$, ce qui correspond à $m(\Omega) \rightarrow 1$. Une spécificité élevée permet de caractériser le fait que les masses sont principalement attribuées sur des sous ensembles de cardinal faible, ce qui indique généralement que la fonction de masse n'est pas indéfinie, mais elle peut être conflictuelle.

De plus, la spécificité d'une fonction de masse peut être reliée à ses degrés de croyance et de plausibilité.

Soit deux fonctions de masse m_1 et m_2 . On peut dire que m_1 est moins spécifique que m_2 si et seulement si :

$$\begin{aligned} bel_1(A) &< bel_2(A) \quad \forall A \subseteq \Omega \Leftrightarrow \\ pl_1(A) &> pl_2(A) \quad \forall A \subseteq \Omega \end{aligned}$$

Remarque. Les mesures d'entropie et de spécificité permettent de caractériser les fonctions de masse. Une fonction de masse informative et non ambiguë doit avoir une spécificité élevée et une entropie faible.

Exemples Considérons les exemples suivant.

$$\text{Soit } m_1 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.9 & 0 & 0.1 \end{bmatrix} :$$

$$E_{m_1} = 0 \quad S_{m_1} = 0.95$$

Ici m_1 est une masse simple, son entropie est nulle, comme la masse est principalement sur F , cette fonction de masse est très spécifique.

$$\text{Soit } m_2 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.1 & 0.1 & 0.8 \end{bmatrix} :$$

$$E_{m_2} = 0.0211 \quad S_{m_2} = 0.6$$

Dans ce cas, comme les valeurs des masses sur F et O sont faibles, on a alors une entropie faible et une spécificité faible.

$$\text{Pour } m_3 = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.4 & 0.4 & 0.2 \end{bmatrix} :$$

$$E_{m_3} = 0.4087 \quad S_{m_3} = 0.9$$

Dans ce cas, on a une entropie et une spécificité élevées.

3.5. Traitements particuliers possibles dans les grilles évidentielles

Les grille évidentielles offrent une grande richesse de traitements et permettent de manipuler des notions qui n'ont pas d'équivalents en probabilité. Dans ce paragraphe, on présente certains de ces aspects.

3.5.1. Représentation de l'ignorance

Principe L'ignorance dans les grilles évidentielles est **explicitement** représentée par une masse sur Ω . Cette masse représente la part de croyance que l'on n'est

pas capable d'attribuer aux éléments plus spécifiques, ici F et O . Cette ignorance correspond au peu d'information renseignant l'état de la cellule. Un état complètement inconnu se traduit par la fonction de masse *vide* m_Ω qui est définie par $m_\Omega = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Cette fonction de masse à une spécificité minimale : $S_{m_\Omega} = \frac{1}{2}$.

Initialisation L'initialisation de la grille est faite avant toute prise d'information. Sans information à priori (par exemple provenant d'une carte), chaque cellule de la grille est dans un état complètement inconnu, donc initialisée avec une masse m_Ω . Dans le cas où l'on dispose d'informations à priori, on peut initialiser la grille en combinant simplement la grille vide (m_Ω) avec la grille d'à priori, en considérant l'à priori comme une source distincte.

Affaiblissement Si la confiance dans l'information d'une fonction de masse diminue, on peut lui appliquer un mécanisme d'affaiblissement. Cet affaiblissement peut être facilement réalisé à l'aide de l'opérateur \odot , voir [Mercier *et al.*, 2005]. Il consiste à construire une fonction de masse m_α définie par 3.39, puis on réalise la fusion disjonctive avec la fonction de masse à affaiblir m , voir eq. 3.40.

$$m_\alpha = \begin{bmatrix} \emptyset & F & O & \Omega \\ 1 - \alpha & 0 & 0 & \alpha \end{bmatrix} \quad (3.39)$$

$${}^\alpha m = m \odot m_\alpha \quad (3.40)$$

Ceci revient à transférer une part α des masses $m(O)$ et $m(F)$ vers $m(\Omega)$.

On utilisera ce principe d'affaiblissement pour modéliser le vieillissement des informations, par une atténuation temporelle. Cette atténuation est nécessaire lorsqu'on combine une G_t^{SG} avec G_{t-1}^{MG} et lorsqu'une cellule n'est plus observée. Le coefficient α est calculé en fonction d'un modèle de durée de vie exponentiel décroissant, cf équation 3.33, partie 2.4.2.

3.5.2. Identification du conflit à partir de $m(\emptyset)$

Principe On parle de conflit quand deux informations qui se contredisent sont fusionnées. Dans le cadre des fonctions de croyance le conflit est explicitement représenté par la masse sur \emptyset .

Si la fonction de masse issue de la fusion présente une masse $m(\emptyset)$ non nulle, cela traduit que les informations fusionnées sont au moins partiellement en contradiction. A partir d'une contradiction, il est difficile de décider quel est l'état de la cellule.

De même, pour ne pas propager le conflit au travers d'un processus de fusion, on utilise la règle de Dempster ou Yager, mais le résultat intermédiaire de la fusion conjonctive permet de détecter et analyser les situations conflictuelles.

Ce conflit peut avoir différentes causes : une divergence d'opinion d'expert ou une modélisation du système incorrecte. Dans notre cas, le conflit vient principalement d'une erreur de modélisation à deux niveaux :

- nous faisons l'hypothèse de grille statique, alors que la scène observée contient des éléments dynamiques,
- nous faisons des approximations géométriques dues à la discrétisation dans le modèle de capteur et lors de la propagation des grilles.

On propose d'exploiter le conflit pour détecter ces erreurs, notamment les erreurs liées à la dynamique de la scène.

Caractérisation de conflit Lors de la fusion conjonctive, si on analyse le terme de conflit $m(\emptyset)$ issue de la fusion :

$$m_{t-1}^{MG} \oplus m_t^{SG}.$$

On peut le décomposer en deux termes :

$$\begin{aligned} C_1 &= m_{t-1}^{MG}(F) \cdot m_t^{SG}(O) \\ C_2 &= m_{t-1}^{MG}(O) \cdot m_t^{SG}(F) \end{aligned}.$$

Le premier terme C_1 correspond au fait qu'une cellule précédemment libre avec un certain niveau de confiance $m_{t-1}^{MG}(F)$ soit observée occupée à l'instant courant avec le niveau de confiance $m_t^{SG}(O)$. Si on considère que le conflit provient du déplacement des objets dans la scène, le conflit C_1 caractérise le fait qu'une cellule libre devienne occupée et donc qu'un objet vient d'entrer dans cette cellule.

Au contraire, le conflit C_2 caractérise le fait qu'une cellule occupée devienne libre et donc qu'un objet a quitté cette cellule.

Si on s'intéresse au conflit généré sur les cellules dues à un objet en mouvement, on pourrait en analysant les conflits C_1 et C_2 déterminer la direction et le sens du mouvement de l'objet. Ceci est illustré sur la figure 3.7.

D'autre part, le fait de savoir qu'une cellule est occupée parce qu'un objet mobile est rentré dans cette cellule, permettrait de segmenter la grille ou de considérer d'autres états (un exemple est présenté en section 5.2.6).

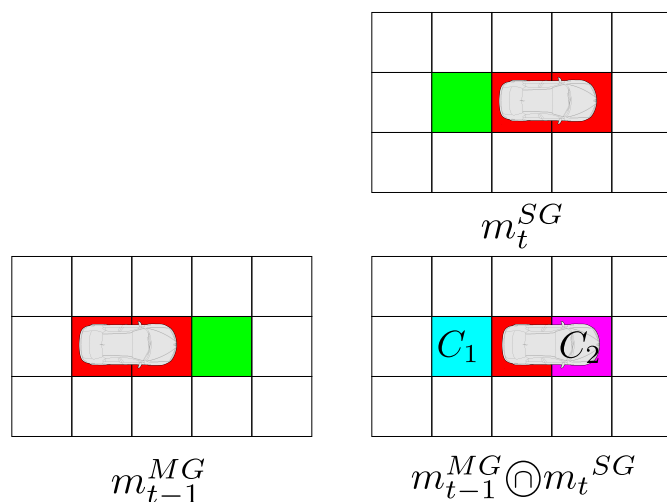


Figure 3.7.: Utilisation du conflit pour caractériser le sens de déplacement d'un mobile (ici de la gauche vers la droite et d'une cellule exactement)

3.5.3. Comparatif des règles de combinaison

La règle de fusion de référence utilisée dans le cadre des fonctions de croyance est la règle de Dempster. Néanmoins, afin de vérifier que cette règle est adaptée à notre problème, nous l'avons comparée aux autres règles dans l'étude suivante.

Les tableaux 3.6, 3.7 et 3.8 résument le résultat de la fusion de deux fonctions de masse m_1 et m_2 .

Table 3.6.: Résultat de la combinaison de deux fonctions de masse coopératives

(a) Exemple 1 : cas de masses fortes coopératives

A	m_1	m_2	$m_1 \odot m_2$	$m_1 \cup m_2$	$m_1 \oplus m_2$	$m_1 \otimes m_2$	$m_1 \wedge m_2$
\emptyset	0	0	0.0	0	0	0	0
F	0.8	0.7	0.94	0.56	0.94	0.94	0.8
O	0	0	0.0	0	0	0	0
Ω	0.2	0.3	0.06	0.44	0.06	0.06	0.2

(b) Exemple 2 : cas de masses forte et faible coopératives

A	m_1	m_2	$m_1 \odot m_2$	$m_1 \cup m_2$	$m_1 \oplus m_2$	$m_1 \otimes m_2$	$m_1 \wedge m_2$
\emptyset	0	0	0.0	0	0	0	0
F	0.4	0.1	0.52	0.04	0.52	0.52	0.4
O	0	0	0.0	0	0	0	0
Ω	0.6	0.9	0.48	0.96	0.48	0.48	0.6

Dans les deux premiers cas, cf tableau 3.6a et 3.6b, on fusionne deux fonctions de

masse ayant toutes les deux une croyance sur le même état : F .

Dans le cas des fonctions basées sur la règle conjonctive (\odot , \oplus , \underline{y}), les résultats sont identiques car il n'y a pas de conflit. Le comportement de ce type de fusion augmente la masse sur F . On remarque que l'augmentation est importante dans l'exemple 1, où les deux masses sont toutes les deux assez équilibrées. Dans l'exemple 2, les masses sont déséquilibrées. L'augmentation par rapport à la masse m_1 est assez faible, cela montre la faible influence de m_2 .

Par contre, dans le cas de la règle disjonctive, c'est la masse de l'état Ω qui augmente. Dans l'exemple 1, malgré le fait que les deux masses fusionnées ont un important niveau de confiance, le résultat est très partagé. Dans l'exemple 2, le résultat est très proche de m_2 . C'est la masse la moins informative qui a le plus d'influence. Cette règle est très utile pour identifier l'information commune.

La règle prudente conserve le même niveau d'information que la source la plus informative, c'est-à-dire m_1 .

Table 3.7.: Résultat de la combinaison de deux fonctions de masse conflictuelles

(a) Exemple 3 : cas de masses conflictuelles

A	m_1	m_2	$m_1 \odot_2$	$m_1 \cup_2$	$m_1 \oplus_2$	$m_1 \underline{y}_2$	$m_1 \wedge_2$
\emptyset	0	0	0.2	0	0	0	0
F	0.5	0	0.3	0	0.375	0.3	0.375
O	0	0.4	0.2	0	0.25	0.2	0.25
Ω	0.5	0.6	0.3	1	0.375	0.5	0.375

(b) Exemple 4 : cas de masses fortement conflictuelles

A	m_1	m_2	$m_1 \odot_2$	$m_1 \cup_2$	$m_1 \oplus_2$	$m_1 \underline{y}_2$	$m_1 \wedge_2$
\emptyset	0	0	0.56	0	0	0	0
F	0.8	0	0.24	0	0.54	0.24	0.54
O	0	0.7	0.14	0	0.32	0.14	0.32
Ω	0.2	0.3	0.06	1	0.14	0.62	0.14

Dans le troisième et le quatrième cas, cf tableau 3.7a et 3.7b, on fusionne la fonction de masse m_1 ayant une croyance sur l'état F avec m_2 qui a une croyance sur O . Les deux sources sont contradictoires.

Dans le cas de la fusion conjonctive, on obtient une masse $m(\emptyset)$ non nulle, ce qui traduit la présence de conflit entre les informations fusionnées. Dans l'exemple 3, le niveau de conflit reste assez faible (0.2), puisque la distribution des bba est assez équilibrée. Le résultat est aussi très équilibré et il est assez difficile de trancher pour un état particulier. Dans le cas de l'exemple 4, le conflit est très important. La présence de conflit est gênante si on veut propager la masse fusionnée pour des traitements ultérieurs, car d'une part le sens de cette masse n'est pas explicite et

d'autre part car cet masse est l'état absorbant de la règle. Les règles de Dempster et de Yager proposent deux méthodes différentes pour annuler ce conflit.

Dans le cas de la règle de Dempster (\oplus), la masse $m(\emptyset)$ est annulée via un processus de normalisation des autres. Dans l'exemple 3, comme dans le cas de la fusion conjonctive, le résultat est assez réparti. Dans l'exemple 4, c'est l'état libre qui prédomine majoritairement. Ce résultat, un peu surprenant, est dû à la normalisation avec la valeur conflit. Ainsi, un écart assez faible avant normalisation devient important si le conflit est prédominant. Cela montre que l'analyse du conflit est une étape préalable très importante pour l'interprétation.

Dans le cas de la règle de Yager (\odot), la masse sur $m(\emptyset)$ est transférée sur $m(\Omega)$. Dans les deux exemples, la masse qui en résulte plaide clairement pour l'ignorance. Le comportement général de cette règle est le suivant : dans le cas d'une confirmation, la confiance augmente, dans le cas contraire, la masse est réattribuée à l'ignorance.

Concernant la règle disjonctive, puisque chaque source n'a aucune information en commun, elle donne une masse vide ($m(\Omega) = 1$).

La règle prudente donne le même résultat que la règle de Dempster car cette dernière, dans ce cas, suit le principe d'engagement minimal.

Considérons maintenant un exemple avec des masses non simples (tableau 3.8), ce qui correspond à une entropie non nulle. Ces masses présentent à la fois une part de compatibilité et une part de conflit.

A	m_1	m_2	$m_1 \odot_2$	$m_1 \oplus_2$	$m_1 \oplus_2$	$m_1 \odot_2$	$m_1 \wedge_2$
\emptyset	0	0	0.31	0	0	0	0
F	0.5	0.2	0.29	0.10	0.42	0.29	0.48
O	0.3	0.5	0.34	0.15	0.49	0.34	0.32
Ω	0.2	0.3	0.06	0.75	0.09	0.37	0.20

Table 3.8.: Résultat de la combinaison de deux fonctions de masse non simple, exemple 5

La fusion conjonctive révèle un conflit important dû à la contradiction des masses et une ignorance spécialement faible car la spécificité des masses est assez importante $S_{m_1} = 0.9$ et $S_{m_2} = 0.85$. De ce fait, les résultats des fusions de Dempster et de Yager sont très indécis. La règle disjonctive, de par la présence de la contradiction, donne une masse très importante sur Ω mais toutefois pas égale à 1 car il existe une part d'information mutuelle cohérente. La règle prudente donne ici un résultat surprenant, avec une masse supérieure pour l'état F alors que les fonctions de masse initiales semblaient plutôt en faveur de O . Ce résultat s'explique par le fait que la masse la plus spécifique est m_1 qui supporte plus l'hypothèse F . Or, la règle prudente privilégie la fonction de masse la plus spécifique.

Cette première analyse a permis d'écarter les règles qui ne sont clairement pas adaptées à notre problème : conjonctive, disjonctive et prudente.

3.5.4. Étude de la dynamique de mise à jour

Comparaison des règles Les différences de comportement des différentes règles de fusion en environnement dynamique ont été mis en avant par le biais d'une simulation. On rappelle que la stratégie de mise à jour est :

$$m_t^{MG} = m_{t-1}^{MG} \otimes m_t^{SG}. \quad (3.41)$$

Cette simulation est réalisée pour une cellule et les résultats sont présentés au travers du graphique 3.8.

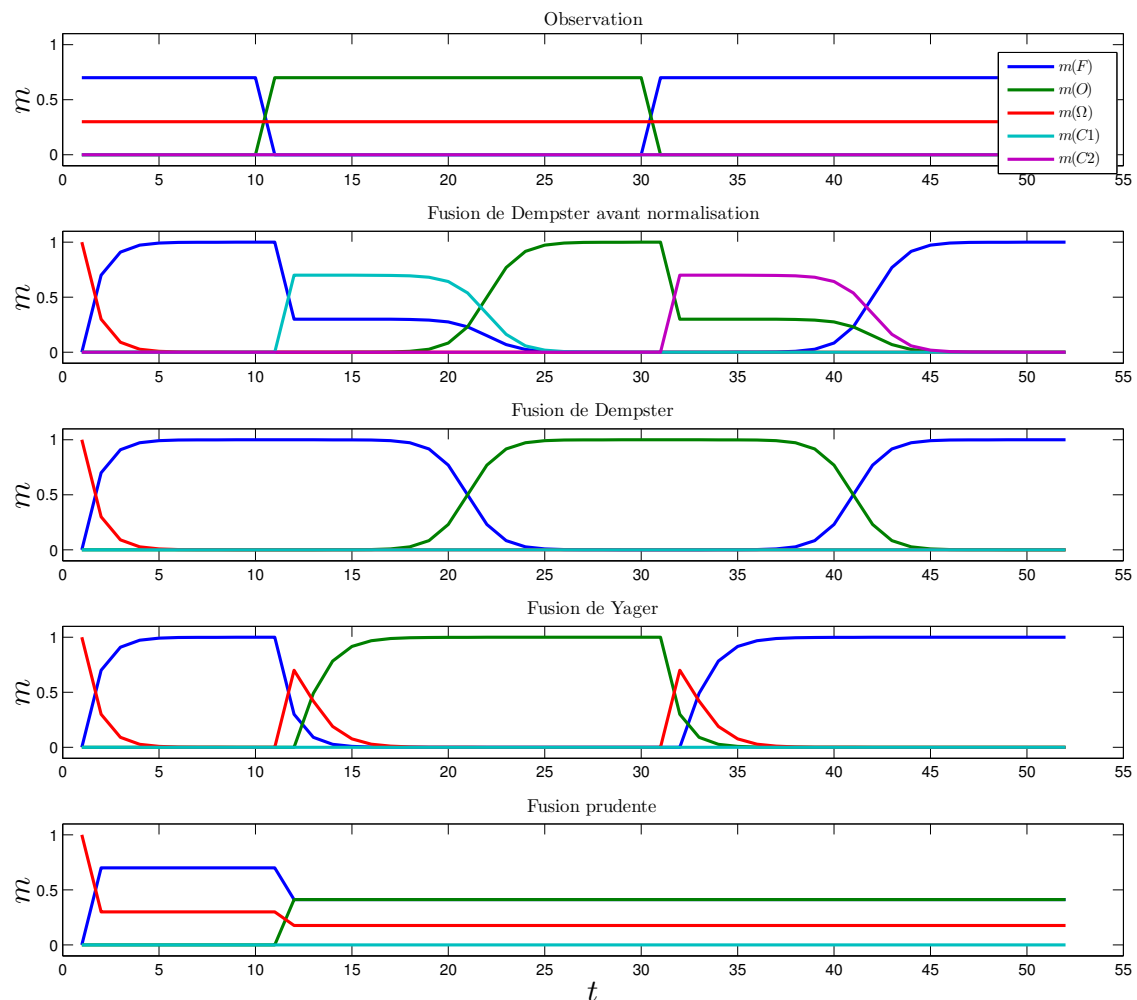


Figure 3.8.: Simulation d'évolution des fonctions de masse issue de la mise à jour temporelle par fusion en utilisant différentes règles. Pour chaque graphe, les courbes représentent les masses sur les différents sous ensemble de Ω , nous avons de plus ici séparé la masse sur \emptyset en indiquant la part sur C_1 (le conflit de F vers O) de la part sur C_2 (le conflit de O vers F)

La première ligne simule la croyance m^{SG} associée à une cellule G_{ij}^{SG} de la ScanGrid qui est construite à partir des mesures à chaque instant de façon indépendante. Chaque graphe suivant représente l'évolution temporelle de la fonction de masse m^{MG} construite à partir de fusion de l'équation 3.41, chaque ligne correspondant à un opérateur \otimes différent.

Dans ce scénario, on considère que la cellule subit une succession de transitions d'état qui simule le passage d'un objet mobile au travers de cette cellule :

- phase 1 : libre de $t = 1$ à 10 avec une fonction de masse $m^{SG} = \begin{bmatrix} 0 & 0.7 & 0 & 0.3 \end{bmatrix}$,
 - phase 2 : occupée de $t = 11$ à 30 avec une fonction de masse $m^{SG} = \begin{bmatrix} 0 & 0 & 0.7 & 0.3 \end{bmatrix}$,
 - phase 3 : libre de $t = 31$ à 51 avec une fonction de masse $m^{SG} = \begin{bmatrix} 0 & 0.7 & 0 & 0.3 \end{bmatrix}$.
- Suivant le type de fusion utilisée, le comportement n'est pas le même.

Sur les graphes 2 et 3, on voit le résultat de la fusion utilisant la règle de Dempster. Le graphe 2 montre le résultat du calcul intermédiaire avant l'étape de normalisation du conflit alors que la ligne 3 montre le résultat après normalisation. La règle de Dempster converge très vite au début car la masse initiale est affectée sur Ω . Par contre lors des transitions, elle accuse un retard important. En effet, la SG bascule à $t = 11$ mais le résultat dans la MG transite seulement vers $t = 21$, dans le graphe 3. Ce retard est directement lié au temps d'observation de la cellule durant la phase 1. En effet, dans le cas présent où l'on utilise des bba symétriques pour l'observation, la cellule ayant été observée 10 fois dans l'état libre, il faut attendre 10 observations de la cellule occupée pour s'apercevoir du changement d'état. Ce problème est un peu similaire au problème de l'accumulation qui a été résolu en bornant l'accumulation et en utilisant des gains différents. Durant ce retard, on voit que la présence du conflit signale une transition d'état dans le graphe 2. Le phénomène est le même lors de la seconde transition. On remarque qu'on est en mesure de distinguer si le conflit est dû à une transition $F \Rightarrow O$ (conflit $C1$) ou $O \Rightarrow F$ (conflit $C2$). Le fait que cette règle ne fasse pas transiter immédiatement permet d'avoir une robustesse face aux erreurs de mesure à la défaveur de la dynamique de mise à jour qui est pénalisée fortement.

Le graphe 4 montre la règle de Yager. Cette règle permet une convergence très rapide aussi bien à l'initialisation que lors des changements d'état. Lors des transitions, la règle fonctionne comme à l'initialisation, la masse étant reportée sur Ω . Cette règle n'est pas dépendante du temps d'observation ce qui la rend sensible aux erreurs de mesure.

Enfin, le dernier graphe montre la règle prudente. L'idempotence de cette règle fait que le résultat n'est influencé que lorsque de nouvelles informations sont fusionnées, c'est-à-dire pour $t = 1$ et pour $t = 11$. Le reste du temps, il s'agit d'informations redondantes, ce qui donne un résultat inchangé. Dans notre cas, cette règle n'est pas adaptée, le résultat produit ne permettant pas d'observer la dynamique de la scène.

Utilisation de l'affaiblissement pour contrôler la dynamique de convergence

Par analogie avec la saturation d'accumulation, introduire un affaiblissement systé-

matique (assez faible) permet d'accélérer la transition d'état. La figure 3.9 montre qu'un affaiblissement (ici $\alpha = 0.05$) est très pertinent pour la fusion de Dempster, puisque la convergence devient quasiment aussi rapide de celle de Yager. Mieux que ça, l'affaiblissement permet de contrôler la vitesse de transition : plus l'affaiblissement est grand, plus la transition est rapide.

En introduisant un affaiblissement, on modifie la valeur maximum atteinte par les combinaisons successives de masses concordantes. En limitant la convergence, on permet une transition plus rapide.

Cet effet d'affaiblissement a un effet négligeable sur la fusion de Yager.

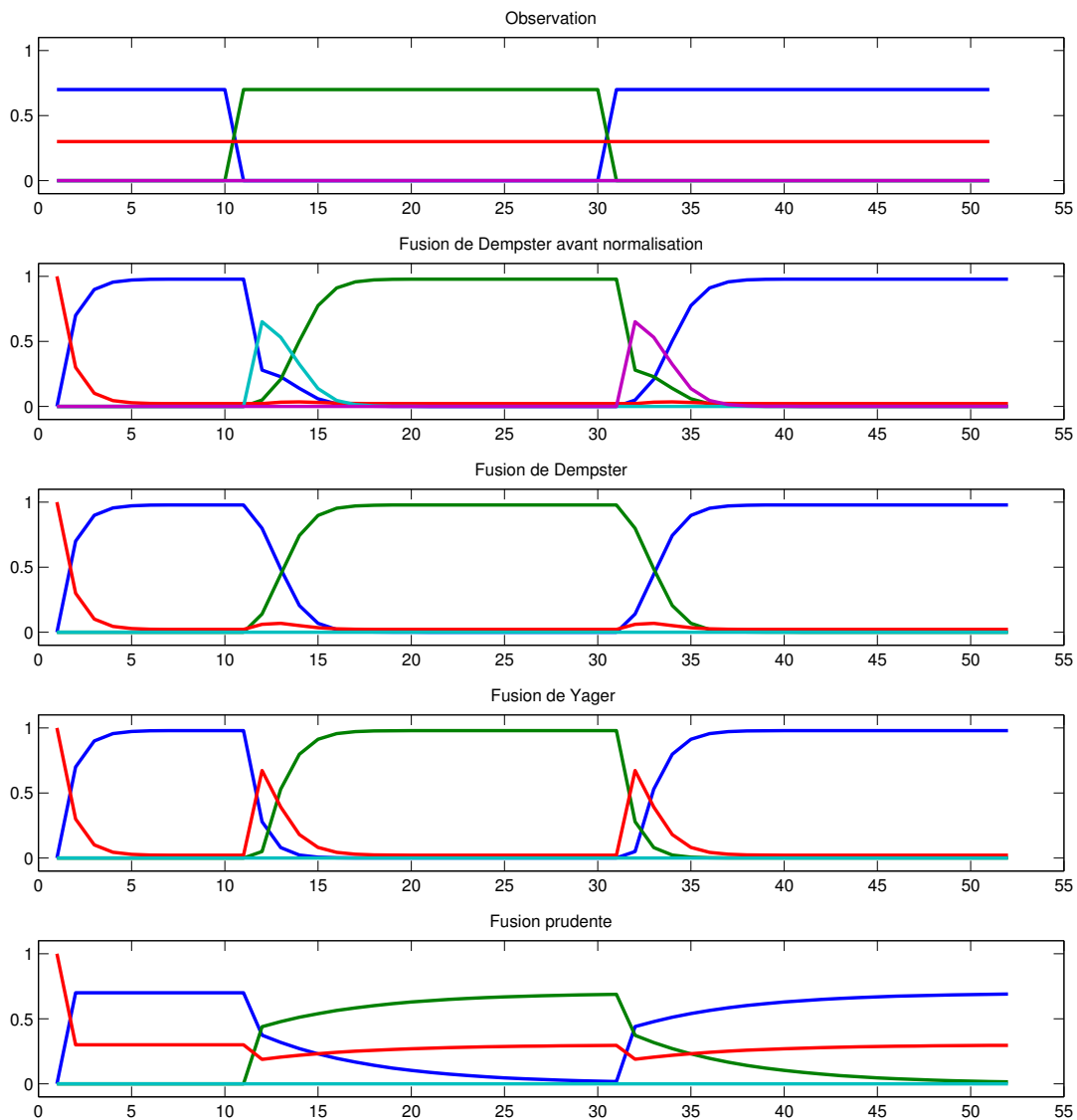


Figure 3.9.: Simulation d'évolution des fonctions de masse avec affaiblissement ($\alpha = 0.05$)

Choix de l'opérateur de fusion L'étude statique de comparaison des opérateurs avait montré que les opérateurs de Dempster et de Yager étaient les plus adaptés à notre problème. Lors de l'étude dynamique, la règle de Yager a montré une bonne vitesse de convergence au dépend d'une sensibilité au bruit. Grâce à un affaiblissement, la règle de Dempster présente l'avantage d'être contrôlable. Cela permet de choisir un compromis entre la dynamique de convergence et la sensibilité aux erreurs, ce qui rend possible l'adaptation de l'approche à l'environnement dans lequel le véhicule évolue en fonction de la qualité des données fournies par les capteurs. Finalement, c'est la règle de référence (Dempster) qui sera utilisé par la suite pour réaliser la fusion de grille.

3.6. Comparaison des cadres probabiliste et évidentiel

Nous allons montrer dans cette section que le cadre évidentiel permet d'une part une meilleure description de l'état des cellules et d'autre part, d'avoir les mêmes performances que la fusion probabiliste en utilisant moins d'hypothèses à priori sur la confiance que l'on a dans les mesures.

En effet, la méthode probabiliste décrit l'occupation d'une cellule par une seule probabilité par cellule, alors que les fonctions de croyance nécessitent le calcul de trois masses (la quatrième étant obtenue par la condition de l'équation 3.25). Le coup calculatoire et l'espace mémoire requis par l'approche évidentielle sont plus importants. Cependant, dans le cadre probabiliste, le transfert de croyance n'est possible qu'entre deux états et ce de façon symétrique. Dans le cadre des fonctions de croyance, la croyance peut être transférée entre quatre états (voir fig. 3.10) et chacune de ces transitions a une signification, une dynamique et une importance différente.

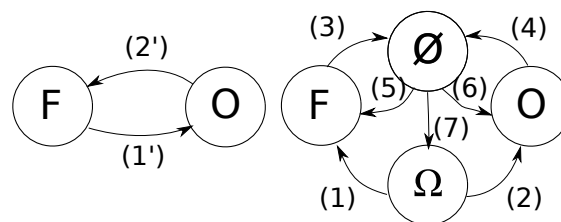


Figure 3.10.: Comparaison des transferts de croyance : A gauche en probabilité et à droite en fonction de croyance en utilisant la règle de Dempster. Dans ce dernier cas, les transitions sont : la fusion (1,2), la génération de conflit (3,4) et la normalisation du conflit (5,6,7)

3.6.1. Représentation de l'ignorance

Dans le cas probabiliste, l'ignorance se traduit par l'utilisation de distributions de probabilité de référence. Si on considère les évènements équiprobables, on utilise une loi uniforme par exemple. On remarque ici que l'on est obligé de faire une hypothèse à priori sur les évènements, postula équiprobable. Dans le cas de la grille d'occupation, cela se traduit par exemple par $P(O) = P(F) = 0.5$. Cependant, lors de la révision de cette probabilité, en utilisant la règle de Bayes par exemple, cette hypothèse de départ va influencer le résultat final.

Dans le cas des fonctions de croyance, nous rappelons qu'un état complètement inconnu se traduit par la fonction de masse *vide* m_Ω qui est définie par $m_\Omega = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Cette fonction de masse est très différente de la fonction de masse $m = \begin{bmatrix} \emptyset & F & O & \Omega \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix}$ qui signifie que la cellule a été observée partiellement vide et partiellement occupée.

La différence de modélisation entre les cadres probabiliste et évidentiel est particulièrement distinguable lors de l'initialisation des grilles, comme le montre la figure 3.11. Si on n'utilise pas d'information à priori, on initialise avec une loi uniforme dans le cas probabiliste ou avec une ignorance totale dans le cas évidentiel.

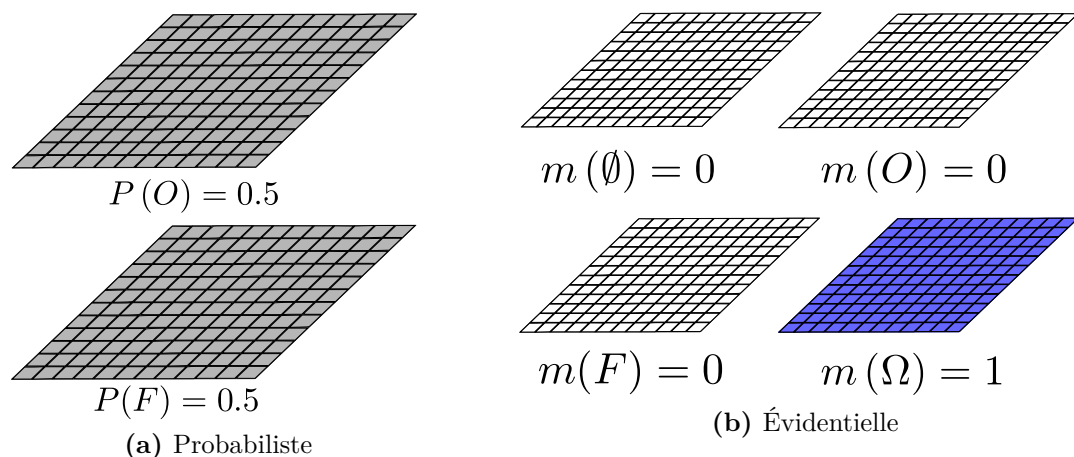


Figure 3.11.: Initialisation d'une grille d'occupation

Affaiblissement Dans le cadre probabiliste, pour mettre en place un affaiblissement, il faut faire converger la probabilité vers une distribution uniforme, c'est-à-dire une valeur de 0.5, cf. équation 3.42. Cela pose une ambiguïté, puisqu'affaiblir ou fusionner avec une information en conflit devient équivalent dans ce formalisme.

$$P = (1 - \alpha) \cdot (P - 0.5) + 0.5 \tag{3.42}$$

La figure 3.12 présente une simulation dans laquelle la cellule est libre, puis elle n'est plus observée et enfin elle est de nouveau observée libre. Cela met en avant le comportement du phénomène d'affaiblissement et permet de comparer les approches probabiliste et évidentielle.

Si on n'a plus d'observation, au fur et à mesure que le temps s'écoule, l'état de la cellule n'est plus pertinent.

Avec l'approche évidentielle, on peut remarquer un instant particulier ($t \simeq 24$) où le niveau de masse sur Ω dépasse le niveau sur F . A partir de cet état, la croyance sur l'ignorance est clairement supérieure à la croyance sur l'état F . Dans le cas probabiliste, comme la probabilité converge vers 0.5, il est très difficile de savoir à partir de quel niveau l'information contenue dans la cellule n'est plus pertinente. Il faut définir des seuils intermédiaires dans les intervalles $[0 \ 0.5]$ et $[0.5 \ 1]$, dont les valeurs sont difficiles à fixer.

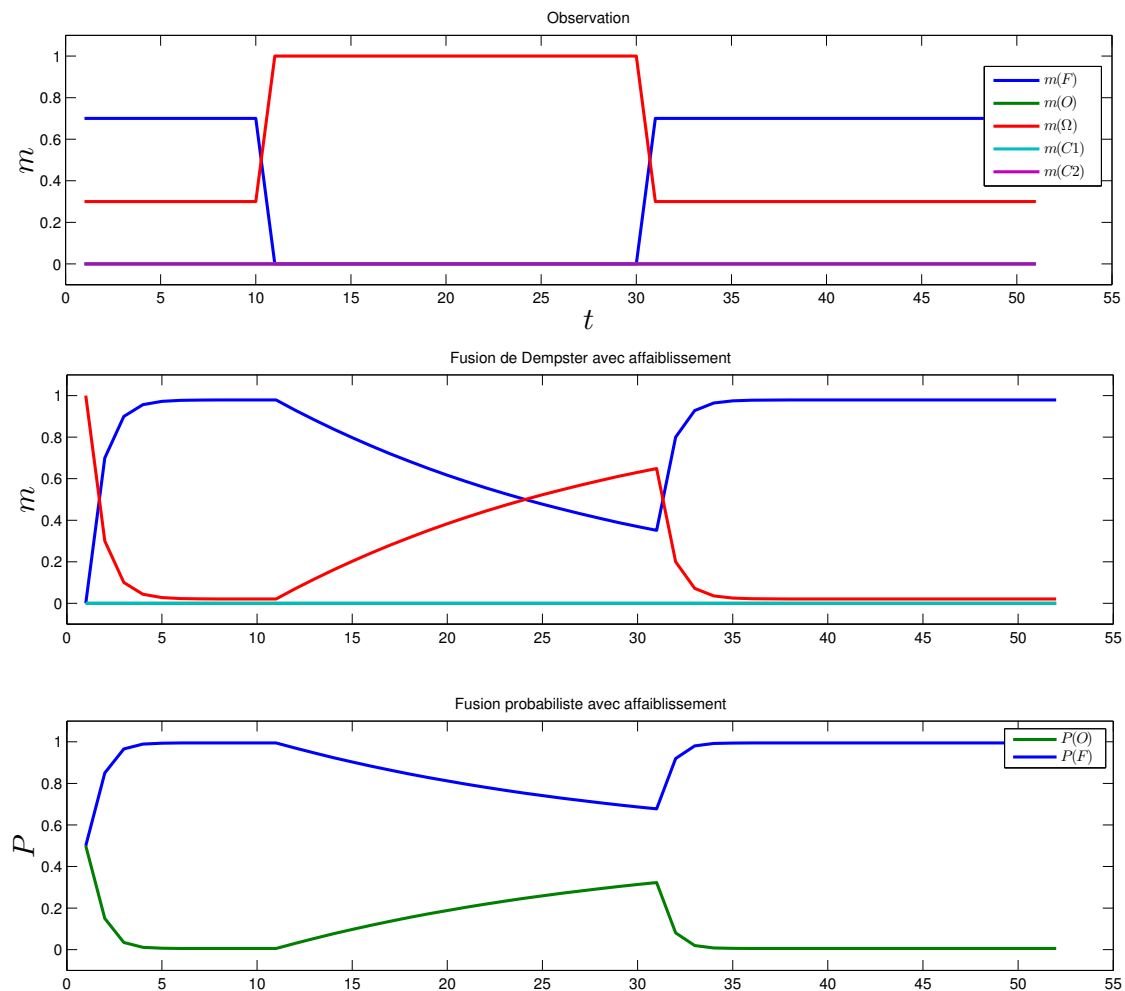


Figure 3.12.: Effet de l'affaiblissement $\alpha = 0.05$

Représentation explicite de l'ignorance avec des probabilités. Certaines approches, par exemple [Baba, 2007], utilisent un cadre bayésien composé de trois états indépendants : occupé, libre ou inconnu $\Psi = \{O, F, I\}$. Il y a donc 3 probabilités par cellule c dont la somme égale 1. L'approche de mise à jour de grille suit un schéma classique prédicteur - correcteur.

La phase de prédiction se déroule de la façon suivante :

$$\hat{P}(O_t^c) = \sum_{B \in \Psi} P(O_t^c | B_{t-1}^c) \cdot P(B_{t-1}^c)$$

$$\hat{P}(F_t^c) = \sum_{B \in \Psi} P(F_t^c | B_{t-1}^c) \cdot P(B_{t-1}^c)$$

Les probabilités $P(A_t^c | B_{t-1}^c)$ correspondent aux probabilités de transition d'états et les probabilités $P(B_{t-1}^c)$ correspondent aux probabilités à priori.

La phase de correction utilise les équations suivantes :

$$P(O_t^c) = P(O_t^c | z) = P(z | O_t^c) \cdot \hat{P}(O_t^c) \cdot P_{dist}$$

$$P(F_t^c) = P(F_t^c | z) = P(z | F_t^c) \cdot \hat{P}(F_t^c)$$

Les probabilités $P(z | A_t^c)$ sont issues du modèle de capteur direct et les probabilités $\hat{P}(A_t^c)$ sont les probabilités prédites, calculées dans la phase de prédiction. P_{dist} est une densité de probabilité déterminée à priori qui permet de pondérer le résultat en fonction de la distance.

Finalement, la probabilité sur l'inconnu est déterminée par normalisation :

$$P(I_t^c) = 1 - (P(F_t^c) + P(O_t^c)).$$

La représentation explicite de l'ignorance par des probabilités permet de résoudre les défauts que nous avons mis en avant. En effet, cette représentation permet de faire la distinction entre les cellules dont l'état est inconnu et les cellules dont l'état est contradictoire. Elle permet d'initialiser et d'affaiblir les grilles sans utiliser d'hypothèse à priori (équiprobable).

Cependant, on peut apporter les critiques suivantes :

- On peut remarquer que l'état inconnu, n'est ni propagé pendant la phase de prédiction, ni estimé durant la phase de correction. De plus, il faut déterminer, à priori, les six probabilités de transitions.
- Contrairement aux fonctions de croyance où $\Omega = \{F, O\}$, l'état I n'a aucun lien avec F et O . Si physiquement une cellule peut être soit occupée soit libre, l'état inconnu exprimé par $P(I)$ n'est pas explicite car les états dans le cadre probabiliste sont exclusifs.
- De plus, la mesure d'entropie probabiliste perd son sens car le maximum se trouve pour une probabilité uniforme $P(O) = P(F) = P(I) = \frac{1}{3}$ qui n'est pas l'état le plus contradictoire.

3.6.2. Masse conflictuelle

On parle de conflit quand deux informations qui se contredisent sont fusionnées. Dans le cadre des probabilités, le conflit et l'ignorance sont représentés de façon identique, avec une probabilité distribuée entre libre et occupé comme illustré sur la figure 3.13.

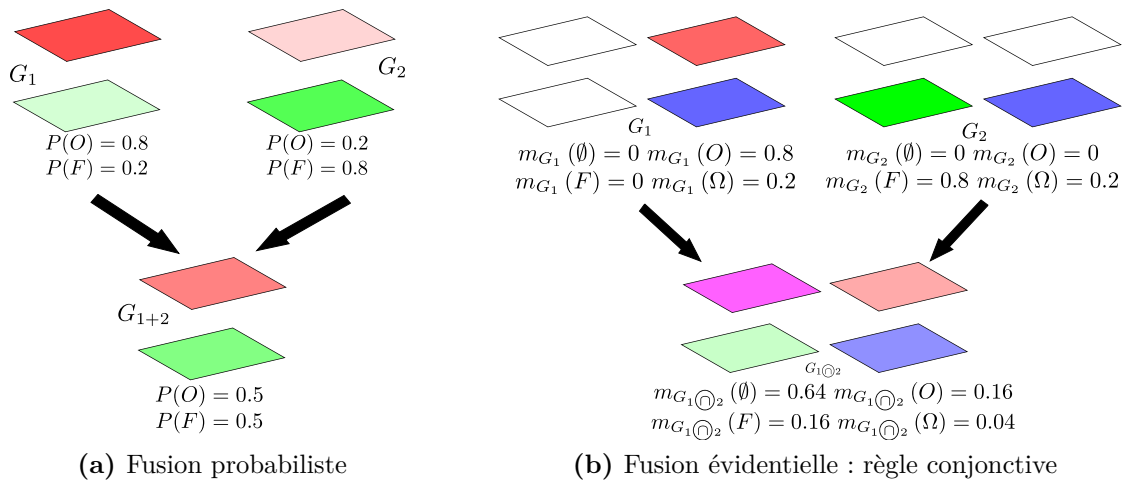


Figure 3.13.: Fusion de deux cellules en conflit.

Dans le cadre des fonctions de croyance le conflit est explicitement représenté par la masse sur \emptyset . En effet, cet état indique, après une fusion conjonctive, qu'aucun état compatible avec la combinaison de **toutes** les masses n'a été trouvé (parmi les états possibles). Donc, si la fonction de masse issue de la fusion présente une masse $m(\emptyset)$ non nulle, cela traduit que les informations fusionnées sont en contradiction, au moins partiellement. A partir d'une contradiction, il est difficile de décider pour un état, encore plus que dans le cas de l'ignorance. Contrairement au cas où on a une ignorance forte qui peut être levée par une mesure très spécifique, dans le cas de conflit, il faut attendre un ensemble de mesures confirmant une hypothèse pour décider.

Ce conflit peut avoir différentes causes : une divergence d'opinion d'expert ou une modélisation du système qui n'est pas correcte par exemple.

Dans notre cas, le conflit vient principalement d'une erreur de modélisation à deux niveaux :

- Nous faisons l'hypothèse de grille statique. Or lorsqu'il y a du mouvement, cela génère du conflit.
- Des erreurs d'approximations géométriques dues à la discrétisation dans le modèle de capteur et lors de la propagation des grilles.

Pour ne pas propager le conflit, on utilise la règle de Dempster. Le résultat final est identique au résultat probabiliste, mais le résultat intermédiaire de la fusion

conjonctive permet de détecter les situations conflictuelles. On peut aussi utiliser la règle de Yager car c'est une règle avec une base conjonctive qui ne génère pas de conflit. Cependant et comme nous l'avons déjà vu, cette règle est très sensible et cela pose un problème dans le cas de mesure bruitée. Enfin, cette règle n'est pas associative. Si cela n'est pas trop gênant pour la fusion temporelle, ça peut l'être dans la cas de la fusion multi-capteurs, puisque l'ordre de fusion va alors avoir un impact déterminant.

3.6.3. Comparaison des comportements de mise à jour

La figure 3.14 reprend la simulation de la partie 3.5.4 illustrant le comportement de la fusion entre la fusion évidentielle et probabiliste. Dans le cas évidentiel, on utilise la règle de Dempster. Les fonctions de masse utilisées sont les mêmes que dans l'exemple de la figure 3.8.

Pour le cas probabiliste, nous avons comparé deux cas :

1. Dans le premier cas, on considère l'état principal de la cellule et on affecte des valeurs de probabilité égales à la masse sur le sous ensemble de cet état : pour la phase 1 et 3 $P(F) = m(F) = 0.7$ et pour la phase 2 $P(O) = m(O) = 0.7$.
2. Dans le second cas, on considère la transformation pignistique pour définir les probabilités : phase 1 et 3 $P(F) = BetP(F) = 0.85$ et phase 2 $P(O) = BetP(O) = 0.85$.

Sur la figure 3.14, la première ligne représente les masses de la *SG*. La seconde ligne montre le résultat de la fusion évidentielle. Enfin, les lignes 3 et 4 montre les résultats de la fusion probabiliste dans les cas 1 et 2, décrits ci-dessus.

La fusion de Dempster et la fusion probabiliste sont très similaires et ont le même type de comportement. Elles sont même quasi équivalentes si on utilise les probabilités pignistiques pour les probabilités d'entrée de fusion dans le cas où il n'y a pas de conflit. Si il y a présence de conflit, ces règles ne sont plus complètement équivalentes.

3.6.4. Choix du mode de décision

Dans les tableaux ci-dessous, nous reprenons les exemples de la section 3.5.3, en comparant les résultats probabiliste et évidentiel (Dempster) au niveau pignistique. Pour chaque exemple, on trouve aussi l'information au niveau crédal pour les fusions conjonctive et de Dempster.

Pour définir les probabilités d'entrée utilisées dans la fusion probabiliste, nous avons utilisé la probabilité pignistique associée aux fonctions de masse, ce qui correspond au cas 2 de la section précédente. On peut remarquer sur chacun des tableaux les résultats identiques entre $BetP(m_1) \oplus BetP(m_2)$ et $BetP(m_{1 \oplus 2})$ (aux erreurs d'arrondi). On peut aussi remarquer que les niveaux de probabilité pignistique associés aux

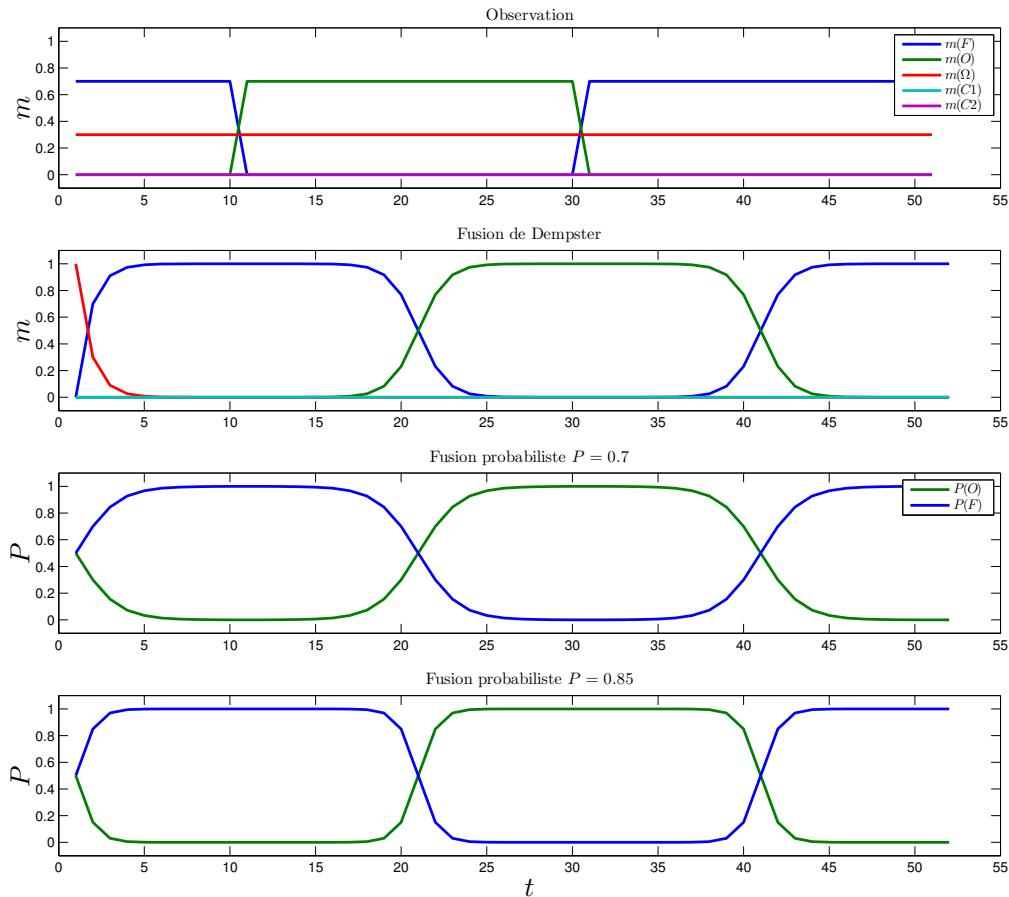


Figure 3.14.: Comparaison du comportement de différentes fusions.

masses sont assez élevés. Dans un contexte de fiabilité, il est plus facile de dire qu'un élément fonctionne bien 70% du temps et ne rien dire pour les 30% restant plutôt qu'un élément fonctionne bien 85% du temps et donne un résultat contraire 15% du temps. Pour obtenir les mêmes résultats, le cadre probabiliste fait des hypothèses plus fortes sur la confiance des sources.

Nous allons voir comment le niveau crédal et les intervalles $[bel \ pl]$ sont intéressants pour estimer la fiabilité des décisions.

Table 3.9.: Comparaison des approches évidentielle et bayésienne : cas de sources coopératives

(a) Exemple 1 : Sources fortes

Niveau crédal				
	m_1	m_2	$m_1 \odot m_2$	$m_1 \oplus m_2$
\emptyset	0	0	0.0	0
F	0.8	0.7	0.94	0.91
O	0	0	0.0	0
Ω	0.2	0.3	0.06	0.09

Niveau pignistique						
	$BetP_{m_1}$	$BetP_{m_2}$	$BetP_{m_1} \odot BetP_{m_2}$	$BetP_{m_1 \oplus m_2}$	$bel_{m_1 \oplus m_2}$	$pl_{m_1 \oplus m_2}$
F	0.9	0.85	0.98	0.97	0.91	1
O	0.1	0.15	0.02	0.03	0	0.09

(b) Exemple 2 : Sources forte et faible

Niveau crédal				
	m_1	m_2	$m_1 \odot m_2$	$m_1 \oplus m_2$
\emptyset	0	0	0.0	0
F	0.4	0.1	0.46	0.46
O	0	0	0.0	0
Ω	0.6	0.9	0.52	0.52

Niveau pignistique						
	$BetP_{m_1}$	$BetP_{m_2}$	$BetP_{m_1} \odot BetP_{m_2}$	$BetP_{m_1 \oplus m_2}$	$bel_{m_1 \oplus m_2}$	$pl_{m_1 \oplus m_2}$
F	0.7	0.55	0.74	0.72	0.46	1
O	0.3	0.45	0.26	0.28	0	0.52

Dans l'exemple 1 (tableau 3.9a 3.9), la probabilité pignistique associée à F est quasi certaine (0.97). Cette probabilité est associée à une croyance forte (0.9) et à un intervalle $[bel \ pl]$ de longueur 0.09 indique que la probabilité $P(F) = 0.97$ est très certaine. Dans ce cas, il n'y a aucun problème pour la prise de décision.

Dans l'exemple 2 (tableau 3.9b), même si la probabilité $P(F) = 0.72$ tend vers F , la borne inférieure $bel(m_1 \oplus m_2)$ est seulement à 0.46 ce qui laisse plus de 50% de risque d'erreur. La taille de l'intervalle $[bel \ pl]$ montre que l'information est très incertaine.

Le conflit est lui aussi un élément important. Si on ne considère que les probabilités pignistiques, la différence entre les résultats de fusion des exemples 2, 3 et 4 est relativement faible. L'information crédale nous montre que cela ne correspond pas à la même situation.

Dans l'exemple 2, la plausibilité associée à F est égale à 1 ce qui montre qu'aucune information s'oppose à cette hypothèse, ce qui est le contraire pour les exemples 3 et 4 où la plausibilité baisse. Dans ces deux exemples, on note aussi la présence d'une masse $m_1 \odot_2 (\emptyset)$ non nulle.

Table 3.10.: Comparaison des approches évidentielle et bayésienne : cas de sources conflictuelles

(a) Exemple 3 : Conflit faible

Niveau crédal				
	m_1	m_2	$m_1 \odot_2$	$m_{1 \oplus 2}$
\emptyset	0	0	0.2	0
F	0.5	0	0.3	0.375
O	0	0.4	0.2	0.25
Ω	0.5	0.6	0.3	0.375

Niveau pignistique						
	$BetP_{m_1}$	$BetP_{m_2}$	$BetP_{m_1} \odot BetP_{m_2}$	$BetP_{m_{1 \oplus 2}}$	$bel_{m_{1 \oplus 2}}$	$pl_{m_{1 \oplus 2}}$
F	0.75	0.3	0.56	0.56	0.375	0.75
O	0.25	0.7	0.44	0.44	0.25	0.625

(b) Exemple 4 : Conflit fort

Niveau crédal				
	m_1	m_2	$m_1 \odot_2$	$m_{1 \oplus 2}$
\emptyset	0	0	0.56	0
F	0.8	0	0.24	0.54
O	0	0.7	0.14	0.32
Ω	0.2	0.3	0.06	0.14

Niveau pignistique						
	$BetP_{m_1}$	$BetP_{m_2}$	$BetP_{m_1} \odot BetP_{m_2}$	$BetP_{m_{1 \oplus 2}}$	$bel_{m_{1 \oplus 2}}$	$pl_{m_{1 \oplus 2}}$
F	0.9	0.15	0.61	0.61	0.54	0.68
O	0.1	0.85	0.39	0.39	0.32	0.46

Dans l'exemple 3 (tableau 3.10a), les deux informations sont en contradiction, mais les fonctions de masses d'entrée sont moyennement spécifique $S_{m_1} = 0.75$ et $S_{m_2} = 0.7$. Dans ce cas, le résultat de fusion est mitigé et les intervalles $[bel \ pl]$ montrent une certaine ignorance sur l'état.

Dans l'exemple 4 (tableau 3.10b), les deux informations sont très contradictoires. Cela peut se voir au niveau pignistique par l'importante masse sur \emptyset et la faible amplitude des intervalles $[bel \ pl]$. Ici, on est typiquement dans une situation où

l'état de la cellule a certainement changé car les informations issues des sources sont fiables et il y a présence de conflit. On peut remarquer que même si la probabilité de F est plus faible dans l'exemple 3 que dans l'exemple 4, la plausibilité de F est plus importante dans ce cas.

La prise en compte du conflit et l'utilisation de la croyance $bel_{m_{1\oplus 2}}$ peuvent permettre d'adapter la prise de décision en analysant les différentes mesures de croyance au niveau crédal.

3.7. Conclusion

Ce chapitre a été consacré à la présentation de méthodes de gestion de l'incertitude dans les grilles de perception. Nous avons présenté les principaux concepts et outils des cadres d'accumulation, probabiliste et évidentiel. Leur utilisation pour la perception robotique à base de grilles a été présentée et adaptée à la même architecture de fusion décrite dans le chapitre 2. La stratégie proposée permet de rendre envisageable l'utilisation de fonctions de croyance en temps réel dans un système embarqué.

Nous avons mis en évidence la plus grande richesse du formalisme de l'approche évidentielle. Nous avons aussi noté que ce cadre généralise bien le cadre probabiliste. Parmi les outils évidentiels, on peut mettre en avant les techniques d'affaiblissement qui permettent d'utiliser de façon efficace l'opérateur de fusion de Dempster dans un contexte dynamique. Enfin, les études comparatives réalisées en simulation sur des cas simples, montrent la faisabilité de l'approche évidentielle. Nous avons observé (de la même façon que [Rey et Carlos, 2006]) que cette approche permet une convergence plus rapide. C'est cette approche qui a notre préférence pour gérer les incertitudes dans les grilles d'occupation. En effet, même si celui-ci est plus complexe à mettre en œuvre, il offre de nombreux avantages qui nous semblent importants vis-à-vis des applications visées. Le tableau 3.11 résume ces avantages.

	Facilité calculatoire	Fusion source fiable	Fusion source non fiable	Fonction d'oubli
Probabiliste	+	+	-	0
Évidentiel	-	+	+	+

	Prise de décision	Construction des connaissances initiales	Détection du conflit	A priori
Probabiliste	0	0	-	+
Évidentiel	+	+	+	+

Table 3.11.: Récapitulatif des avantages et inconvénients des approches probabiliste et évidentielle.

4. Construction d'une *ScanGrid* à partir d'un lidar

Sommaire

4.1. Introduction	87
4.2. Télémètre laser à balayage	88
4.3. Modèle de capteur lidar	95
4.4. Conclusion	112

4.1. Introduction

Dans ce chapitre, nous présentons une méthode de construction de la *ScanGrid*, grille instantanée de perception. La *ScanGrid* est construite à chaque instant où le capteur de perception fournit une nouvelle mesure. Elle traduit l'information du capteur dans la représentation de grille. Elle implémente donc, sous une certaine forme, le modèle de capteur utilisé dans l'algorithme. Nous proposons différents modèles utilisant un unique télémètre laser à balayage multi-échos et multi-nappes. Ce capteur est intégré au pare-choc avant du véhicule. Du fait de sa couverture angulaire limitée (160°) et de sa faible hauteur par rapport au plan de route, cette configuration favorise les masquages et les réflexions spéculaires.

Dans ce chapitre, nous allons nous intéresser dans un premier temps au fonctionnement des lidars. Nous verrons le principe de fonctionnement mais aussi les spécificités des lidars utilisés pour les applications automobiles comme les technologies multi-échos et multi-nappes. Puis, nous décrirons la construction d'une *ScanGrid* à partir de données lidar en utilisant un modèle de capteur inverse. Nous proposerons plusieurs modèles permettant de prendre en compte les effets de masquage au sein de la grille. Nous montrerons alors comment créer une *ScanGrid* en temps réel en utilisant un modèle simplifié et comment créer une *ScanGrid* plus précise par la fusion indépendante de l'information de chaque écho.

4.2. Télémètre laser à balayage

Le télémètre laser aussi nommé LIDAR (Light Detection And Ranging) par analogie au radar, est un capteur à observations télémétriques (mesures de distances). A la différence du radar qui utilise une technologie radio, il utilise une technologie optique laser.

Les capteurs de type lidar sont utilisés dans un grand nombre d'applications couvrant plusieurs domaines :

- pour l'étude de l'atmosphère,
- pour la réalisation de Modèles Numériques de Terrain (MNT),
- pour la topographie,
- pour la robotique : perception de l'environnement.

Le principe physique de ce type de capteur est le suivant. Le capteur émet un faisceau laser dans une direction particulière et mesure l'écho retour. A partir de l'écho retour, on peut extraire différentes informations physiques liées à la propagation du laser.

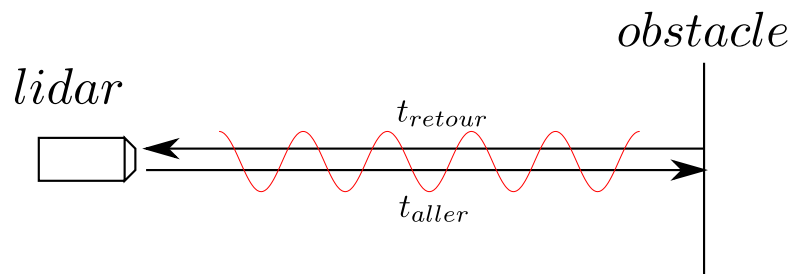


Figure 4.1.: Principe de mesure lidar

Dans le cas qui nous intéresse, l'objectif est la détection des obstacles dans l'environnement du robot et la caractérisation de l'espace roulant. L'écho retour est alors utilisé pour détecter la présence d'un obstacle et mesurer la distance entre le capteur et cet obstacle. Le principe du *temps de vol* est généralement utilisé pour réaliser cette mesure mais certains modèles utilisent une technique de mesure à différence de phase ([Kneip *et al.*, 2009]) que nous ne détaillerons pas ici. Pour réaliser la mesure, le laser émet une impulsion à l'instant t_e . S'il y a présence d'un obstacle à réflexion diffuse, une partie de l'énergie lumineuse est alors réfléchi vers le capteur à l'instant t_r . Connaissant la célérité de la lumière c , la mesure du temps de l'aller retour de l'onde lumineuse $\Delta t = t_r - t_e$ permet de déduire la distance d (cf. équation 4.1). Le laser se propage suivant la droite qui correspond à son axe optique, cette droite est généralement paramétrée par ses orientations, les angles φ et ψ (on suppose que ces angles sont connus avec précision). A partir de cette direction et de la distance de l'écho, on peut calculer le point d'impact dans l'espace 3D par ses coordonnées pseudo-sphériques (d, φ, ψ) .

$$d = \frac{c \cdot \Delta t}{2} \quad (4.1)$$

En réalité le phénomène est plus complexe. Les imperfections du laser et les différentes propriétés de la surface incidente ont des effets importants sur la mesure. Cela introduit des incertitudes sur l'existence ou non d'un obstacle, sur la distance mesurée ainsi que sur la l'orientation angulaire du laser. En effet, l'impulsion lumineuse à une durée non négligeable, la divergence du laser n'est pas nulle, la surface incidente n'est en général pas orthogonale au laser et ses propriétés de réflexion peuvent être très diverses.

Il existe deux types de réflexion en ce qui concerne la lumière. La réflexion peut être soit spéculaire, soit diffuse. La réflexion finale est composée par ces deux réflexions, dont les proportions dépendent de la surface réfléchissante, de la longueur d'onde et de l'angle d'incidence du faisceau.

4.2.1. Modèle de faisceau laser et mesure

En réalité, l'impulsion émise du laser n'est pas parfaite et le signal n'est donc pas un créneau idéal. Spatialement, l'impact du laser n'est pas un point mais une tache formée par l'intersection du cône d'ouverture du laser avec l'environnement. Les faisceaux lumineux réfléchis mettent donc plus ou moins de temps à parcourir la distance en fonction de la position d'impact dans la tache. Ces déformations spatiales ont tendance à étaler le signal.

Dispersion spatiale En optique, un faisceau gaussien (c.f. figure 4.2), est une solution particulière de l'équation de propagation de Helmholtz (au même titre qu'une onde plane) dans le cadre de l'approximation paraxiale. Ce modèle produit une bonne description des rayonnements cohérents utilisés par les faisceaux lasers ([Optics, 2006]). Il considère que l'intensité du faisceau suit une distribution gaussienne suivant chacun de ses axes radiaux. Cette distribution est représentée sur la figure 4.2a. Par conséquent, et à partir des principes de conservation énergétique, on peut calculer l'intensité en fonction de la distance comme le montre l'équation 4.2.

$$I(r, z) = I_0 \left(\frac{w_0}{w(z)} \right)^2 e^{-\frac{2r^2}{w^2(z)}} \quad (4.2)$$

Le faisceau suit alors l'ouverture théorique représentée par la courbe rouge sur la figure 4.2b. Pour des distances suffisamment grandes, cette ouverture peut être approximée par un cône d'angle fixe et c'est l'usage courant en robotique :

$$\tan \left(\frac{\Theta}{2} \right) = \frac{\lambda}{\pi \cdot w_0} \quad (4.3)$$

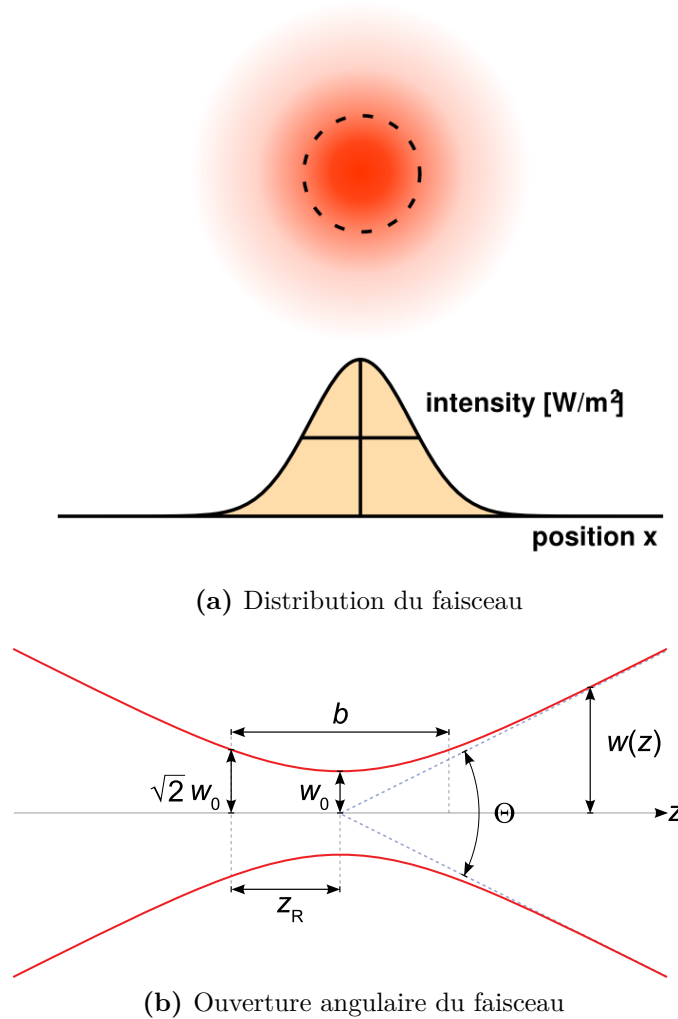


Figure 4.2.: Modèle de faisceau gaussien

Le tableau 4.1 montre le diamètre de l’empreinte du laser qui a une ouverture de 0.25° en fonction de la distance.

La cellule de réception est équipée d’un capteur optique de type photo-transistors. Le signal optique de retour est converti en tension, puis ce signal est comparé à un seuil comme le montre le chronogramme 4.3. Si le signal dépasse le seuil, le capteur retourne un écho dans la direction du laser à une distance correspondant au retard temporel de ce dépassement. Certains capteurs sont capables de détecter plusieurs échos à partir du même signal de retour. On parle alors de technologie multi-échos. Cette technologie, permet de détecter des obstacles même lors de présence de réflexions parasites causées par des gouttes de pluie par exemple ou par la réflexion partielle du laser.

(m)	15	50	100
(cm)	6.5	22	44

Table 4.1.: Taille de la tache laser en fonction de la distance pour une ouverture angulaire de 0.25°

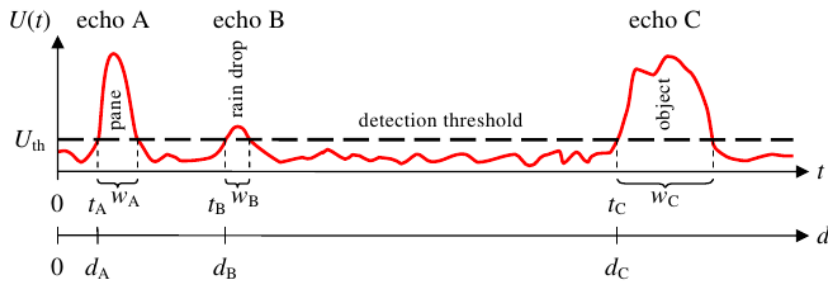


Figure 4.3.: Signal de retour d'un lidar multi-échos (source : IBEO Alasca user manual).

4.2.2. Amélioration du lidar pour les applications robotiques

Pour l'utilisation en robotique mobile, un ensemble d'améliorations ont été apportées au principe de base du lidar pour obtenir une bonne caractérisation de l'environnement.

Balayage angulaire Afin de fournir des mesures dans un plan, un miroir rotatif incliné est souvent installé en sortie du bloc d'émission/réception (E/R) comme indiqué dans la figure 4.4. Ce miroir est actionné par un moteur à une fréquence de rotation donnée et réglable. Le capteur procède alors à une succession de mesures, ce qui permet d'avoir une coupe laser suivant un plan défini par l'angle entre le miroir et le laser et par l'axe de rotation du miroir. Chaque rotation permet d'obtenir un *scan* dont la zone de couverture s'étend généralement à environ 180° . Les fréquences de rotation usuelles sont de l'ordre de la dizaine de tours par minute. En général, on date le scan¹ entier avec une seule date qui correspond au début du scan. Ceci peut poser des problèmes dans le cas où la dynamique de l'ego-véhicule ou de l'environnement sont importantes. Par exemple, soit un ego-véhicule se déplaçant à 10 m/s et on considère que le capteur a une ouverture de 180° et qu'il fonctionne à 10 Hz. Entre le premier et le dernier point, il s'est écoulé un temps $\Delta t = \frac{0.1}{2} = 0.05$ s ce qui correspond à un déplacement du véhicule de l'ordre de 0.5 m. L'importance de ce phénomène, qui est sensiblement lié à la vitesse de déplacement du porteur par rapport à la fréquence d'acquisition du capteur, n'est pas négligeable pour certaines

1. Un scan est un ensemble de mesures fournies par le capteur suite à un balayage.

applications. Les distorsions créées dans les scans par ces phénomènes sont bien connues des gens du domaine.

Pour corriger ce phénomène, on peut réaliser la compensation du mouvement dans le scan. Pour cela, il faut connaître le mouvement de l'ego-véhicule, ce qui permet de calculer la position de l'ego-véhicule à chaque instant où on a une mesure (interpolation de position) de façon à replacer tous les points dans un repère commun, qui est par exemple le repère initial.

Enfin, on peut constater le même phénomène sur les obstacles en mouvement. Par contre dans ce cas, à moins de connaître les paramètres dynamiques de l'obstacle, on ne peut pas réaliser de compensation.

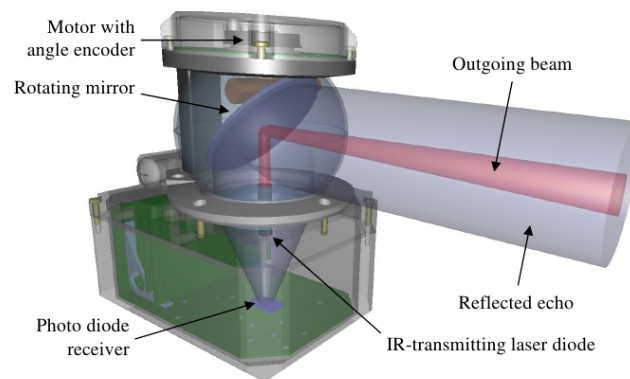


Figure 4.4.: Balayage angulaire du laser utilisant un miroir rotatif (source : IBEO Alasca user manual).

Multi-nappes Afin d'augmenter le nombre de mesures, on peut augmenter le nombre de plans en dupliquant le nombre de cellules E/R et en les plaçant de façon à ce qu'elles soient légèrement décalées. Cela permet de créer des *nappes* supplémentaires comme on peut le voir sur la figure 4.5. Le nombre ainsi que l'inclinaison relative des nappes sont fixés pour chaque capteur. L'augmentation du nombre de nappes permet d'avoir plus de points et permet aussi d'avoir des angles d'incidence différents pour les nappes. Les nappes avec différents angles permettent de régler des problèmes complémentaires. Par exemple, une nappe horizontale est très adaptée pour régler des problèmes de localisation mais elle ne détectera pas les obstacles plus petits que sa hauteur de fixation. À l'inverse, une nappe frappant le sol avec un certain angle sera très utile pour détecter les petits objets en utilisant le mouvement du porteur qui crée alors un balayage. Dans ce cas, le montage n'est pas très intéressant pour la localisation du porteur.

Données brutes d'un lidar Le lidar fournit des *scans* selon une cadence fixe. Les données sont transmises à un système informatique par connexion série ou Ethernet.

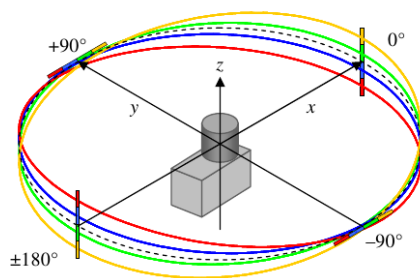
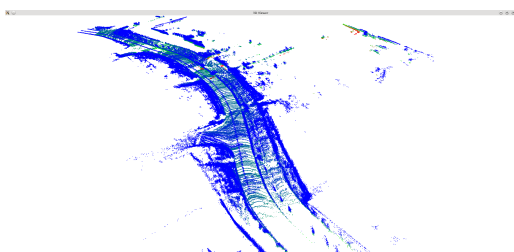
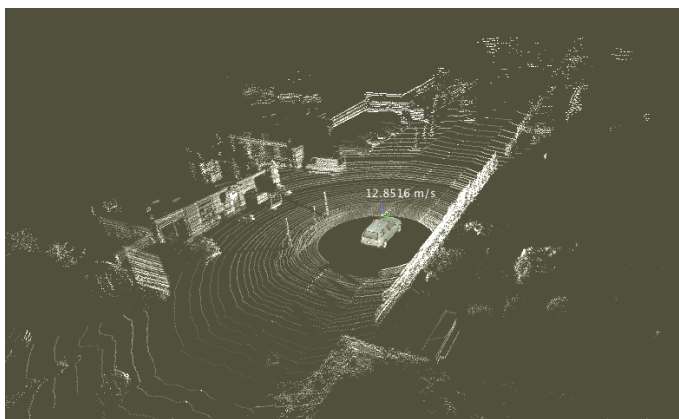


Figure 4.5.: Télémètre laser 4 nappes (source : IBEO Alasca user manual)

Un scan est composé d'une entête et d'un ensemble de données relatives à chaque écho laser du scan tel que les coordonnées de l'écho (en repère cartésien, cylindrique ou sphérique), le numéro de la nappe, un facteur de qualité (intensité du signal ou largeur d'impulsion) et le numéro de l'écho dans le cas de multi-échos.



(a) Illustration d'un ensemble de scan fournis par le lidar Alasca XT



(b) Illustration d'un scan fourni par le lidar Velodyne de l'université de Munich

Figure 4.6.: Exemple de données brutes issues de lidars.

La qualité d'un scan pour représenter l'environnement dépend principalement du nombre de point. Les lidars conçues pour des applications automobiles fournissent

des scans d'environ un millier de points. Ce type de scan seul est assez pauvre pour représenter correctement l'environnement pour des tâches telle que la navigation, mais la fusion d'un ensemble de scans fournit nettement plus d'information. La figure 4.6 montre la représentation de différents nuages de point laser.

4.2.3. Lidar utilisé

Dans ces travaux, nous avons principalement utilisé le lidar Alasca XT. C'est un télémètre laser à balayage développé par la société IBEO pour les applications automobiles embarquées. Ce capteur dispose de 4 nappes séparées par un angle de 0.8° . Le faisceau laser utilisé est de classe 1. Il a une ouverture de 0.25° (5 mrad) et opère en infrarouge dans une longueur d'onde de 905 nm. Le lidar mesure des distances dans la plage $[0.3 \ 200]$ m avec une précision en distance de l'ordre de 10 cm (résolution en distance de 4 cm). Enfin, le miroir rotatif tourne à une vitesse réglable entre 8 Hz et 40 Hz. En fonction de la fréquence de rotation, la résolution angulaire varie. La résolution varie aussi en fonction de la zone angulaire suivant 4 zones représentées sur la figure 4.7. Nous avons utilisé la configuration à 12.5 Hz.

Ce capteur est doté de capacités multi-échos, car il est capable de retourner jusqu'à 4 points par impulsion laser. Enfin, ce capteur est aussi doté d'un calculateur capable de traiter en temps réel les données pour les segmenter et réaliser un suivi de piste. Nous n'utiliserons par la suite que les points 3D et nous considérerons ce capteur comme le capteur de perception.

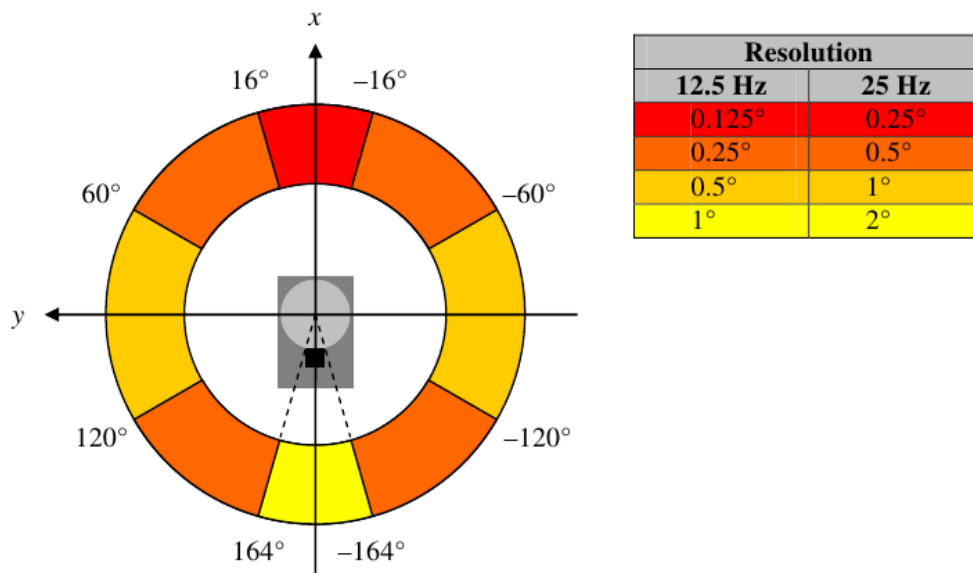


Figure 4.7.: Résolution angulaire de l'Alasca XT

4.3. Modèle de capteur lidar

4.3.1. Lien entre mesure et occupation

Modèle de capteur de mesure Le lidar est un capteur polaire télémétrique optique. Une mesure du capteur est un point 3D de l'espace. Ce point peut être paramétré par une droite qui passe par le foyer du lidar et par son abscisse sur cette droite.

Cependant, les mesures sont en général bruitées et incertaines (e.g. mesure fantôme [Jensfelt, 2001]). Pour modéliser ces erreurs, on crée un modèle de mesure et on se place souvent dans un cadre probabiliste. Le modèle de capteur direct est défini par la fonction de vraisemblance $p(r|z)$ où on utilise classiquement un modèle de bruit gaussien (figure 4.8). L'équation 4.4 montre l'expression d'un modèle de capteur 1D à bruit gaussien de variance σ_r^2 , où z représente la distance mesurée et r représente la distance réelle.

$$p(r|z) = \frac{1}{\sqrt{2\pi}\sigma_r} \cdot e\left(-\frac{1}{2} \frac{(r-z)^2}{\sigma_r^2}\right) \quad (4.4)$$

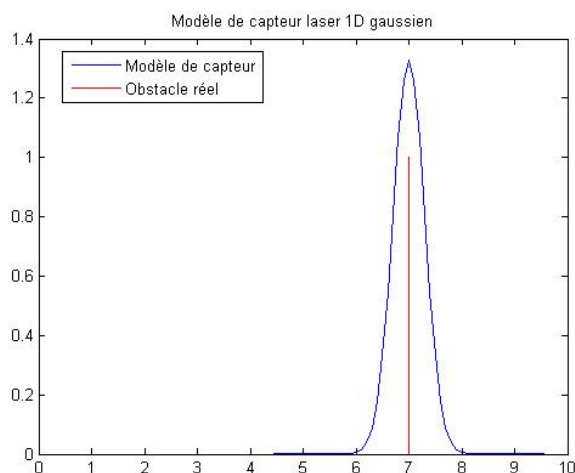


Figure 4.8.: Modèle de capteur 1D

Modèle de capteur sur grille Dans le chapitre 2, nous avons introduit la notion de modèle de capteur pour faire le lien entre l'état des cellules de la grille et les mesures du capteur. Ce type de modèle donne une interprétation des mesures qui traduit les incertitudes provenant des erreurs de mesure (modèle de mesure) et des hypothèses considérées.

A partir du modèle de capteur $p(r|z)$, on peut exprimer une relation avec les cellules de la grille sur l'axe traversées par la projection sur la grille du faisceau laser produisant la mesure r . Cette densité de probabilité peut être calculée par la relation

suivante qui traduit le fait que toutes les cellules k situées avant la cellule i sont libres :

$$p(r|z) = p(r|O_i \wedge F_k, k < i) \text{ si } z \in G_i^O \quad (4.5)$$

On peut exprimer le modèle de capteur sur grille $p(r|O)$ à partir de l'ensemble des configurations de grille g_{ij}^O qui conduisent à produire cette observation :

$$p(r|O) = \sum_{g \in \{g_{ij}^O\}} p(r|O, g) P(g|O) \quad (4.6)$$

Le terme g_{ij}^O représente l'ensemble des configurations de la grille qui vérifie la contrainte $O(G_{ij}^O) = 1$, qui dans notre cas doit vérifier la condition de l'équation 4.5. La figure 4.9 montre le résultat que produit cette approche avec un capteur à ultrason et un modèle gaussien 2D.

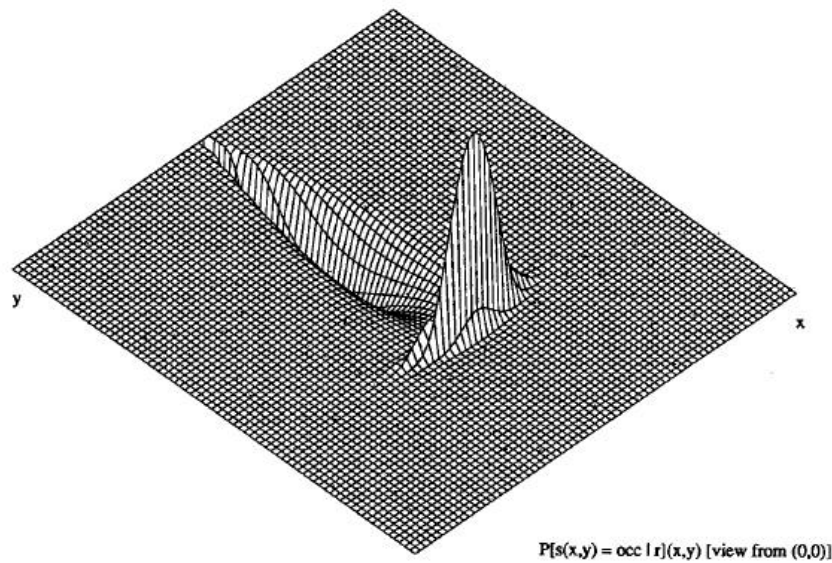


Figure 4.9.: Modèle de capteur ultra-sonore 2D (extrait de [Elfes, 1989c]).

4.3.2. Principes pour l'élaboration de modèles inverses de capteur

Pour les raisons que nous avons évoquées dans les chapitres précédents, nous utilisons un modèle de capteur sur grille inverse.

Hypothèses Ce modèle repose sur les hypothèses suivantes :

- l’obstacle est vertical et en contact plein avec le sol,
- l’obstacle produit une réflexion diffuse avec suffisamment d’énergie pour la détection (on ne considère pas les obstacles transparents, parfaitement spéculaire ou absorbants),
- le sol est localement plan.

A partir de ces informations, on peut construire un modèle précisant l’existence et la position d’un obstacle, en faisant les considérations suivantes :

- Si le faisceau laser est réfléchi, alors il y a un obstacle qui est responsable de cette réflexion et il est situé au point de l’écho.
- Si le faisceau laser ne s’est pas réfléchi avant le premier obstacle, c’est qu’il n’y a pas d’obstacle avant, donc l’espace est libre.
- Après le premier écho laser, le secteur angulaire est totalement ou partiellement masqué et donc l’espace est non observable. On ne peut donc pas savoir si cet espace est libre ou occupé.

Nous proposons d’utiliser un modèle de capteur sur grille polaire 2D pour rester proche des données brutes du capteur. Dans ce modèle, la grille est portée par un plan de référence parallèle au plan de la route et les points lasers sont projetés orthogonalement sur ce plan (hypothèse d’obstacle vertical). Chaque colonne de la grille polaire correspond à un secteur angulaire nommé Θ_i et chaque ligne correspond à un intervalle en distance $R_j = [r_j^-, r_j^+]$. Pour chaque mesure $Z_k = \{r_k, \theta_k\}$, on peut déduire que la cellule (Θ_i, R_j) telle que $\theta_k \in \Theta_i$ and $r_k \in R_j$ est occupée et que les cellules $(\Theta_i, R_{1:j-1})$ sont libres. Le modèle de faisceau laser est donc très simple à construire dans la grille polaire. Ce modèle prend en compte les incertitudes de mesure et les phénomènes de masquage via des mécanismes que nous allons détailler.

Modélisation des masquages Les lidars sont capables de renvoyer des points sur plusieurs obstacles situés à des profondeurs différentes dans un même secteur angulaire. Ceci est dû d’une part à la capacité multi-nappes et multi-échos du capteur, et d’autre part à la résolution angulaire si celle-ci est plus petite que la taille des secteurs angulaires de la grille. Pour prendre en compte les masquages, il faut détecter le point le plus proche au sein de chaque secteur angulaire.

La figure 4.10 montre le principe de la construction d’un secteur angulaire au travers d’un exemple. Le secteur angulaire contient trois impacts lasers : deux sont dans la cellule R_8 et un est dans la cellule R_{11} . Jusqu’au premier écho, les cellules sont libres (vert). Les cellules qui contiennent un ou plusieurs échos sont occupées (jaune). Enfin, les cellules non occupées situées derrière des cellules occupées sont classées inconnues (bleu) car masquées et donc potentiellement occupées.

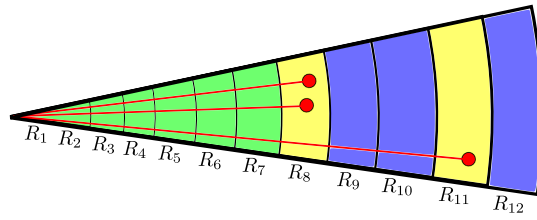


Figure 4.10.: Exemple de construction d'un secteur angulaire à partir de trois échos.

Transformation cartésienne Finalement, la grille polaire peut être transformée en utilisant les transformations de grilles présentées dans l'annexe ?? pour être exprimée dans un repère cartésien. La figure 4.11 montre le résultat sur la grille polaire contenant le secteur angulaire précédent projeté en repère cartésien.

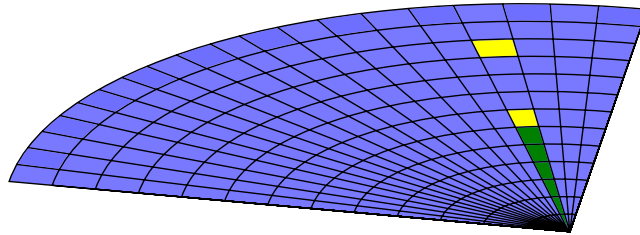


Figure 4.11.: Modèle de capteur 2D.

4.3.3. Détection du plan du sol

Le modèle de capteur présenté ci-dessus est un modèle d'obstacle et fait l'hypothèse que la réflexion du faisceau provient uniquement d'obstacles positifs par rapport au sol. Mais à bord d'un véhicule automobile, la réflexion peut aussi se produire sur le sol, pour les nappes inclinées vers la bas par exemple. Dans ce cas, l'hypothèse n'est plus vérifiée et on risque d'introduire un degré d'occupation dans des cellules qui doivent rester vides. Pour éviter ce phénomène, il faut détecter et filtrer les points qui sont sur le sol.

Nous avons utilisé un algorithme qui a été développé dans le cadre du projet LOVE [Fayad, 2009]. Cet algorithme permet de détecter le plan du sol et de calculer l'angle

de tangage du Lidar par rapport à ce plan (pour des petits angles). Cette méthode suppose que le sol est plan et que la majorité des points situés devant le véhicule de la nappe la plus basse touche le sol. Elle fait aussi l'hypothèse que la hauteur du lidar, la divergence des faisceaux laser ainsi que leurs orientations relatives sont connues.

Une fois les points du sol labellisés, on peut les utiliser dans un modèle de capteur de sol. Ce modèle est similaire au modèle d'obstacle, sauf qu'il ne considère pas la cellule contenant l'impact occupée mais libre. La figure 4.12 reprend l'exemple précédent dans le cas où les deux premiers échos se situent sur la route.

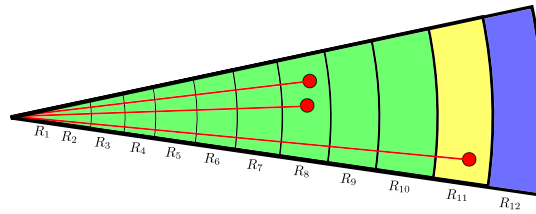


Figure 4.12.: Exemple de construction d'un secteur angulaire contenant trois échos dont deux tapant le sol (R_8).

4.3.4. Modèles inverses de capteur sur grille polaire

On cherche donc à construire une ScanGrid, c'est-à-dire une grille de perception instantanée projetée au niveau du sol à partir des données lidar multi-échos et multi-nappes. Nous proposons dans cette partie plusieurs modèles adaptés au mode de représentation des incertitudes choisi.

Modèle d'accumulation Le modèle de capteur d'accumulation fonctionne suivant le principe décrit dans le chapitre 3.

La première étape consiste à projeter au sol tous les points qui sont les obstacles. On suppose ici que les points situés sur la route ont été correctement éliminés par la procédure décrite dans la section 4.3.3. On segmente les points par secteur angulaire et pour chaque secteur angulaire on applique la procédure suivante.

Pour chaque cellule, le niveau d'accumulation² Al est initialisé à 0. On incrémente l' Al à chaque écho et on décrémente les cellules qui précèdent la première cellule occupée dans le secteur angulaire. Les autres cellules restent à 0. La figure 4.13 montre un exemple de construction pour un secteur angulaire contenant 3 échos.

² On rappelle que dans le cadre d'accumulation, chaque cellule contient un niveau d'accumulation Al .

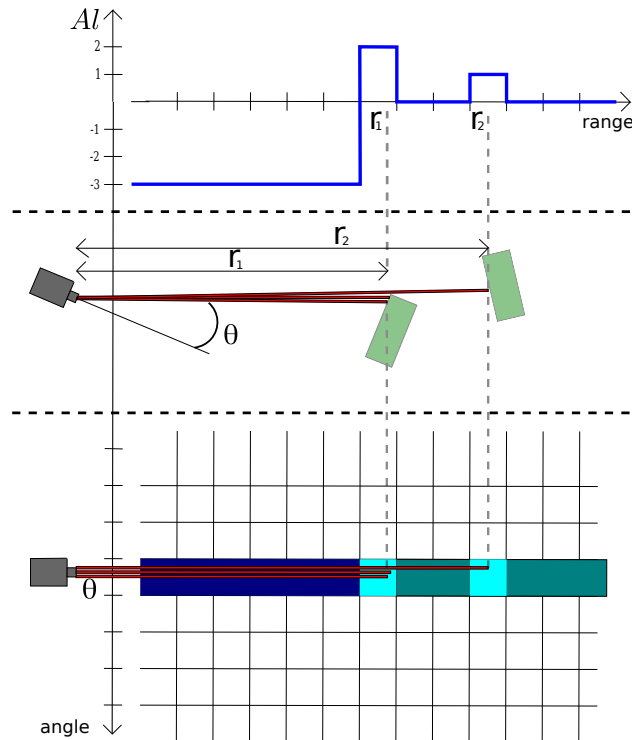


Figure 4.13.: Modèle de capteur par accumulation.

Modèle évidentiel par traitement procédural des secteurs angulaires Ce modèle suit un principe similaire au modèle d'accumulation, c'est-à-dire qu'on projette les échos et on les traite par secteur angulaire.

Pour indiquer les cellules G_{ij} , on considère un repère de grille $\mathcal{R}_O(O, \vec{\theta}, \vec{r})$ tel que r est vers le haut positif et θ est vers la droite. Ainsi, une cellule est paramétrée par le couple $\{\Theta_i, R_j\}$, où $\Theta_i = [\theta_i^- \theta_i^+]$ $R_j = [r_j^- r_j^+]$ et est indiquée par son centre (θ_i, r_j) .

On considère le cadre de discernement défini dans le chapitre 3, $\Omega = \{F, O\}$.

Pour chaque secteur angulaire Θ_i , on considère les n_i mesures (r_k, θ_k) telles que $\theta_k \in \Theta_i$ et pour chaque cellule $\{\Theta_i, R_j\}$, le contenu de la cellule $I_{i,j}$ est la fonction

de masse $[m_S(\emptyset) \ m_S(O) \ m_S(F) \ m_S(\Omega)]$ défini comme suit :

$$\begin{aligned}
 & \forall k \in [1 \ n_i] \mid \theta_k \in \Theta_i \\
 & m_{\theta_i, r_j}(\emptyset) = 0 \\
 & m_{\theta_i, r_j}(O) = \begin{cases} 1 - \lambda_{FA} & \text{si } \exists r_k \in R_j \\ 0 & \text{sinon} \end{cases} \\
 & m_{\theta_i, r_j}(F) = \begin{cases} 1 - \lambda_{MD} & \text{si } r_j^+ < \min(r_k) \\ 0 & \text{sinon} \end{cases} \\
 & m_{\theta_i, r_j}(\Omega) = \begin{cases} \lambda_{FA} & \text{si } \exists r_k \in R_j \\ \lambda_{MD} & \text{si } r_j^+ < \min(r_k) \\ 1 & \text{sinon} \end{cases}
 \end{aligned} \tag{4.7}$$

où λ_{FA} et λ_{MD} sont les taux de non détection et de fausse alarme du capteur. Ils sont supposés connus et sont des paramètres de la méthode. En pratique ces paramètres ont été estimés à partir des caractéristiques techniques et d'une étude statistique du capteur. La figure 4.14 montre un exemple de profil d'allocation suivant un secteur angulaire obtenu dans le cas où il y a 3 échos laser. Il est à noter que les masses sur les singletons n'atteignent pas la valeur maximale de 1, ce qui permet de prendre en compte les erreurs de modélisation (par exemple la présence de vitres, etc..).

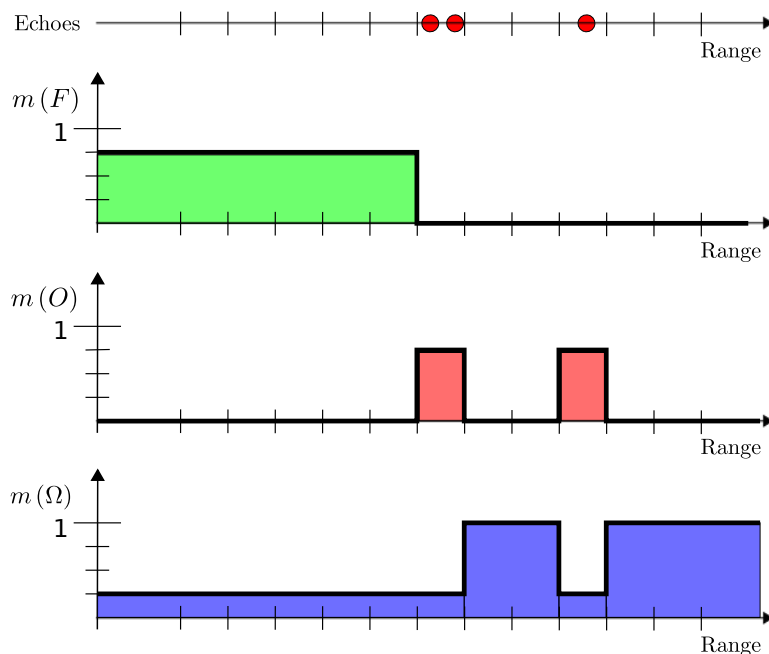


Figure 4.14.: Modèle de capteur évidentiel simplifié. (Rq. ici $\lambda_{FA} = \lambda_{MD} = 0.25$)

Modèle évidentiel par fusion de tous les échos On peut définir un modèle de capteur élémentaire qui considère individuellement chaque écho pour construire une bba puis fusionne les bba de tous les échos d'un secteur angulaire entre eux pour produire la *ScanGrid*.

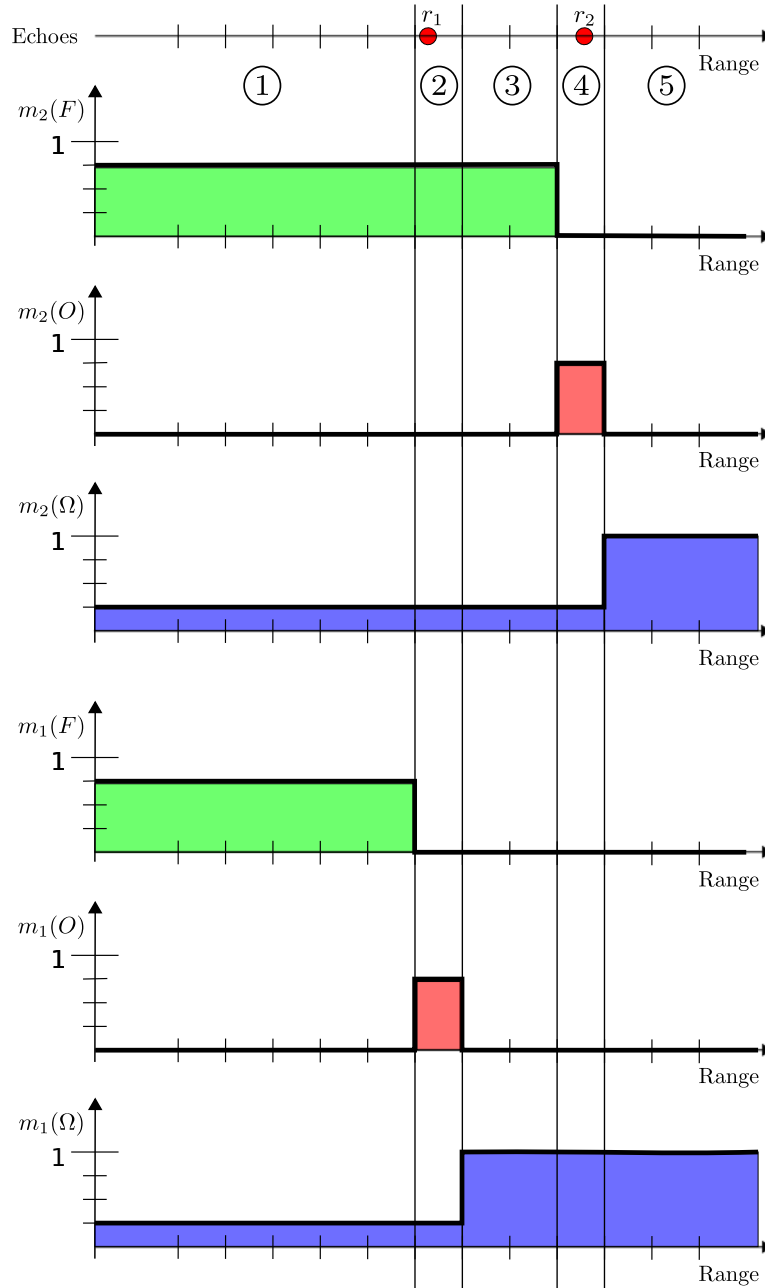


Figure 4.15.: Profils du modèle de capteur pour deux mesures élémentaires.

Cependant, une fusion classique des échos ne permet pas de prendre en compte les effets de masquage, effets pris en compte d'une façon procédurale précédemment. L'exemple ci-dessous permet de mettre en avant les problèmes d'une fusion conjonctive pour fusionner de façon indépendante les bba de deux échos. Supposons le cas 1D où l'on dispose de 2 échos positionnés respectivement à r_1 et r_2 . Les deux fonctions de masse élémentaires m_1 et m_2 sont représentées sur la figure 4.15.

Sur cet exemple, on peut décomposer la distance en 5 intervalles (de la gauche vers la droite). Lors de la fusion conjonctive (voir section 3.4.4), les termes non nuls sont :

- Dans la partie 1, $m_1(F)$ avec $m_2(F)$, $m_1(F)$ avec $m_2(\Omega)$, $m_1(\Omega)$ avec $m_2(F)$ et $m_1(\Omega)$ avec $m_2(\Omega)$.
- Dans la partie 2, $m_1(F)$ avec $m_2(O)$, $m_1(F)$ avec $m_2(\Omega)$, $m_1(\Omega)$ avec $m_2(O)$ et $m_1(\Omega)$ avec $m_2(\Omega)$.
- Dans la partie 3, $m_1(F)$ avec $m_2(\Omega)$ et $m_1(\Omega)$ avec $m_2(\Omega)$.
- Dans la partie 4, $m_1(O)$ avec $m_2(\Omega)$ et $m_1(\Omega)$ avec $m_2(\Omega)$.
- Dans la partie 5, $m_1(\Omega)$ avec $m_2(\Omega)$.

Le résultat de la fusion conjonctive est illustrée sur la figure 4.16. Le résultat de la fusion ne correspond pas ce que l'on voudrait. Cela vient du fait que les masquages ne sont pas modélisés avec ce type de fusion. De ce fait, dans la partie 2, on obtient une masse non nulle sur libre due au terme $m_1(F) \cdot m_2(\Omega)$ alors que la cellule est occupée. Dans la partie 3, on obtient une importante masse sur libre à cause du terme $m_1(F) \cdot m_2(\Omega)$ alors que le masquage devrait nous amener à considérer cette cellule inconnue.

Pour améliorer la méthode, on peut mettre en place la stratégie suivante.

Dans un premier temps, on détecte l'espace libre en accord avec tous les points. Pour cela, on utilise la fusion disjonctive puisqu'elle permet de conserver l'information commune (ici, pour détecter l'espace libre, il y a une source fiable : l'écho le plus proche). Pour cela, on calcule donc une distribution de fonctions de masse m_f (le long du secteur angulaire) en réalisant la fusion disjonctive par cellule des k bba calculées à partir des échos :

$$m_f = \bigoplus_k m_k.$$

La fonction de masse m_f sera soit une fonction de masse vide pour les cellules situées derrière le premier impact (c.f. secteur 2, 3, 4, et 5 sur la figure 4.17), soit une fonction de masse avec un seul élément focal F pour les autres. Comme $m_f(F)$ est le résultat d'un produit de nombres plus petits que 1, on normalise cette masse à 1.

$$m_f(F) = \frac{m_f(F)}{\max(m_f(F))} \text{ si } \max(m_f) \neq 0$$

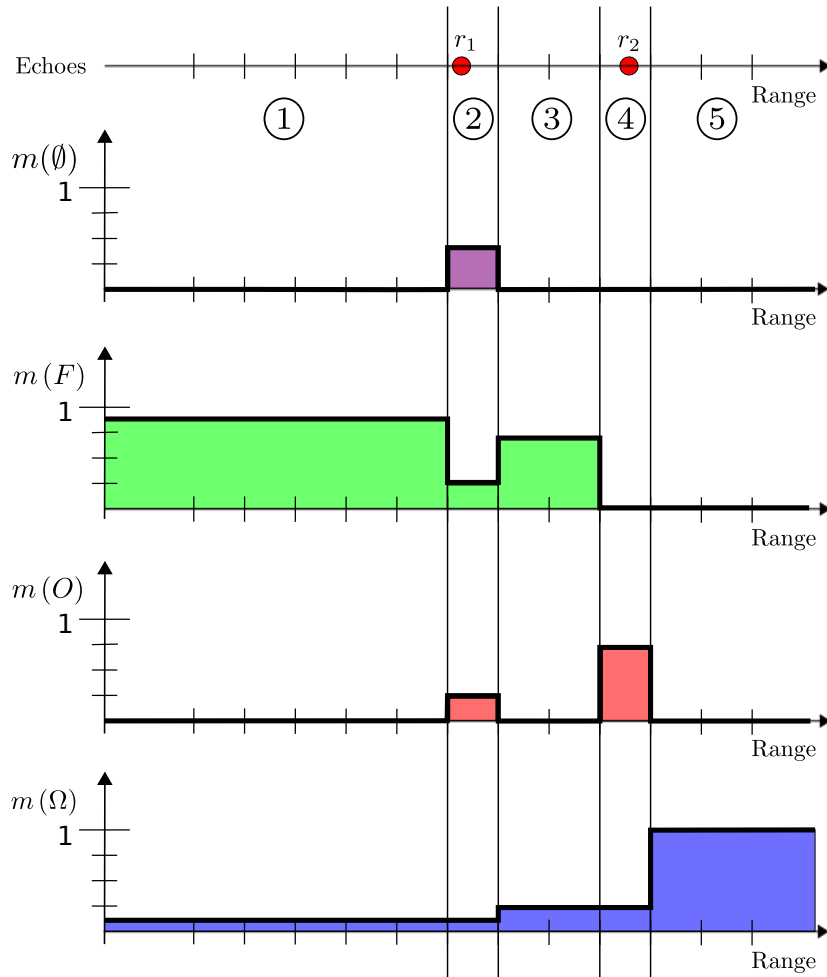


Figure 4.16.: Résultat de la fusion des deux bba avec une fusion conjonctive.

$$m_f(\Omega) = 1 - m_f(F)$$

La figure 4.17 montre le résultat après la fusion et la normalisation.

Cette fonction de masse m_f permet de définir un masque sous la forme d'une fonction de masse m_m .

m_m est définie de la manière suivante :

$$m_m = \begin{bmatrix} \emptyset & F & O & \Omega \\ m_f(F) & 0 & m_f(\Omega) & 0 \end{bmatrix}.$$

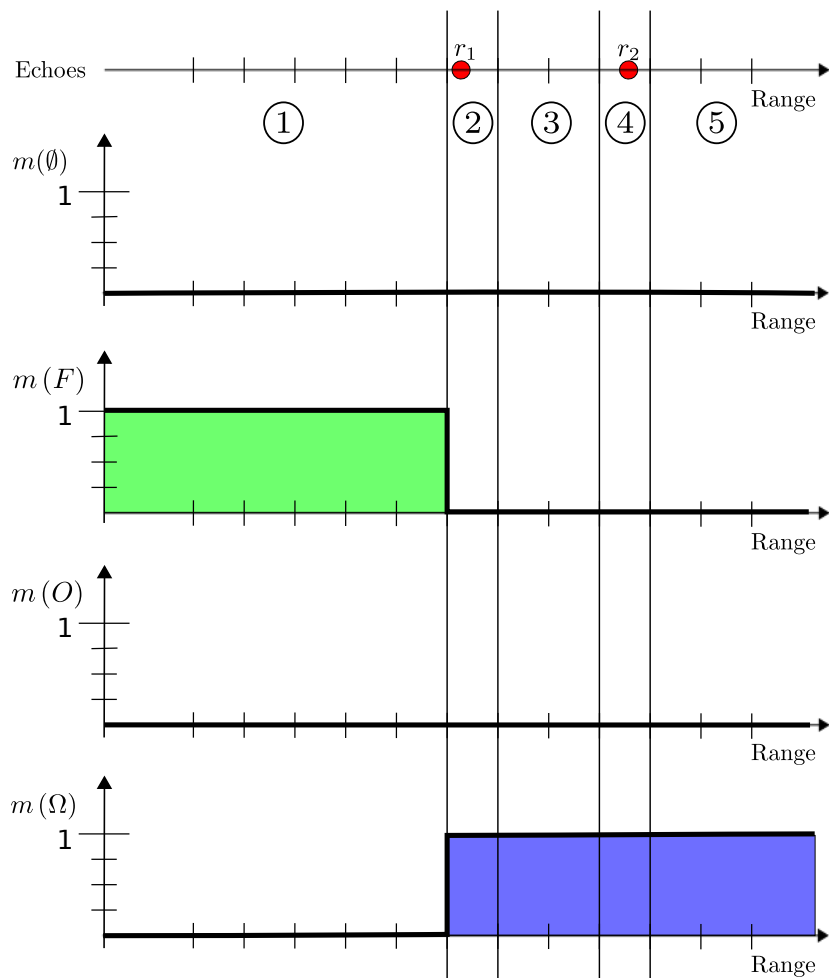


Figure 4.17.: Résultat de la construction de m_f à partir des deux bba.

La combinaison disjunctive entre m_m et chaque bba correspondant à un point d'impact m_k crée une nouvelle distribution m_{mk} adaptée au modèle :

$$m_{mk} = m_k \odot m_m$$

L'application de ce masque correspond à un affaiblissement contextuel tel qu'il est défini dans [Mercier *et al.*, 2005] et est montré dans l'annexe ??.

La figure 4.18 montre les bba prenant en compte les masquages. On peut remarquer que la première distribution de fonction de masse a maintenant une masse vide sur les cellules des intervalles 2 et 3.

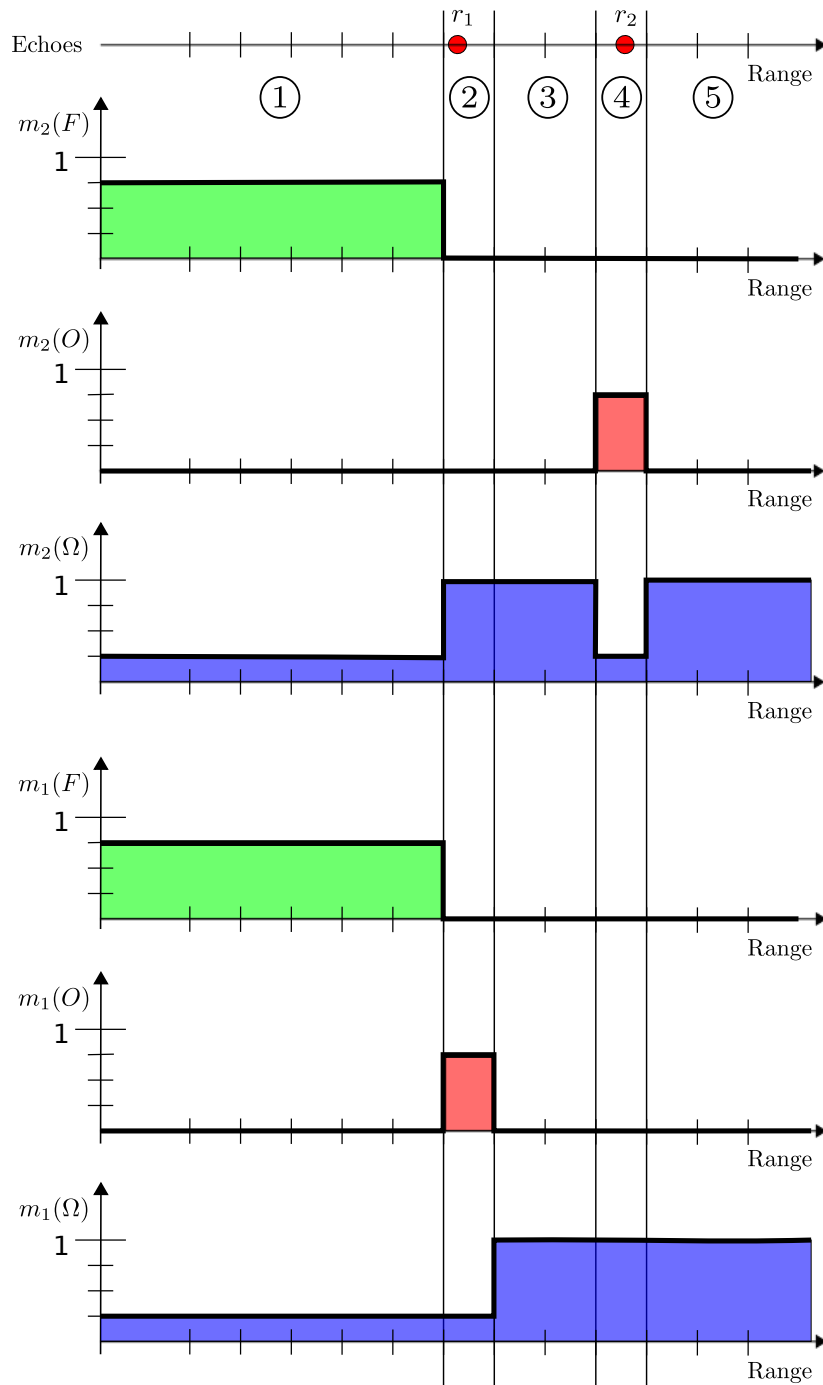


Figure 4.18.: Profil des bba élémentaires prenant en compte les effets de masquages après affaiblissement. On peut remarquer que dans les zones 2 et 3, les fonctions de masse $m_2(F)$ sont nulles maintenant.

Finalement, on peut réaliser la fusion de chaque fonction de masse avec Dempster, par exemple :

$$m^{SG} = \oplus_k m_{mk}$$

Cela permet d'obtenir le résultat de la figure 4.19.

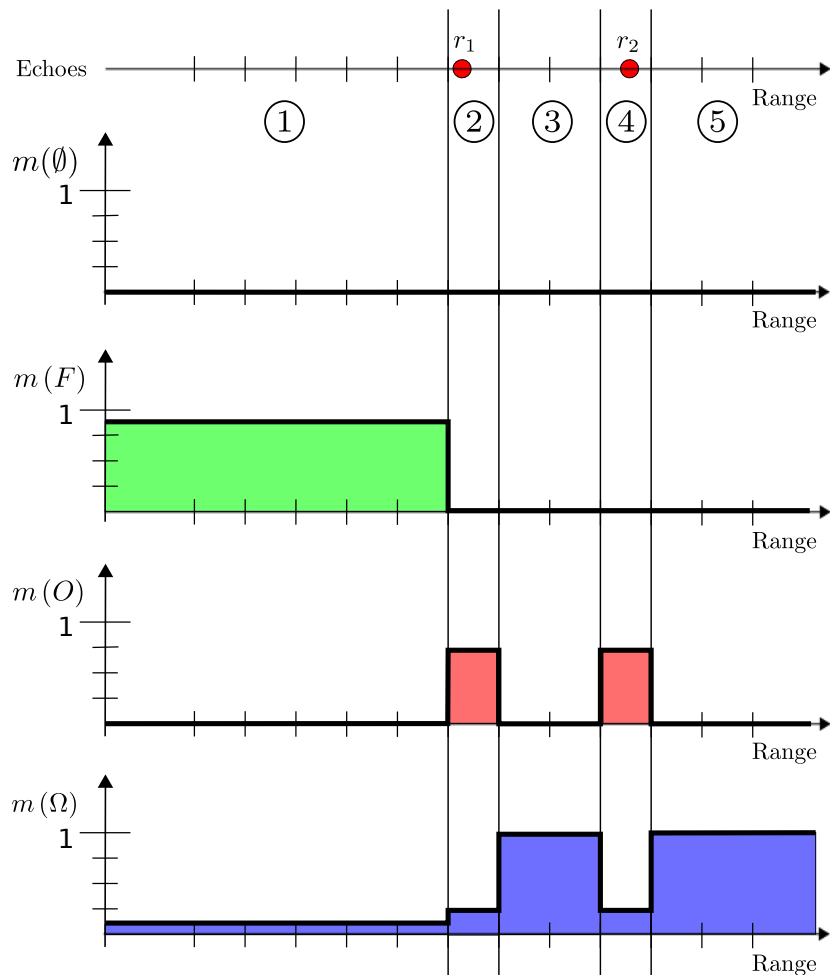


Figure 4.19.: Résultat de la fusion des deux bba prenant en compte les effets de masquage.

On remarque que les problèmes de fusion précédemment soulignés sont corrigés par cette méthode. Il n'y a plus de conflit et de masse sur libre dans la cellule de l'intervalle 2 et les cellules de l'intervalle 3 sont inconnues.

De plus, on peut remarquer que la masse sur libre dans l'intervalle 1 se renforce avec chaque nouvel écho et il en est de même pour la masse sur occupé pour les cellules occupées par plusieurs points. Ceci constitue le principal avantage de ce modèle par rapport au modèle procédural précédent.

Modèle évidentiel 2D gaussien Les modèles considérés jusqu'à présent étaient des modèles 1D par secteur angulaire. Pour ne plus faire d'approximation dans le modèle de capteur liée à la géométrie de la grille, on peut définir un modèle de capteur à partir de la distribution spatiale d'incertitude des erreurs de mesure (comme on peut le trouver dans certaines approches probabilistes). Pour s'affranchir de la contrainte de grille, on utilise des cellules de très faible dimension pour échantillonner les distributions. En pratique, ce modèle peut être utile si la taille des cellules est petite par rapport à l'empreinte du faisceau laser et par rapport à l'imprécision de mesure longitudinale.

Utilisons un modèle d'incertitude gaussien pour r et θ et considérons les écarts-types σ_r et σ_θ connus. Supposons aussi pour simplifier que les erreurs de mesure sont indépendantes entre elles.

A partir d'un écho laser de coordonnées r_k et θ_k , on peut construire la distribution de fonction de masse 2D pour chaque cellule (Θ_i, R_j) (de centre (θ_i, r_j)) de la façon suivante :

$$\begin{aligned}
m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (O) &= (1 - \lambda_{fa}) \cdot \frac{1}{\sigma_r \sqrt{2\pi}} \cdot \exp \left(-\frac{1}{2} \left(\frac{r_k - r_j}{\sigma_r} \right)^2 \right) \cdot \frac{1}{\sigma_\theta \sqrt{2\pi}} \cdot \exp \left(-\frac{1}{2} \left(\frac{\theta_k - \theta_i}{\sigma_\theta} \right)^2 \right) \\
m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (F) &= \begin{cases} (1 - \lambda_{md}) \cdot \left(1 - \frac{1}{\sigma_r \sqrt{2\pi}} \cdot \exp \left(-\frac{1}{2} \left(\frac{r_k - r_j}{\sigma_r} \right)^2 \right) \right) & \text{si } r_j < r_k \\ \cdot \frac{1}{\sigma_\theta \sqrt{2\pi}} \cdot \exp \left(-\frac{1}{2} \left(\frac{\theta_k - \theta_i}{\sigma_\theta} \right)^2 \right) & \\ 0 & \text{sinon} \end{cases} \\
m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\Omega) &= 1 - m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (O) - m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (F) \\
m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\emptyset) &= 0
\end{aligned} \tag{4.8}$$

On peut se représenter la distribution de fonction de masse sur F comme étant un "tunnel" gaussien qui va du capteur jusqu'à l'écho et qui décroît exponentiellement à l'approche de l'écho.

Comme ici on considère des distributions 2D, on ne peut pas définir le concept de "cellules précédant ou suivant un écho", car un écho occupe plusieurs cellules (idéalement, le support d'une gaussienne s'étend jusqu'à l'infini). Pour créer le modèle de capteur, on va donc combiner les bba de chaque écho en utilisant les mêmes principes que la méthode par fusion de mesures élémentaires décrite dans le paragraphe précédent. La première différence provient du fait qu'il faut faire une fusion 2D sur l'ensemble de la grille (et non par secteur angulaire). La seconde différence est due à l'existence d'un nouveau type d'ignorance qui intervient dans les cellules hors de l'ouverture du laser, à la différence de l'ignorance de masquage, qui intervient dans les cellules non observées car masquées.

Ici, il faut donc faire la distinction entre l'espace inconnu masqué et l'espace inconnu non observé, car pour modéliser l'effet de masquage, l'affaiblissement ne doit être appliqué que sur les parties masquées et non sur les parties non observées. Cela intervient au niveau de la fusion disjonctive qui sert à calculer m_f .

Pour détecter l'espace libre, on crée la distribution de fonction de masse \widetilde{m} de la façon suivante. A partir de la distribution du k -ième écho m_k , on sépare l'ignorance de masquage qui est affectée sur Ω (comme nous nous avons fait précédemment) de l'ignorance de non observation qui est affectée sur \emptyset car il s'agit de élément neutre dans la fusion disjonctive.

Après calculs on montre que cela s'écrit comme suit :

$$\begin{aligned}
 \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (O) &= m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (O) \\
 \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (F) &= m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (F) \\
 \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\Omega) &= \frac{1}{\sigma_\theta \sqrt{2\pi}} \cdot \exp \left(-\frac{1}{2} \left(\frac{\theta_k - \theta_i}{\sigma_\theta} \right)^2 \right) - \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (O) - \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (F) \\
 \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\emptyset) &= 1 - \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\Omega) - \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (O) - \widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (F)
 \end{aligned} \tag{4.9}$$

La troisième équation $\widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\Omega)$ adapte la masse $m_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\Omega)$ mais pondère celle-ci en fonction de l'écart angulaire θ (On peut remarquer que pour $\theta = 0$ on retrouve la même expression). L'information est reportée sur l'élément neutre $\widetilde{m}_{k,\theta_i,r_j} \{ \theta_k, r_k \} (\emptyset)$.

La détection de l'espace libre se fait, comme dans l'approche précédente, par fusion disjonctive entre tous les échos :

$$m_f = \bigoplus_k \widetilde{m}_k$$

On normalise m_f de la même façon :

$$m_f (F) = \frac{m_f (F)}{\max (m_f (F))}$$

$$m_f (\Omega) = 1 - m_f (F)$$

A partir de m_f , on crée le masque m_m pour réaliser l'affaiblissement :

$$m_m = \begin{bmatrix} \emptyset & F & O & \Omega \\ m_f(F) & 0 & m_f(\Omega) & 0 \end{bmatrix}$$

Puis, on applique l'affaiblissement sur les distributions de masse 4.8 de chaque écho :

$$m_{mk} = m_m \odot m_k$$

Enfin, on réalise la fusion de Dempster sur tous les échos :

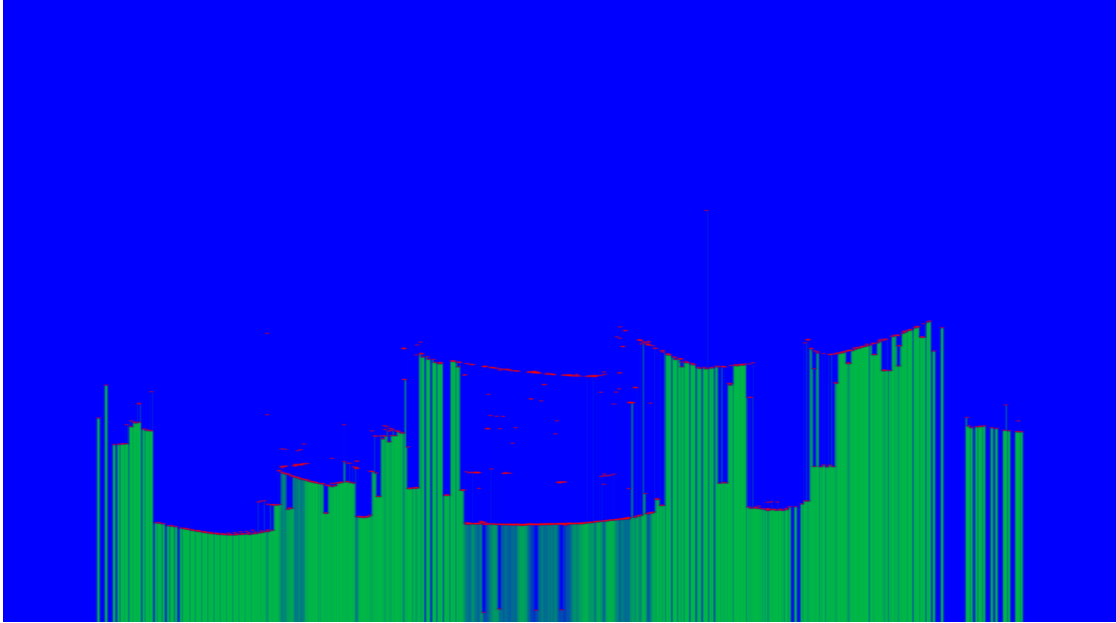
$$m^{SG} = \oplus_k m_{mk}$$

Les figures 4.20 montrent la grille polaire résultant de la fusion de tous les échos d'un scan lidar réel, soit environ 800 points. Ici, les écart-types des gaussiennes sont $\sigma_r = 0.1\text{m}$ et $\sigma_\theta = 0.2^\circ$ (c'est-à-dire typiquement l'ordre de grandeur des incertitudes des lidars fréquemment utilisés en robotique) et la résolution de la grille est de 0.05° par 0.01 m .

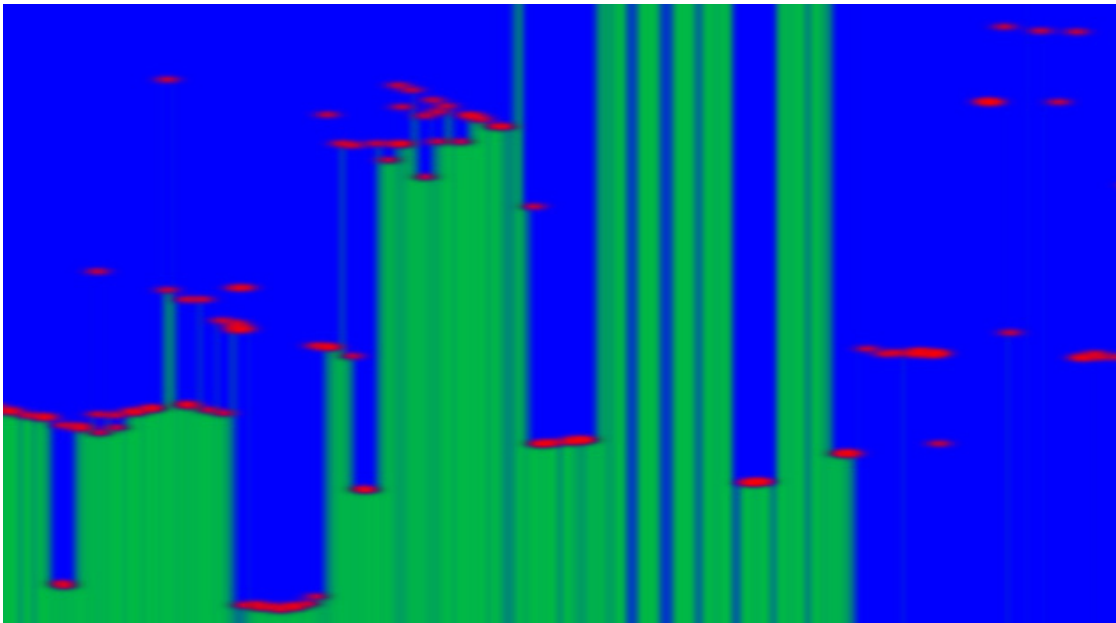
Sur ces figures, les couleurs de chaque pixel représentent la valeur de la fonction de masse associée à cette cellule. La figure 4.20a montre l'ensemble de la grille. Chaque ordonnée représente une distance r différente allant de $r = 0\text{ m}$ pour la ligne du bas jusqu'à $r = 30\text{ m}$. Chaque abscisse représente un secteur angulaire. L'axe central correspond à la direction $\theta = 0^\circ$. Même si en repère polaire, il est assez difficile d'interpréter la scène, nous avons choisi ici de monter les grilles polaires car ces dernières sont très précises et nous ne voulons pas ajouter d'erreurs d'interpolation.

On peut remarquer d'abord que cette méthode de fusion permet bien de créer une *ScanGrid*. Elle permet de détecter les cellules occupées (en rouge) et l'espace libre avant ces cellules (en vert). Les masquages sont bien gérés. En effet, on peut remarquer que les cellules rouge qui sont masquées sont précédées de bleu et non de vert ce qui signifie inconnu.

Sur la figure 4.20b, on peut remarquer que chaque impact laser est représenté par une petite ellipse (rouge) qui représente la distribution gaussienne suivant r et θ . On peut aussi remarquer que la masse attribuée sur l'espace libre suit aussi une distribution gaussienne suivant θ . Enfin, on peut remarquer que lorsque plusieurs échos sont assez proches, leur contributions s'ajoutent. On peut notamment voir cet effet sur les ensembles de points alignés dont les distributions se chevauchent sur la figure 4.20b.



(a) ScanGrid polaire obtenue par la fusion élémentaire des fonctions de masses 2D de 791 points laser. Chaque pixel correspond à une cellule élémentaire de la grille qui va de -70° à 70° (abscisse) et de 0 à 30m (ordonnée, la ligne 0 est en bas). L'intensité des couleurs correspond à la valeur des masses : vert pour libre, rouge pour occupé et bleu pour inconnu.



(b) Zoom sur la partie centrale (légèrement au gauche) de la figure

Figure 4.20.: Modèle de capteur évidentiel gaussien obtenu par fusion de chaque écho.

Ce modèle de capteur est donc très précis et il est capable de traiter l'ensemble

des cellules concernées par chaque écho, aussi bien de façon transversale que de façon longitudinale en polaire. Cette méthode est néanmoins très calculatoire, et nécessite de recalculer l'ensemble de la grille pour chaque écho (complexité en $O(N_{points} \cdot N_{cellules})$). Néanmoins, on peut réduire la complexité en limitant le support des gaussiennes et en réalisant une fusion locale. C'est le modèle simplifié présenté auparavant qui a été utilisé en temps réel dans la suite de cette thèse.

4.4. Conclusion

Dans ce chapitre, nous avons rappelé le principe de fonctionnement du lidar qui est le principal capteur utilisé dans nos expérimentations. Nous avons proposé plusieurs modèles de capteur inverse permettant de prendre en compte facilement les masquages et les incertitudes et intégrant les capacités multi-échos, multi-nappes et multi-résolutions du capteur. Ces modèles permettent de transformer un ensemble d'impacts lidar en une grille de perception instantanée appelée *ScanGrid*. Une première méthode, procédurale, offre un compromis entre la complexité calculatoire et une précision suffisante au regard de l'application. Cette approche a été utilisée avec les formalismes d'accumulation et évidentiel et a été mise en œuvre dans un démonstrateur temps réel. Un autre modèle considérant que les erreurs des points lidar suivent des distributions d'incertitudes continues (1D ou 2D), est défini sur la base de combinaisons disjonctives et conjonctives de fonctions de masse élémentaires.

5. Fusion séquentielle multi-grilles et mise en œuvre expérimentale

Sommaire

5.1. Introduction	113
5.2. Fusion séquentielle halo-centrée	114
5.3. Fusion séquentielle égo-centrée	136
5.4. Conclusion	149

5.1. Introduction

Nous nous intéressons dans ce chapitre à l'étude et à la mise en œuvre des méthodes de perception utilisant plusieurs grilles pour percevoir l'environnement. La combinaison des grilles est réalisée dans un schéma global selon les deux approches présentées dans le chapitre 2.

Nous considérons ici un système perception complet qui repose principalement sur la technologie lidar combinée à un système de positionnement (GPS aidé d'une centrale inertielle ou données proprioceptives provenant du bus CAN) et à des données cartographiques géolocalisées.

Ce chapitre s'articule de la façon suivante. Dans la première partie, nous décrirons la mise en œuvre du système de fusion de grilles halo-centrée en utilisant les formalismes d'accumulation et évidentiel. Les expérimentations réalisées dans le cadre du projet CityVIP ont permis de tester et de comparer ces deux méthodes. Enfin dans la seconde partie, nous présenterons une version temps réel de l'approche égo-centrée qui a été implémentée sur le véhicule démonstrateur du laboratoire Carmen. Dans les deux cas, l'exploitation de l'information provenant des cartes numériques a été étudiée et intégrée dans le schéma de perception.

5.2. Fusion séquentielle halo-centrée

5.2.1. Rappel du principe

Comme détaillé dans la partie 2.4.4.1, la méthode halo-centrée utilise deux grilles appelées MG et SG . On définit un repère de navigation East, Nord, Up (ENU) selon le standard du projet CityVIP dont l'origine est située sur une station de base de l'IGN (Institut Géographique National) à Saint Mandé. La grille MG est fixe dans ce repère; elle est donc référencée dans l'espace de manière absolue. La grille SG est construite dans le repère du capteur en utilisant le modèle de capteur.

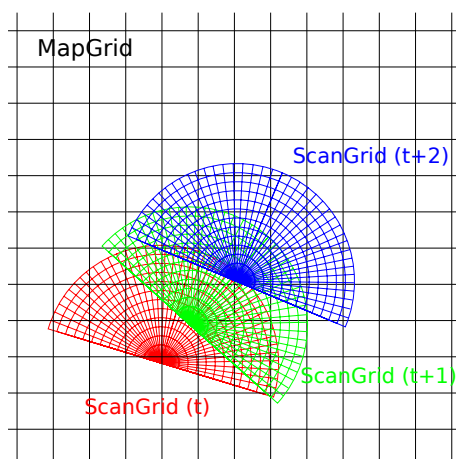


Figure 5.1.: Principe de l'approche de la fusion halo-centrée : la MG est fixe et chaque nouvelle SG est repositionnée dans le repère de la MG puis fusionnée.

A chaque instant où l'on reçoit un scan laser, on considère une SG polaire construite à partir des deux premiers modèles de capteur décrits dans le chapitre 4.3. Puis, cette SG est transformée dans un repère cartésien et ensuite repositionnée dans le repère de référence de la MG en utilisant la pose provenant d'un système de positionnement (par exemple une centrale inertielle) ainsi que la calibration extrinsèque du lidar par rapport au repère du véhicule. Pour cela, on utilise la transformation polaire cartésienne et les déplacements de grille définis dans le chapitre 2 utilisant des interpolations bi-linéaires. On fusionne alors la SG dans la MG en utilisant, soit la fusion d'accumulation, soit la fusion de Dempster dans le cas évidentiel. Enfin, à partir de la MG mise à jour, on cherche les cellules qui peuvent contenir des objets mobiles en utilisant soit le conflit soit, dans le cas de l'accumulation, en surveillant le niveau de l' Al des cellules détectées.

Le véhicule évoluant dans la MG (c.f. figure 5.1), cette dernière doit donc être suffisamment grande pour que le véhicule reste à l'intérieur durant toute sa mission. Si le véhicule s'approche d'une limite, il suffit de décaler la MG d'une distance multiple du pas échantillonnage spatial pour éviter de devoir interpoler.

Avant de décrire la mise en œuvre de la fusion halo-centrée, nous présentons dans la suite de ce paragraphe, le système expérimental utilisé.

5.2.2. Système expérimental

Dans le cadre du projet CityVIP, une vaste campagne d'acquisition a été réalisée dans Paris. Le véhicule équipé d'un grand nombre de capteurs, cf. figure 5.2, a enregistré plusieurs jeux de données. Le jeu de données utilisé a été enregistré dans le 12ème arrondissement de Paris. La trajectoire, qui est représentée sur la figure 5.3, forme une boucle d'environ 2 km. La boucle a été réalisée 3 fois.



Figure 5.2.: Véhicule utilisé pour l'enregistrement du jeu de données. Le capteur Alasca est installé sur le pare-choc avant et la caméra montée en dessous est protégée par un cache.

Les enregistrements disponibles que nous avons pu utiliser sont les données du télémètre Ibéo Alasca XT et celles d'un système de positionnement Applanix comprenant un récepteur GPS couplé à une centrale inertielle. Une caméra située à l'avant du véhicule au niveau du lidar permet de visualiser la scène.

Le lidar a été orienté dans cet essai de façon à avoir sa nappe du bas parallèle au sol (contrainte du projet CityVIP).

Le véhicule était équipé de 3 PC distincts qui permettaient l'enregistrement synchronisé des données de l'ensemble des capteurs.

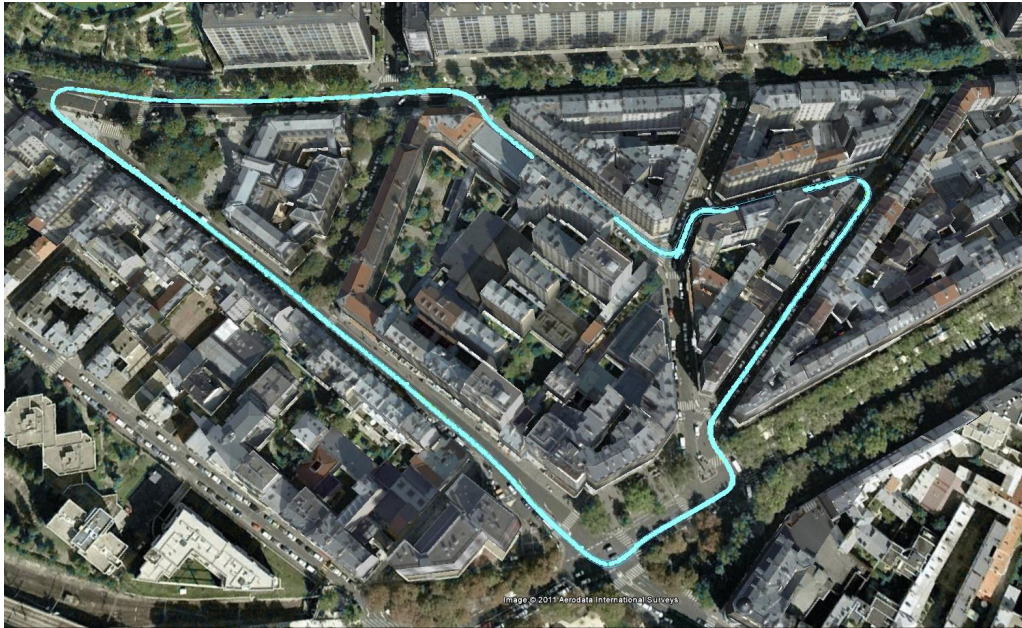


Figure 5.3.: Trajectoire du véhicule lors de l’enregistrement du jeu de données.

Données lidar Lors des acquisitions, le PC (PC1) qui a enregistré le lidar, a aussi servi à acquérir les données d’un récepteur GPS (PolarX Septentrio). Les données enregistrées sur ce PC sont datées suivant le format du temps UNIX (en microseconde depuis le 1er janvier 1970). Les données sont sauvegardées dans deux fichiers sous le format binaire “*dbt*” : un fichier d’index et un fichier de données. Pour faciliter l’importation dans Matlab et les échanges avec les partenaires du projet, ces données ont été converties en format texte via un programme de conversion développé pour l’occasion.

Système de positionnement Nous avons utilisé le système de positionnement Applanix Pos LV 220 embarqué sur le véhicule. Ce système est composé d’une centrale inertielle (IMU) et d’un récepteur GPS. Ces deux composants sont en couplage serré c’est-à-dire que les mesures brutes du GPS sont utilisées pour recalibrer la centrale. La centrale est aussi équipée d’une entrée odométrique qui était reliée à un odomètre sur les roues. La sortie de ce système est une pose 3 axes complète à laquelle sont associés des paramètres d’incertitude (covariance) à une fréquence de 100Hz. Les données ont été enregistrées sur un autre PC (PC2) en format texte et ont été datées en utilisant le temps GPS. Nous n’avons pas utilisé le fichier de pose à 100Hz, mais nous avons utilisé un fichier simplifié qui ne contient que les mesures des variations significatives.

D’autre part, l’Applanix est capable d’opérer à la volée des corrections RTK ou DGPS pour améliorer la précision et on peut réaliser un post-traitement des données hors ligne pour obtenir une qualité de positionnement maximale obtenue par lissage. De plus, dans le cadre du projet CityVIP, l’IGN a contribué à améliorer la précision

obtenue après post-traitement via des techniques de photogrammétrie, ce qui permet d’avoir une erreur de précision inférieure à 50 cm sur toute la trajectoire. Cette qualité optimale de positionnement n’a cependant pas été utilisée dans ces travaux.

Caméra Une caméra Marlin 146-C était située juste au dessous du lidar. Elle était équipée d’un objectif Pentax (6.5 mm, F 1.8). Cette caméra a enregistré au format “raw bayers 8 bits” avec une résolution de 1392x1038. La caméra avait une acquisition déclenchée par le lidar Alasca (à une fréquence de 15 Hz) mais ces données ont été enregistrées sur le PC2. Les données étaient composées d’un ensemble de fichiers images et d’un fichier d’index au format texte qui permettait de faire le lien entre les dates et les noms des fichiers images. La datation utilisée ici a été le format *Filetime* Windows (en centaine de nanosecondes depuis le 1er janvier 1601). Les images ont ensuite été traitées et leur résolution a été réduite.

Synchronisation des données Les différentes données ont été datées avec des systèmes de datation différents et ont été enregistrées sur des PC différents. La figure 5.4 résume les différents temps. Comme le PC1 n’a pas pu être synchronisé avec un signal PPS, nous avons dû estimer le décalage entre les horloges des PC. Pour cela, nous avons utilisé le récepteur GPS PolarX pour calculer le décalage de temps entre le temps GPS et le temps système enregistré sur le PC1. En effet, ce GPS a été enregistré sur le PC1 et donc ses données sont datées dans le temps de ce PC. La resynchronisation est possible grâce aux trames NMEA qui sont enregistrées car elles contiennent le temps commun du GPS. La date des images de la camera et des données Applanix sont ensuite converties en temps Unix puis dans le temps du PC1. Il est à noter que cette méthode ne prend pas en compte la latence due au calcul et à l’envoi de la trame GPS du récepteur PolarX (que l’on estime très faible car inférieure à 20 ms compte tenu que le capteur utilise une liaison Ethernet locale très rapide).

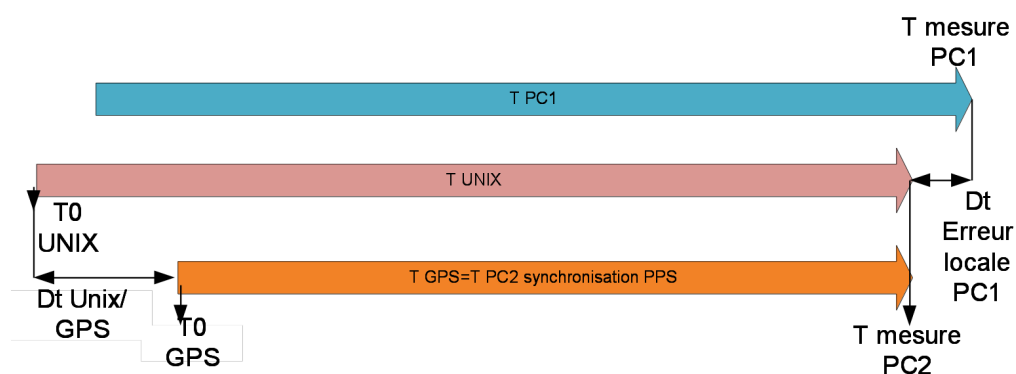


Figure 5.4.: Synchronisation des données.

5.2.3. Injection d'information cartographique

Nous avons cherché à étudier l'apport d'informations cartographiques dans un système de perception embarqué. En effet, les bases de données géographiques sont de plus en plus complètes et précises (figure 5.5). D'autre part, elles sont mises à disposition à tous les utilisateurs de manière gratuite ou peu coûteuse. On peut citer à titre d'exemple les cartes OpenStreetMap [OSM, 2004] disponibles librement sur internet et mises à jour par la communauté d'utilisateurs. Afin d'injecter l'information cartographique dans notre système de perception, nous proposons d'utiliser le formalisme des grilles pour intégrer et combiner l'information des cartes. Deux approches ont été étudiées. L'une utilise la grille de la carte comme une contrainte sur l'espace navigable. L'autre méthode propose de fusionner la grille de la carte avec la grille de perception. Cette dernière méthode a été étudiée dans le cadre des travaux de master de Marek Kurdej [Kurdej, 2011].

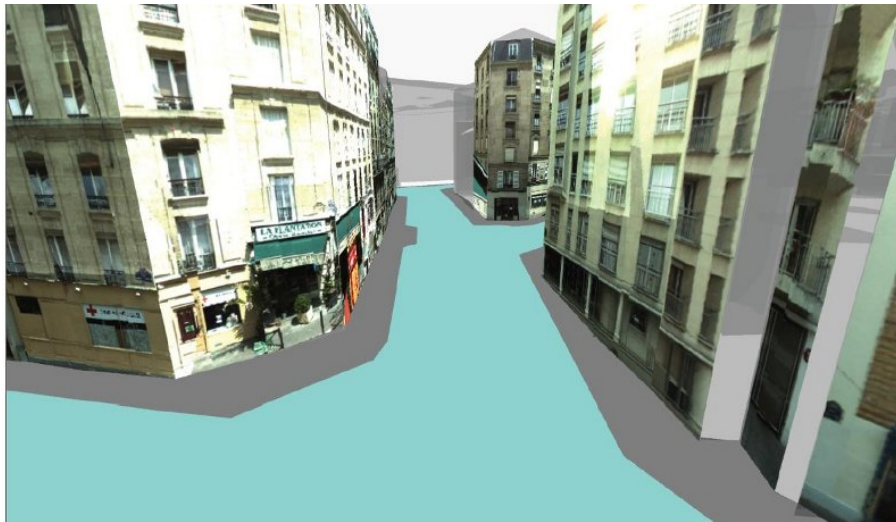


Figure 5.5.: Représentation dans un moteur en rendu temps réel de la carte texturée et de la chaussée.

Carte utilisée Dans le cadre du projet CityVIP, nous avons pu exploiter une carte vectorielle du trajet effectué dans le 12^{ème} arrondissement de Paris. Cette carte contient les informations sur la chaussée ainsi que sur le bâti.

La chaussée est constituée d'un maillage 3D de facettes triangulaires (figure 5.6) qui représentent la surface de la route. Ce maillage est décrit dans un fichier texte qui contient les coordonnées des sommets dans le référentiel Lambert 93 ainsi que les relations de connexion. Ce maillage a été créé manuellement par des opérateurs experts de l'IGN à partir des techniques de photogrammétrie sur des images aériennes haute résolution.

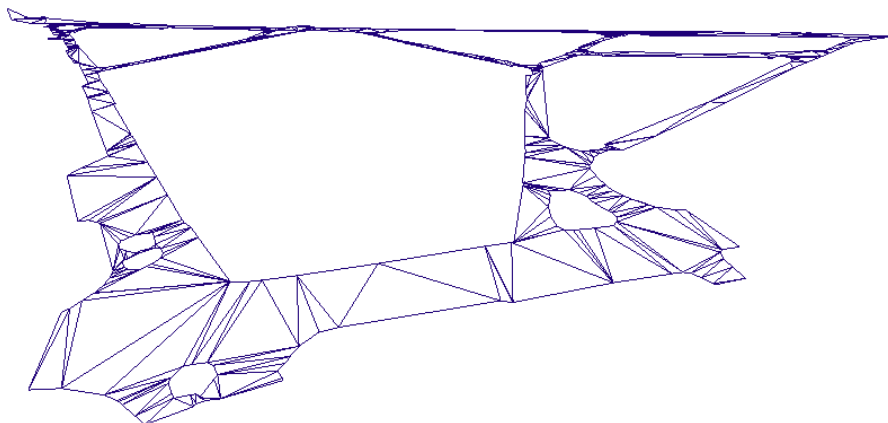


Figure 5.6.: Représentation en fil de fer du maillage de la chaussée.

Le bâti représenté sur la figure 5.7 est, quant à lui, au format CityGML. Il est composé d'un maillage 3D très précis des bâtiments simplifiés. Il a été réalisé en fusionnant les informations du cadastre et les données de télémètres lidar haute précision. Ces objets 3D sont de plus texturés avec des images provenant d'une caméra omnidirectionnelle composée de 16 caméras très haute résolution (1920x1080). La figure 5.5 représente la visualisation de la carte dans un moteur de rendu temps réel. Nous considérerons dans la suite de ces travaux que cette carte est parfaitement juste.

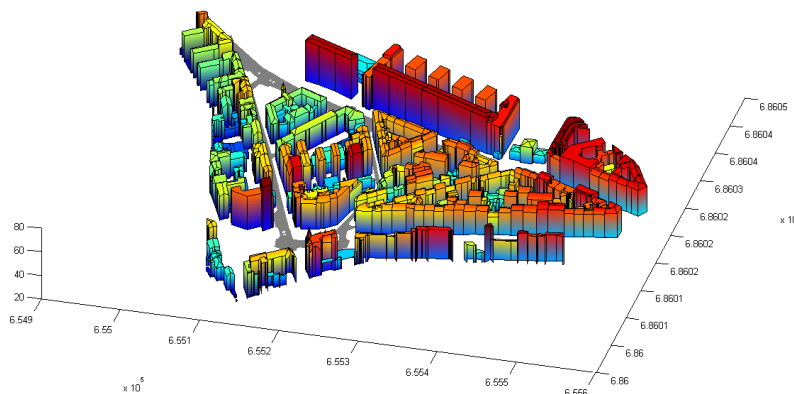


Figure 5.7.: Représentation 3D de la Carte CityVIP

Construction d'une "PriorGrid" On définit une grille appelée PriorGrid (PG) qui va contenir les informations provenant de la carte. Cette grille est de même taille et de même résolution que la MapGrid et chaque cellule contient une valeur entière qui représente soit le bâti (2), soit la chaussée (1) soit le reste (0).

Les éléments de la carte, à savoir les sommets des polygones du maillage, sont tout d'abord ramenés dans le repère ENU par une série de conversions sur les coordonnées :

- Lambert 93 vers coordonnées géographiques (lat, long, alt),
- coordonnées géographiques vers coordonnées ECEF (Earth Centred Earth Fixed),
- coordonnées ECEF (Earth Centred Earth Fixed) vers ENU CityVIP.

Pour chaque facette de la carte, on fait la projection orthogonale des sommets de cette facette dans le plan EN de la grille. Puis, pour chaque pixel de la grille, on met la valeur de cette cellule à :

- 2 si le centre de ce pixel est contenu à l'intérieur d'un polygone du bâti,
- 1 si le centre de ce pixel est contenu à l'intérieur d'un polygone de la route,
- sinon on la laisse à 0.

Cette procédure n'est faite qu'une seule fois à l'initialisation. Le résultat obtenu est représenté par la figure 5.8.

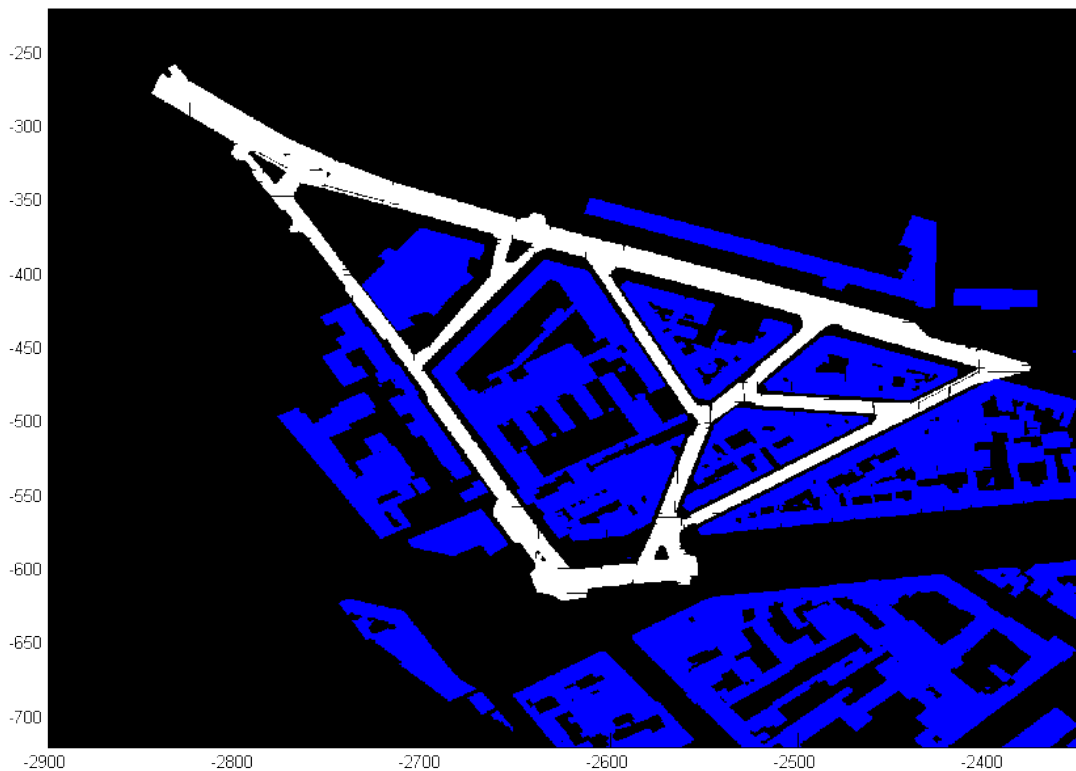


Figure 5.8.: Représentation de la grille obtenue à partir de la carte (PG). Les axes sont gradués en mètres. Le phénomène de pixellisation est nettement visible. Les cellules blanches représentent la route, les cellules bleues le bâti et les cellules noires le reste.

Utilisation de la PG comme un masque binaire On peut utiliser la carte pour filtrer les obstacles qui ne sont pas sur la chaussée. Dans ce cas, on construit une grille binaire à partir de la PG décrite ci-dessus en appliquant la règle suivante :

$$\text{Si } I_{ij}^{PG} = 2 \text{ alors } I_{ij}^{PG} \leftarrow 0.$$

A chaque fois que la MG est mise à jour, on la multiplie par la grille PG binaire. Ainsi, on élimine les cellules en dehors de l'espace navigable (sans tenir compte des contraintes réglementaires) afin de ne prendre en compte que les cellules utiles pour la navigation.

Utilisation de la PG comme une source d'information contextuelle Une autre approche consiste à utiliser les connaissances des cartes comme une source d'information pouvant être fusionnée avec les données des capteurs. Étant donné que le contenu sémantique des grilles d'occupation est différent de celui des grilles calculées à partir des cartes, il est nécessaire de redéfinir les cadres de discernement pour chacune des grilles utilisées. Ainsi, la grille PG est définie sur $\Omega_{PG} = \{R, I, B\}$, où R correspond à la route (valeur 1), I correspond à l'infrastructure (valeur 2) et B correspond au reste (valeur 0). Le cadre de discernement utilisé jusqu'à maintenant $\Omega = \{F, O\}$ est raffiné pour obtenir le cadre suivant : $\Omega_{MG} = \{F, C, N, S, V\}$ où F est l'espace libre, C les éléments de l'infrastructure cartographiés, N les éléments de l'infrastructure non cartographiés, S les objets mobiles à l'arrêt et V les objets en mouvement. Les ensembles focaux sont décrits dans la figure 5.9.

Une correspondance entre les éléments de Ω_{PG} et Ω_{MG} est faite à partir d'hypothèses à-priori, comme : “il n’y a pas de cellule mobile dans l’infrastructure” ou “Sur la route il n’y a que des objets en mouvement, des objets mobiles à l’arrêt ou de l’espace libre”, etc.

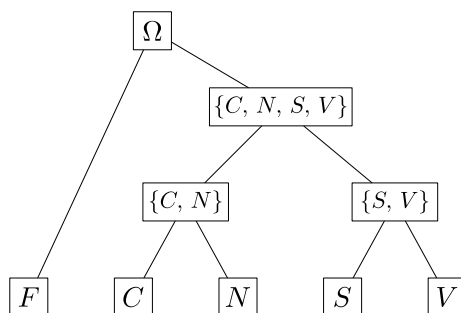


Figure 5.9.: Cadre de discernement raffiné

Dans un premier temps, la SG est conditionnée par la PG . Ceci permet de différencier, par exemple, les cellules occupées correspondant au bâti (où $I_{ij}^{PG} (I = 1)$) des

cellules occupées correspondant aux véhicules à l'arrêt (où $I_{ij}^{PG} (R = 1)$). Le résultat est ensuite introduit dans le mécanisme de mise à jour de la MG en considérant le cadre de discernement Ω_{MG} . Afin de transférer correctement les masses des cellules des objets mobiles, une première approche a été de construire une règle de fusion particulière à partir de la règle de Yager. Une description plus complète de cette règle est disponible dans [Kurdej, 2011].

Le niveau sémantique de la grille résultante est plus élevé grâce à l'injection d'information cartographique. Cependant, il faut souligner que l'augmentation de la taille du cadre de discernement accroît la complexité calculatoire de l'algorithme.

5.2.4. Implémentation

L'approche a été implémentée sous Matlab en utilisant les cadres d'accumulation et évidentiel.

La MapGrid est repérée dans le repère ENU du projet CityVIP. Sa résolution est de 0.5 m et ses dimensions sont de 800 m par 700 m. L'algorithme est synchronisé par rapport au télémètre. Ainsi, la MG est mise à jour à chaque fois qu'un nouveau scan du lidar est disponible. Dans ce cas, la SG polaire, de résolution 0.5 m x 1° et de dimensions 100 m x 180°, est créée en utilisant les modèles de capteur décrits dans la section 4.3. On transforme ensuite la SG dans un repère cartésien puis on la déplace dans le repère global en utilisant la pose, et enfin on fusionne les grilles. Les traitements appliqués aux grilles (transformation et fusion) sont réalisés sur toutes les cellules de façon indépendantes. Les figures 5.10a et 5.10b montrent l'architecture du système pour les deux méthodes.

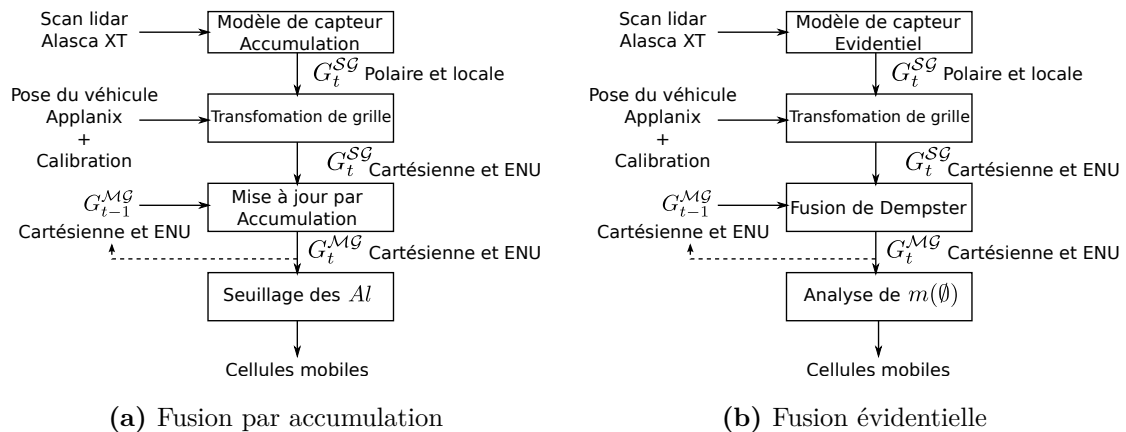


Figure 5.10.: Architecture du système halo-centrée

Chaque temps où un scan laser est disponible ne correspond pas exactement avec les instants des données de positionnement fournies par l'Applanix. Comme le processus est réalisé en traitement hors ligne, on peut calculer précisément les positions

qui correspondent aux instants des scans lidars par interpolation des positions de l'Applanix.

Concernant la méthode d'accumulation, chaque cellule contient un Al qui est borné. Cette méthode a l'inconvénient d'utiliser un grand nombre de paramètres qui sont assez difficile à fixer. Nous avons réalisé un ensemble d'expérimentations pour trouver des valeurs pertinentes. Nous avons ainsi fixé les bornes d'accumulation à 0 et à 30. Les cellules sont initialisées avec un $Al = 5$. Les coefficients de gain d'incrémentatation et des décrémentation sont fixés à 5 et à 1 respectivement.

Concernant la méthode évidentielle, nous avons considéré que l'angle d'ouverture du laser est $\varphi = 0,25^\circ$ et que la résolution angulaire du capteur (dans le pire cas) est $\beta = 0,5^\circ$ d'après la documentation du fabricant. Le taux de couverture, c'est à dire le rapport entre l'aire observée par les faisceaux lasers (supposés conique) et l'aire du plan est $\frac{\varphi}{\beta}$, cela nous permet de définir $\lambda_{MD} = 0.5$. D'autre part, une étude statistique nous a permis d'estimer λ_{FA} à 0.5.

Pour faciliter l'implémentation, la SG polaire couvre un champ angulaire de 360° et a une portée de 100 m, avec une résolution de 0.5 m par 1° . Cela permet de créer indépendamment de la direction une SG cartésienne carrée de 200 m de côté. C'est donc un bloc de 400 x 400 cellules qui est concerné par la fusion à chaque étape.

5.2.5. Résultats de la méthode d'accumulation

Cartographie Nous comparons la carte construite au cours de la réalisation d'une boucle de la trajectoire de référence (c.f. figure 5.3) avec la carte de la chaussée construite à partir des données fournies par l'IGN. Sur la figure 5.12, les cellules noires montrent les cellules occupées, et on a superposé, en gris, la carte de la chaussée pour faciliter la compréhension. Les cellules qui ne sont pas noires sont soit libres, soit inconnues.

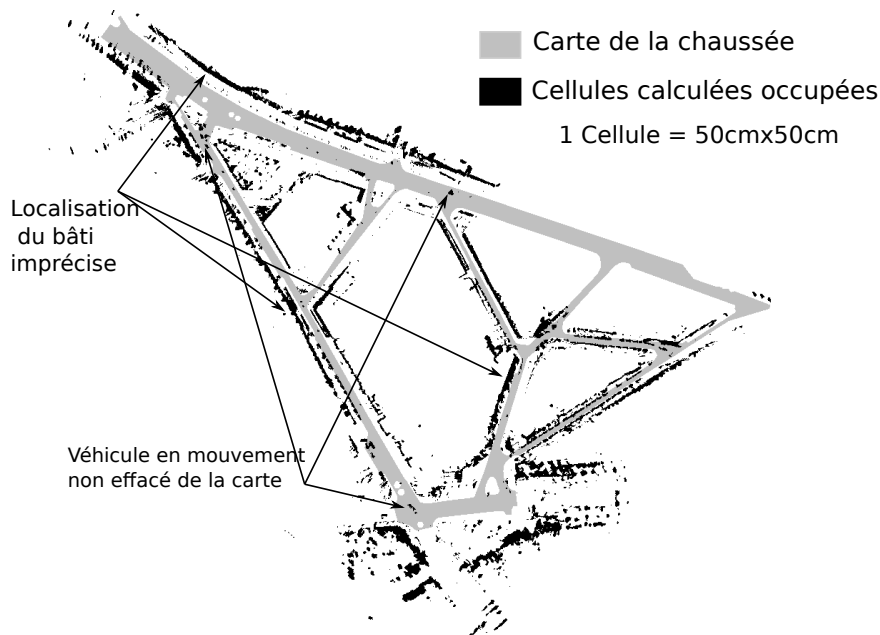


Figure 5.11.: Résultat de la cartographie utilisant le formalisme d'accumulation

Le point clé que nous voulons mettre en avant ici, est qu'en dépit d'un grand nombre d'objets en mouvement présents dans la scène, les objets statiques comme les bâtiments sont relativement bien cartographiés et, dans le même temps, l'espace libre est correctement déterminé malgré les objets qui sont passés dans le champ du capteur au cours du temps. La capacité à prendre en compte les masquages dans le modèle de capteur permet de cartographier plusieurs objets alignés dans un secteur angulaire à des distances différentes. Ce phénomène est visible à plusieurs endroits du parcours, notamment lorsqu'il y a des voitures garées devant un immeuble. Dans ce cas, les nappes supérieures du lidar sont utiles pour cartographier les bâtiments. La figure 5.12 présente la trajectoire avec le scan courant dans le repère ENU (en

haut à gauche), l'image de la scène depuis le point de vue du capteur (en haut à droite) et la grille d'accumulation avec la trajectoire locale (en fond d'image). Bien que la voiture (visible dans l'image) crée un masquage du bâtiment, les nappes supérieures du lidar permettent de bien cartographier le bâtiment ce qui est facilité par le montage du capteur dont la nappe du bas est parallèle à la route.

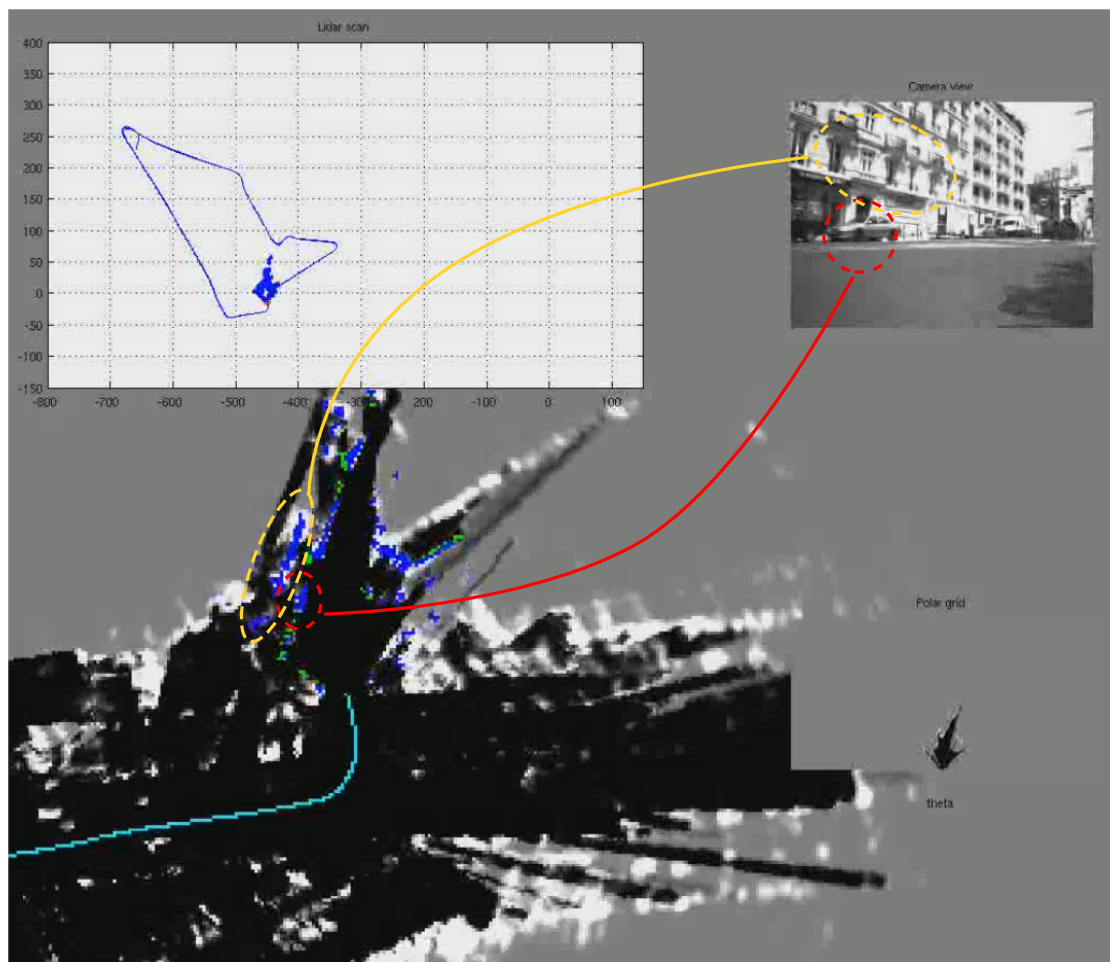


Figure 5.12.: Illustration de l'intérêt du multi-nappe

L'intérêt de la capacité multi-écho est illustré sur la figure 5.13. Le véhicule est dans un carrefour avec un trottoir protégé par les petits poteaux métalliques (à peine visible dans l'image). Ces poteaux (qui apparaissent en vert dans la *MG*) créent des occultations partielles qui n'empêchent pas de cartographier correctement l'infrastructure statique de la scène.

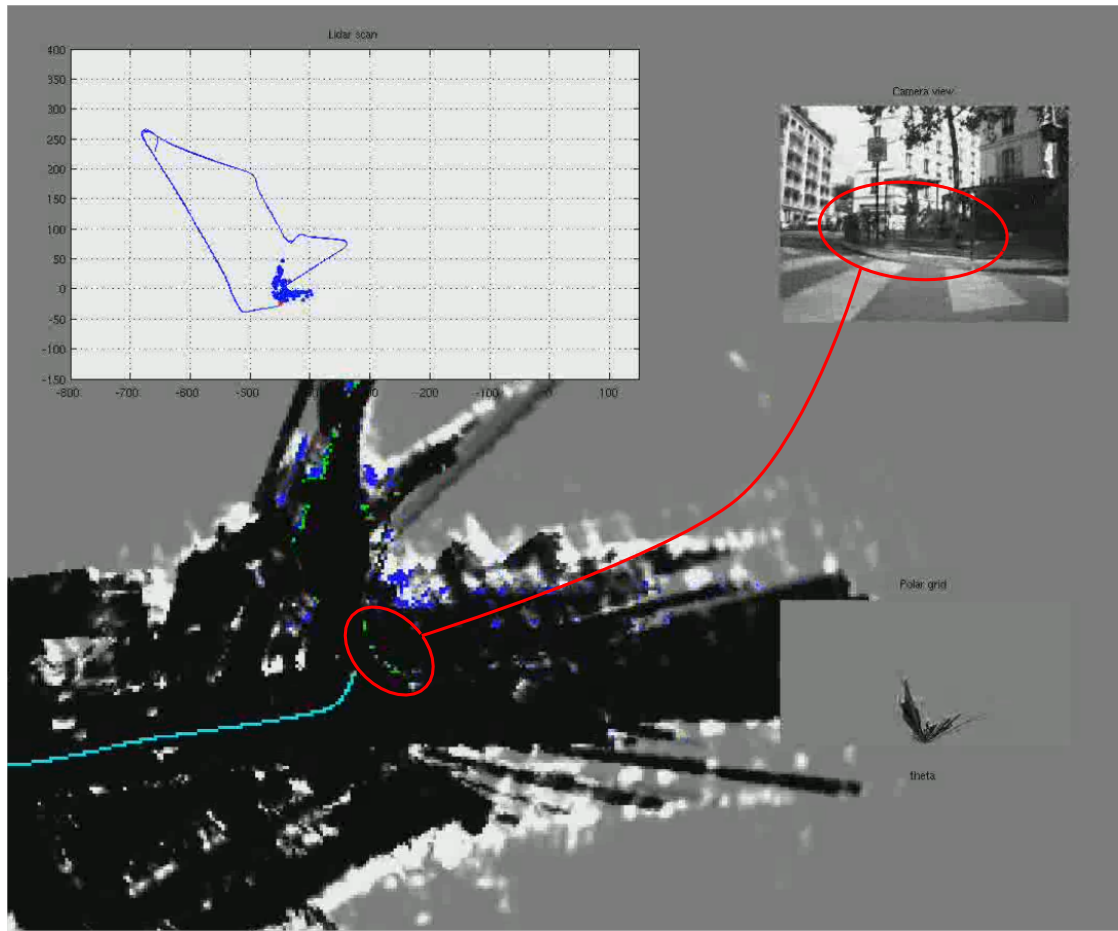


Figure 5.13.: Illustration de l'intérêt du multi-écho

On remarque quelques petits défauts de cartographie, comme des cellules sur l'espace libre qui restent occupées dans la carte. En effet, comme les objets mobiles ne sont pas suivis au cours du temps et il se peut qu'ils soient momentanément statiques (en s'arrêtant à un feu de circulation par exemple). Lorsque les objets mobiles quittent le champ de vision du capteur à cause du déplacement du porteur, ils ne sont plus mis à jour et restent cartographiés dans la carte. D'autre part, les petits objets (typiquement moins de 0.5 m ou objet creux) comme des barrières ou des poteaux indicateurs ne peuvent pas être correctement cartographiés à l'aide de la méthode proposée. Cela est dû essentiellement aux résolutions de la grille et du capteur. Ce phénomène est typique d'un sous-échantillonnage spatial.

Détection des cellules mobiles Nous avons extrait de la séquence deux scénarios qui montrent les performances de l'algorithme pour détecter les objets en mouvement dans différents cas. La classification en sortie est représentée sur la *GM*, en superposition avec la cartographie, en utilisant deux couleurs : le bleu et le vert. Les

cellules vertes représentent les cellules qui contiennent un objet se déplaçant tandis que les cellules bleues représentent les cellules statiques.

Pour chaque scénario, les lignes montrent l'évolution de la scène à différentes instants. Pour chaque ligne, il y a deux colonnes, la première colonne contient un zoom de la *GM* autour du véhicule avec les cellules classifiées. Dans la deuxième colonne, une vue de la scène est donnée (comme déjà précisé, la caméra a été installée juste sous le lidar). Les objets détectés sont mis en évidence manuellement ici en utilisant des boîtes englobantes afin de faire le lien entre les objets détectés dans le *GM* et de leur position dans l'image.

Considérons les deux scénarios suivants :

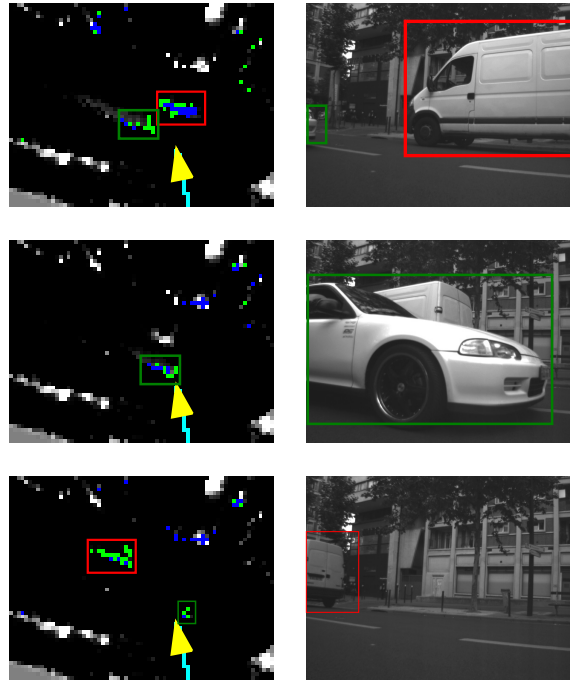
– Intersection

- Ce scénario est illustré par la figure 5.14b. L'ego-véhicule est arrêté à une intersection et deux véhicules se croisent devant celui-ci. Les véhicules proviennent de droite et de gauche avec une importante vitesse radiale. La voiture est correctement détectée puisque quasiment toutes les cellules sont vertes tandis que le fourgon est partiellement mal classifié avec un nombre important de pixels bleu. D'une manière général, certaines mauvaises classifications ont été observées dans le cas des véhicules longs, parce que leur longueur importante induit une accumulation spatiale pendant une longue durée ce qui provoque le dépassement du seuil.

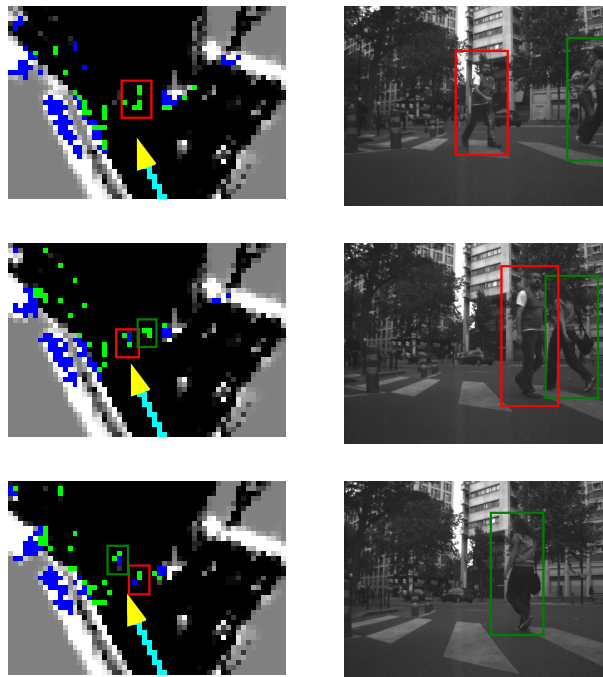
– Traversées de piétons

- Ce scénario est illustré par la figure 5.14a. Le véhicule s'approche d'une intersection et s'arrête pour laisser traverser la route à deux piétons. La classification des piétons en tant qu'objet en mouvement est partiellement fausse car ils marchent lentement (certaines cellules sont bleues). On peut remarquer que le champ de vision de la caméra (70°) est faible en comparaison avec le lidar (140°). Donc, les objets sur les cotés sont détectés par le lidar avant d'être visibles sur l'image.

Finalement, les résultats sont un peu mitigés : certains objets mobiles sont bien classifiés, d'autres ne le sont pas, soit parce qu'ils présentent une surface apparente grande pour le lidar, soit parce qu'ils sont lents. On pourrait bien sûr modifier les différents paramètres de la méthode (comme le seuil de détection d'objet mobile par exemple) mais cela créerait d'autres problèmes comme la génération de nombreux faux positifs dans le cas où on abaisserait le seuil par exemple.



(a) Scénario 1 : passage d'un véhicule



(b) Scénario 2 : traversées de piétons

Figure 5.14.: Résultats obtenus lors de deux scénarios illustrant la détection d'objets en mouvement. Les figures à gauche représentent la MG , les niveaux de gris correspondent au niveau d'accumulation Al , le bleu correspond aux objets actuellement détectés fixes et le vert aux objets détectés en mouvement. Le triangle jaune représente l'égo-véhicule et sa trajectoire réalisée est en cyan. Sur la droite, on retrouve une image de la scène avec un encadrement des objets concernés. Chaque ligne d'un scénario correspond à un instant donné.

5.2.6. Résultats de la méthode évidentielle

Cartographie La figure 5.15 présente la carte obtenue en utilisant la méthode évidentielle dans les mêmes conditions que pour la méthode d'accumulation. Comme dans le cas de l'accumulation, la cartographie a été réalisée en conditions de circulation réelle, c'est-à-dire avec la présence de nombreuses voitures et piétons. Les résultats de la méthode évidentielle sont sous forme crédale, c'est-à-dire que l'on affiche ici en noir $m(O)$ qui correspond à la croyance $bel(O)$ dans l'état O . Nous avons aussi superposé la carte de l'IGN en gris.

Concernant le bâti, qui est représenté par les cellules au bord de la chaussée, il est cartographié de la même façon qu'avec la méthode précédente. L'approche d'accumulation laisse de nombreuses cellules occupées disséminées sur l'espace libre. Ces phénomènes sont dus à des objets en mouvement. La méthode évidentielle est elle moins affectée car cette méthode offre un mécanisme qui permet une meilleure gestion des objets en mouvement. En effet, l'analyse du conflit permet de détecter des objets en mouvement sans les cartographier (la normalisation n'augmentant que très peu la masse d'occupé sur des cellules vides). Ainsi, ce mécanisme de mise à jour effectue un filtre rejetant les objets en mouvement (et par extension les bruits et mesures aberrantes) de la carte, tout en profitant des redondances spatiale et temporelle pour cartographier les objets fixes. La méthode d'accumulation ajoute les objets mobiles dans la carte et les efface quand ils quittent les lieux. Mais si un objet n'est pas observé quand il quitte une cellule, cette cellule restera occupée dans la carte.

On peut noter que la méthode évidentielle n'arrive toujours pas à cartographier les très petits obstacles pour les mêmes raisons évoquées précédemment.

Détection des cellules mobiles Nous avons comparé la capacité des deux méthodes à classifier les cellules fixes et en mouvement. La figure 5.16 montre les résultats de chacune des approches au cours d'une petite séquence. Cette séquence montre deux véhicules qui passent devant de l'ego-véhicule depuis la droite, puis un camion traverse depuis la gauche. Pour chaque instant de la séquence, on retrouve, en haut, l'image de la caméra filmant la scène, au milieu, le résultat de détection de la méthode d'accumulation (bleu : fixe et vert : mobile) et en bas les résultats de la méthode évidentielle (vert : mobile). Les cartes sont affichées dans le repère de la MG en vue de dessus. L'ego-véhicule est représenté par le triangle jaune, la pointe du triangle indiquant sa direction. Enfin, les véhicules ont été manuellement entourés de la même couleur sur l'image caméra et sur les grilles afin de les identifier facilement.

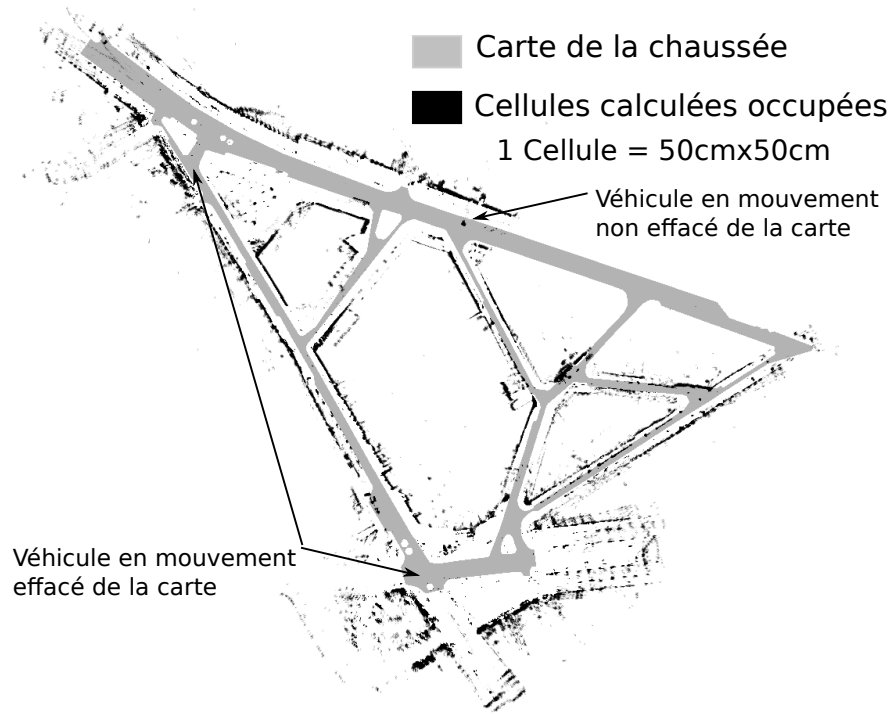


Figure 5.15.: Résultat de la cartographie utilisant le formalisme évidentiel

On s'intéresse dans un premier temps à l'approche d'accumulation. Sur cette séquence, elle présente des résultats assez médiocres. En effet, pour chaque véhicule, même si l'avant est partiellement détecté comme mobile (cellules vertes), ils sont pour la plupart classifiés comme statique (bleu). Comme on l'avait déjà vu dans les résultats présentés plus haut, cette méthode ne fonctionne pas très bien si la scène contient des objets avec des dynamiques différentes et la modification des paramètres de la méthode ne permettrait d'améliorer les résultats sur la classification du camion qu'au détriment de fausses alarmes générées par les erreurs de transformation de grilles.

L'approche évidentielle montre au contraire une très bonne classification de chaque véhicule (on peut remarquer par exemple que le camion est parfaitement classifié). Les objets mobiles ne sont pas restés assez longtemps sur les cellules pour que ces dernières deviennent occupées. Le principal inconvénient de cette méthode est la présence de nombreuses fausses alarmes sur les objets statiques causées par les

approximations lors des transformations de grille. Par exemple, sur le cliché 3, on peut remarquer la présence de deux objets juste à l'avant gauche du véhicule (non visibles sur la caméra) qui sont classifiés en mouvement alors qu'ils sont statiques et de petite taille (feux tricolore).

On peut conclure ici que l'analyse du conflit améliore grandement les performances de détection par rapport à l'approche d'accumulation. Les résultats sur la classification des cellules mobiles sont très bons. Malheureusement cela augmente le nombre de fausses alarmes sur les cellules fixes.

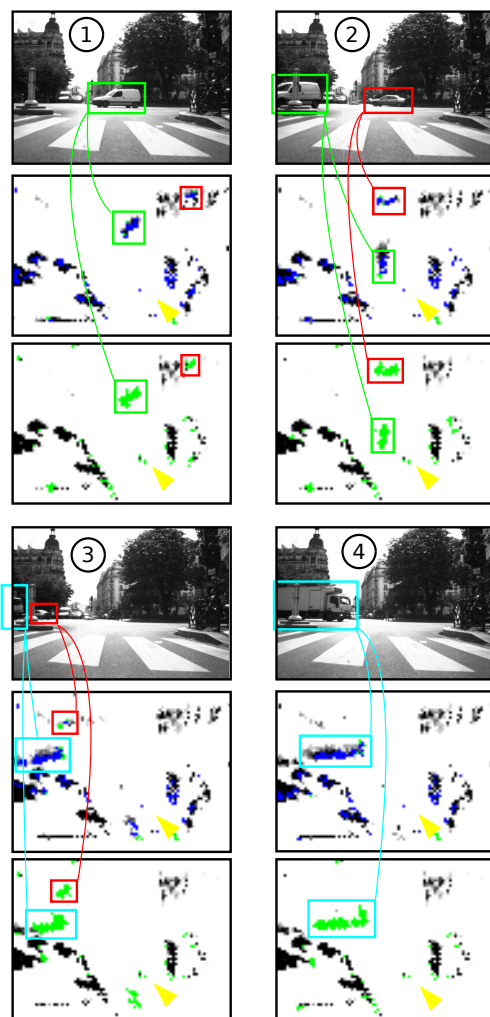


Figure 5.16.: Résultat de la détection d'objets mobiles des méthodes d'accumulation et évidentielle au travers d'une séquence de quatre clichés. Chaque cliché est illustré par une image de la caméra de la scène et deux grilles d'objets en mouvement. Pour l'approche d'accumulation (au milieu), le bleu montre le statique et le vert se réfère au mouvement et dans l'approche évidentielle (en bas), le vert fait référence aux objets en mouvement.

Capacité à extraire les objets mobiles sur la chaussée et à caractériser le sens de déplacement Nous mettons en avant, dans la figure 5.17, la capacité à détecter le mouvement dans la scène et à classifier si ce mouvement provient de cellules précédemment libres qui deviennent occupées (en rouge) ou de cellules précédemment occupées qui deviennent libres (en bleu). Nous avons introduit dans la méthode évidentielle un facteur d'affaiblissement α réglé de façon empirique, de façon à avoir une constante de temps de rémanence τ soit égale à 1.3 s. Cela a permis d'avoir de bon résultat et semble cohérent. Nous avons de plus filtré les résultats en utilisant la chaussée pour ne considérer que les obstacles sur la route (en blanc).

Chaque grille contient uniquement les cellules détectées mobiles. Pour faciliter l'interprétation, nous avons replacé les résultats dans le repère du véhicule. Ici, chaque grille représente une superficie de 50 m par 50 m située devant le véhicule, la résolution étant toujours de 0.5 m. L'égo-véhicule est représenté par le carré vert en bas de la grille et se dirige vers le haut. Sur la figure 5.17, on peut remarquer qu'un véhicule passe de droite à gauche, ce qui génère du conflit rouge à l'avant de celui-ci et du conflit bleu à l'arrière. Ce phénomène permet de retrouver le sens de déplacement.

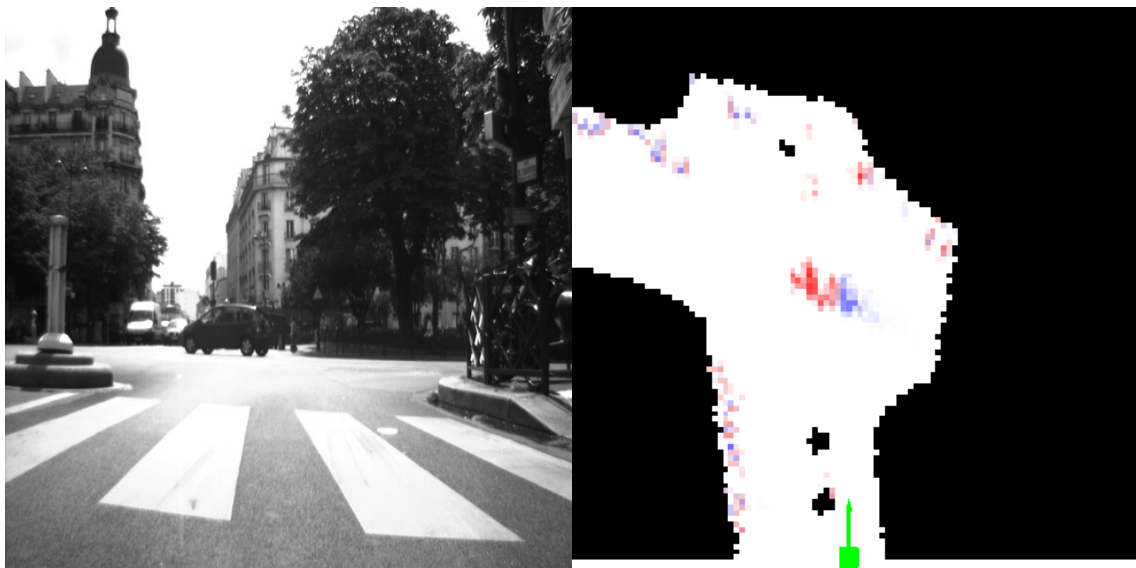
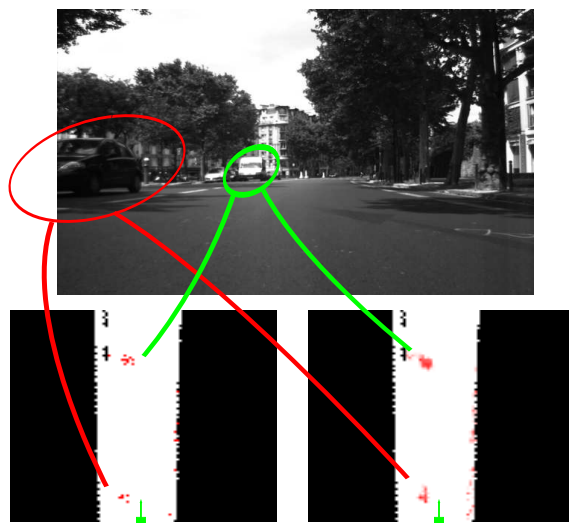


Figure 5.17.: Résultat de détection des objets en mouvement par l'approche évidentielle : à droite la grille ramenée dans le repère du véhicule.

Nous avons aussi comparé les résultats avec la méthode d'accumulation au travers d'une séquence illustrée par les figures 5.18a, 5.19a et 5.19b. A chaque pas de temps,

l'image de la caméra ainsi que deux grilles (à gauche la méthode d'accumulation et à droite la méthode évidentielle) sont affichées. Au cours de la séquence, le véhicule se déplace à la vitesse de 35 km/h. Il croise deux voitures et une camionnette.

- Sur la figure 5.18a, les deux méthodes réussissent à détecter correctement le fourgon. Le côté gauche de la voiture est plus reconnaissable avec l'approche évidentielle.
- La figure 5.19a montre un phénomène intéressant : la voiture au fond est correctement détectée par les deux méthodes. La fourgon est bien détecté par la méthode évidentielle, mais pose de gros problèmes à la méthode d'accumulation. La camionnette est bien vue par le lidar (les données utilisées sont les mêmes), mais la majorité des cellules est classée comme statique avec cette approche. La camionnette présente une grande surface de réflexion et cela produit un grand nombre de mesures lidar qui augmentent rapidement le niveau d'accumulation des cellules, ce dernier dépassant rapidement le seuil entre le mobile et statique.
- La figure 5.19b montre des résultats similaires avec une meilleure détection dans l'approche d'accumulation, mais la méthode évidentielle est toujours plus efficace. Nous pouvons voir la forme des véhicules et aussi voir l'effet de balayage du lidar. En effet, l'avant du véhicule le plus proche forme une pointe, comme un "V". Ceci est causé par le balayage comme décrit dans la section 4.2.2. Le balayage capteur étant dans le sens horaire, le côté gauche est scanné avant le côté droit.

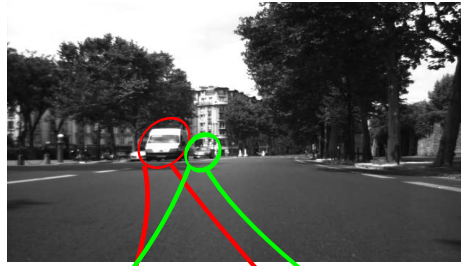


(a) Résultats de la détection des objets en mouvement : scène 1

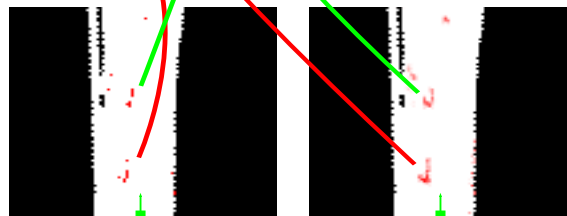
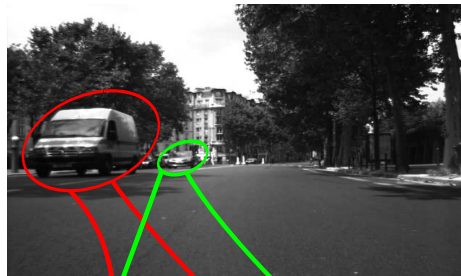
Figure 5.18.: Comparaison des capacités de détection des objets mobiles des méthodes d'accumulation et évidentielle. A gauche avec la méthode d'accumulation et à droite avec la méthode évidentielle. 1 / 2

De façon globale, les résultats de détection de la méthode évidentielle sont bien meilleurs que ceux de la méthode par accumulation. De plus, la méthode évidentielle

nécessite moins de paramètres de réglage qui sont de surcroît des paramètres de haut niveau facile à interpréter.



(a) Résultats de la détection des objets en mouvement : scène 2



(b) Résultats de la détection des objets en mouvement : scène 3

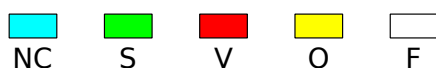
Figure 5.19.: Comparaison des capacités de détection des objets mobiles des méthode d'accumulation et évidentielle. A gauche avec la méthode d'accumulation et à droite avec la méthode évidentielle. 2 / 2

Augmentation du niveau sémantique de la grille par injection d'information cartographique

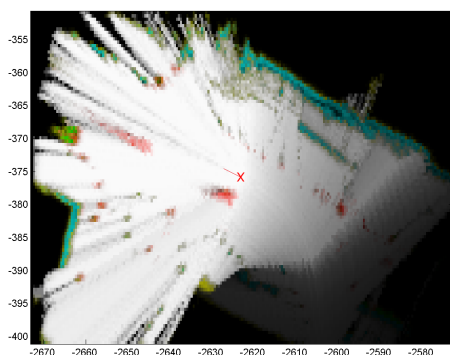
Pour améliorer la détection des objets en mouvement et pour distinguer les véhicules à l'arrêt des éléments de l'infrastructure, nous utilisons l'information cartographique comme une source d'information contextuelle (c.f. paragraphe 5.2.3). La figure 5.20 présente quelques résultats obtenus par cette méthode.

On rappelle que le cadre de discernement est $\Omega_{MG} = \{F, C, N, S, V\}$ et on affiche dans chaque cellule le maximum de probabilité pignistique.

On peut remarquer sur la première grille que les deux véhicules en mouvement (rouge) sont bien détectés. Sur les deux grilles, le bus garé sur la voie de gauche et visible dans le fond de l'image, est classifié comme un objet mobile à l'arrêt (vert). L'infrastructure de couleur cyan est aussi correctement classifiée et on voit bien la différence avec le bus. Les cellules en jaune correspondent à un maximum de probabilité pignistique sur le sous ensemble caractérisant l'occupation $O = \{C, N, S, V\}$ pour lequel il n'y a pas d'information permettant de choisir l'une de ses sous-classes.



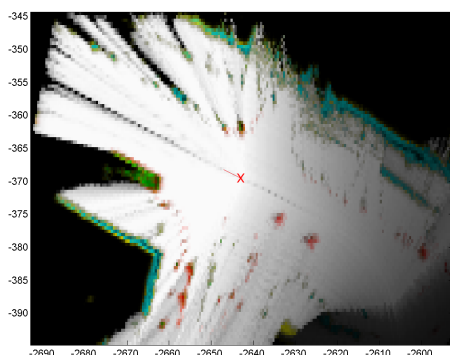
(a) Légende des couleurs



(b) Grille calculée au cliché 1, l'égo-véhicule est représenté par la croix rouge au centre.



(c) Image du cliché 1. Le second véhicule en mouvement est en dehors du champ de la caméra.



(d) Grille calculée au cliché 2, l'égo-véhicule est représenté par la croix rouge au centre.



(e) Image du cliché 2

Figure 5.20.: Résultat de fusion contextuelle

Ces résultats montrent qu'il est possible de mieux analyser le contenu des cellules en raffinant la classification grâce à la carte.

5.3. Fusion séquentielle égo-centrée

La fusion séquentielle égo-centrée a été développée dans un prototype embarqué temps-réel. Il a fait l'objet de plusieurs démonstrations, notamment à la conférence IEEE Intelligent Vehicles 2012 à Baden-Baden en Allemagne.

5.3.1. Rappel du principe

La méthode égo-centrée, présentée dans la section 2.4.4.2, utilise deux grilles : la Carte Locale Dynamique (*CLD*) qui est dans le repère du véhicule et la *SG* qui est dans le repère du capteur. L'architecture du système est représentée sur la figure 5.21. La *CLD* se déplace avec le véhicule, il faut donc la propager à chaque fois que le véhicule se déplace. La *SG* est construite, à chaque instant, à partir d'un scan lidar en utilisant le modèle de capteur évidentiel présenté en section 4.3.4. On utilise l'algorithme de détection des points du sol pour classer les points du scan. Pour les points du sol, on utilise le modèle de capteur du sol et pour les autres, on utilise le modèle d'obstacle. Puis cette *SG* polaire est convertie en repère cartésien et replacée dans le repère véhicule. Les deux repères étant alignés et la translation étant arrondie à la cellule près, il n'y a pas d'interpolation de grille.

On calcule ensuite le déplacement du véhicule entre l'instant courant et l'instant de la dernière mise à jour. La *CLD* précédente est déplacée pour compenser le déplacement en translation et en rotation du véhicule. Une partie des cellules disparaissent et de nouvelles rentrent dans le domaine de la nouvelle grille. Ces nouvelles cellules sont initialisées avec une fonction de masse vide ($m(\Omega) = 1$ ignorance totale). On applique ensuite un affaiblissement sur la *CLD* pour prendre en compte la dynamique de la scène. Puis, on fusionne la *SG* avec la *CLD* pour faire la mise à jour de cette dernière.

Si on dispose d'une pose absolue et d'une carte, on l'utilise pour filtrer les cellules en dehors de la chaussée. Les cellules fortement libres représentent alors l'espace navigable, les cellules fortement occupées les obstacles. La méthode de gestion du conflit fournit les cellules qui contiennent des objets en mouvement. Enfin, on réalise la projection dans l'image d'une caméra grand angle pour l'affichage et l'analyse des résultats.

5.3.2. Présentation de la plate-forme expérimentale

Le laboratoire Heudiasyc dispose d'une plate-forme expérimentale PACPUS (Perception et Assistance Pour une Conduite PIUs Sûre) qui est composée de moyens

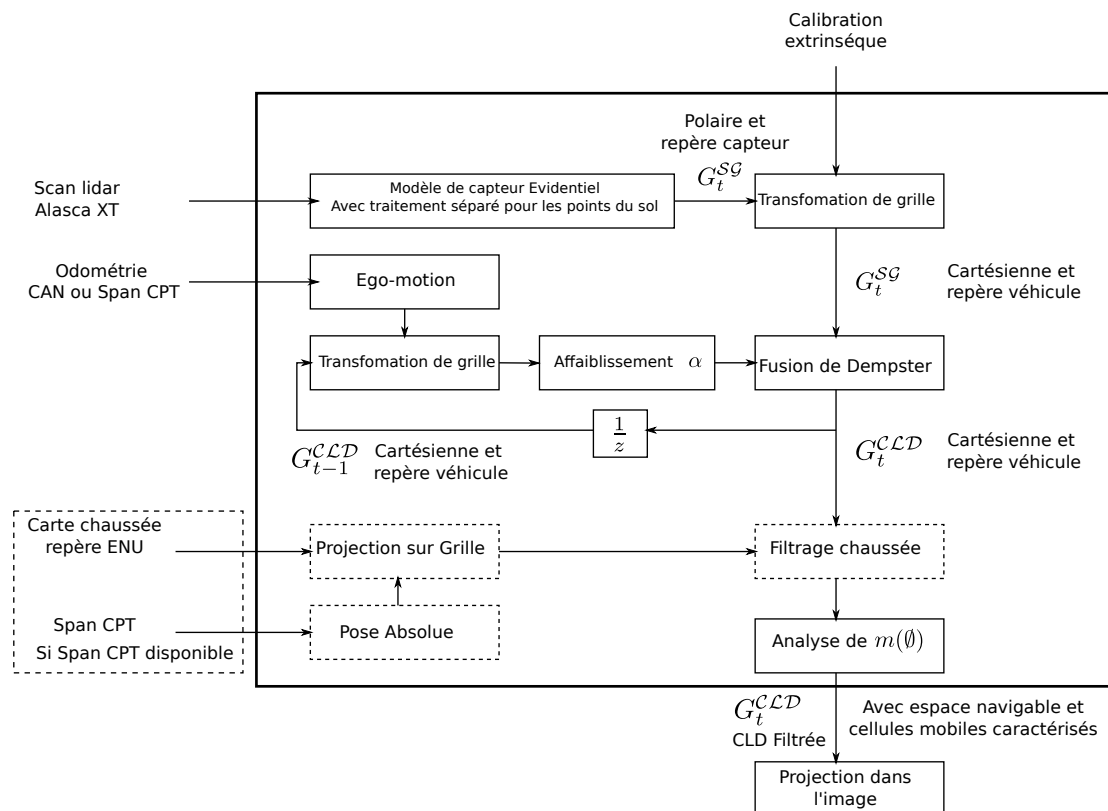


Figure 5.21.: Architecture du système égo-centré

logiciels et matériels (véhicules expérimentaux instrumentés) pour mener à bien des recherches et des expérimentations dans le domaine des véhicules intelligents.

Plate-forme matérielle Le véhicule utilisé pour ces travaux est le véhicule Carmen (figure 5.22) du laboratoire Heudiasyc.

Ce véhicule est instrumenté pour la perception et est équipé de plusieurs capteurs proprioceptifs et extéroceptifs. Dans ces travaux, les capteurs utilisés sont le lidar IBEO Alasca XT, les capteurs odométriques du véhicule via une passerelle avec le BUS CAN, un système de positionnement IMU-GPS Novatel SPAN CPT et une caméra Sony pour la visualisation des résultats. Le véhicule est aussi équipé d'un système de stéréo-vision Videre, d'un radar 77 GHz, d'un récepteur GPS haute qualité Septentrio PolarX et d'un télémètre laser à balayage à 64 faisceaux Velodyne 64 HEL qui ne sont pas utilisés dans ces travaux.

Tous ces capteurs sont connectés à un PC via différentes interfaces (Série, Ethernet, USB et Firewire). Ce PC est équipé d'un processeur Quad Core du Type Intel Quatro, de 4 Go de mémoire vive et d'une carte Graphique Nvidia Gforce 8400GT.



Figure 5.22.: Le véhicule expérimental Carmen : le lidar est intégré dans la calandre du véhicule

Système de positionnement L'approche proposée a besoin de connaître le déplacement du véhicule et éventuellement sa position globale si une carte est utilisée. Le véhicule CARMEN est équipé de plusieurs systèmes de positionnement qui peuvent être utilisés indifféremment : nous les présentons dans ce paragraphe.

Odomètres du bus CAN Le véhicule est équipé d'un ensemble de capteurs utilisés pour les systèmes d'aides à la conduite embarqués d'origine développés par le constructeur. Le système ABS (Anti-lock Braking System) utilise des codeurs de roue et le système ESP (Electronic Stability Program) utilise en plus un gyromètre de lacet. Ces capteurs sont des capteurs bas coût à performance modeste. Ils communiquent avec l'ordinateur de bord via une interface CAN (Controller Area Network) [Lawrenz, 1997]. Une passerelle CAN permet de faire le lien entre le bus CAN et le PC.

On peut utiliser un modèle de véhicule pour reconstruire le mouvement relatif du véhicule. Par opposition au GPS qui calcule une trajectoire absolue, ici la trajectoire est relative par rapport à la position de départ. Donc, si on connaît la pose initiale, on peut reconstruire la trajectoire dans un repère absolu. Mais, cette méthode diverge et la pose absolue fournie est rapidement erronée. Par contre, la trajectoire reconstruite par intégration est continue et assez lisse, ce qui est une très bonne propriété pour notre méthode. Généralement, on met en place un filtrage de type filtre de Kalman [Bonnifait *et al.*,] pour estimer l'état du véhicule à partir des mesures odométriques recalées par un capteur absolu.

Centrale inertielle avec récepteur GPS Le système GPS (Global Positioning System) est un système satellitaire de positionnement terrestre mis en place par le département de la défense des États-Unis d'Amérique. Le système GPS est capable de fournir une position, mais n'est pas capable de fournir un cap à l'arrêt. Ce système n'est pas localement précis car, il est sujet à des sauts GPS causés par divers phénomènes, comme des multi-trajets ou des changements du nombre de satellites en vue. Cela induit que la position fournie est généralement juste (ou faiblement biaisée) mais imprécise. Ce type de positionnement seul n'est pas toujours intéressant pour l'utilisation de grilles car la prédiction peut subir des sauts surtout lorsque des satellites sont masqués.

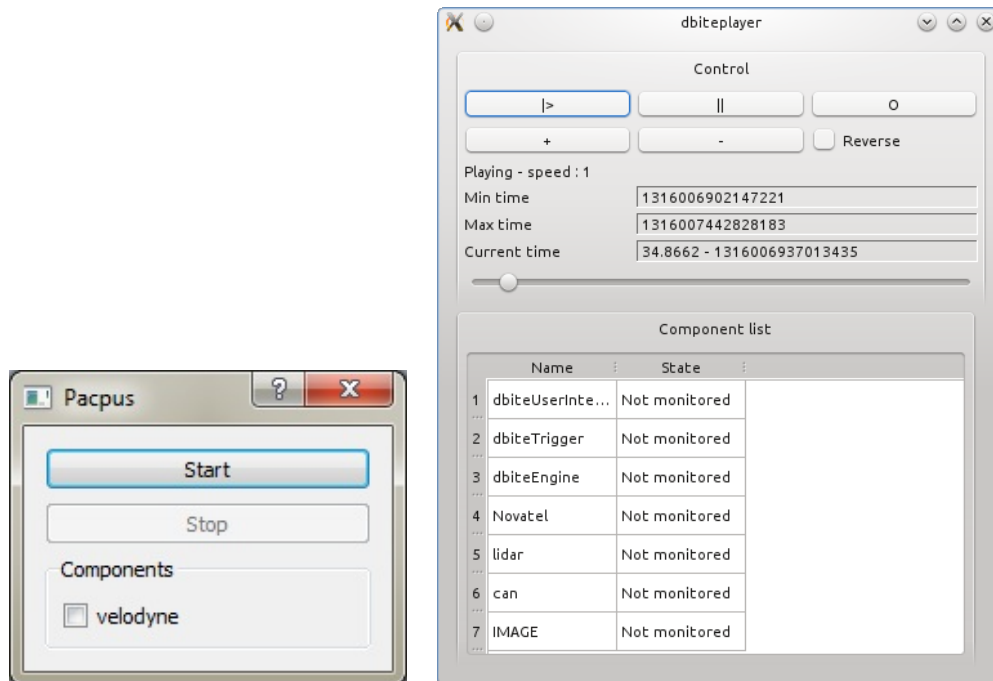
Bien que l'utilisation de GPS seul n'est pas pertinente avec l'approche proposée, l'utilisation d'une solution hybridée alliant un GPS à des capteurs inertiels permet de corriger les défauts du GPS seul. Ainsi, le système de positionnement Novatel Span CPT (voir figure 5.23) est comme le système Applanix composé d'une centrale inertielle couplée à un récepteur GPS équipé d'une antenne Trimble de bonne qualité. La centrale est équipée de gyromètres de bonne qualité à fibre optique et d'accéléromètres bas coûts utilisant la technologie MEMS (Micro Electro-Mechanical Systems). Le système GPS est capable d'utiliser les systèmes d'augmentation SBAS (Satellite-Based Augmentation System), les signaux en bande L ainsi que les corrections RTK (Real Time Kinematic) afin d'augmenter la précision de la pose. Enfin, la centrale possède une entrée odométrique, qui n'a pas été utilisée faute d'interface matérielle. Il est à noter que le système est capable de fournir une pose à 100 Hz même en cas de masquage GPS ce qui en fait un capteur précieux pour les applications dynamiques en environnement complexe introduisant de nombreux masquages.



Figure 5.23.: Système de positionnement : Span CPT

Plate-forme logicielle Pour permettre l'implémentation en temps réel, le système informatique utilise une plate-forme logicielle (appelée aussi intergiciel) développée au sein du laboratoire. Pacpus contient un ensemble d'outils logiciels, bibliothèques et exécutables, qui permettent la création, la gestion et la communication de plusieurs composants indépendants au sein d'une application temps réel. Il permet notamment la récupération des données capteurs en temps réel, leur enregistrement ainsi que le rejeu des données enregistrées, à vitesse réelle ou ralentie (figure 5.24). L'un des points forts de cet intergiciel est que le rejeu est réalisé de façon transparente pour

l'application cliente. Pacpus est développé sur les architectures X86 et X86_64 et est compatible avec plusieurs systèmes d'exploitation (Windows, Mac OS, Linux) et possède une licence Open source.



(a) Exemple d'interface entre les capteurs et les applications réalisant notamment l'enregistrement des données.

(b) Dbiteplayer : outil de rejeu de données

Figure 5.24.: Principaux outils de la plate-forme logicielle de PACPUS

Les travaux de cette thèse ont participé à l'amélioration de cette plate-forme ainsi qu'à l'interfaçage de plusieurs bibliothèques très utilisées dans le domaine robotique telles que OpenCV [Bradski et Kaehler, 2008] et PCL [Rusu et Cousins, 2011] mais aussi l'utilisation du calcul GPU en OpenCL [Opencl *et al.*, 2011].

5.3.3. Implémentation

Nous avons implémenté cette méthode en C++ au travers de la plate-forme Pacpus. Nous avons utilisé la méthode de fusion évidentielle dans cette implémentation. Avec cette méthode, une cellule contient 3 valeurs (il y a 4 masses dont la somme fait 1). Pour optimiser les calculs, nous avons utilisé une structure informatique de type image couleur avec 3 canaux RGB pour manipuler les grilles. Nous avons utilisé la bibliothèque OpenCV pour réaliser toutes les opérations sur les grilles, cette bibliothèque permettant d'optimiser et simplifier la programmation. Nous avons implémenté un

ensemble de classes informatiques pour les grilles qui contiennent la notion de grille métrique dans les structures images d'OpenCV.

L'application a été conçue de façon modulaire sous forme de composants indépendants. Chaque composant récupère sa configuration dans un fichier xml au moment de l'exécution. Cela permet notamment de paramétrer toute la grille (taille résolution, etc.). Un exemple de fichier xml est donné ci-dessous. On retrouve les paramètres des deux grilles comme le domaine spatial, les pas d'échantillonnage et les paramètres de l'algorithme.

Listing 5.1: Exemple de fichier XML

```
<pacpus>
<components>
  <LidarInterface type="LidarInterface" />

  <ComputingComponent type="ComputingComponent" />

  <Lot1Component type="Lot1Component" verbose="false" output_file="false" />

  <CamInterface type="CameraInterface" width="640" height="480" format="color" />

  <CLDInterface type="CLDInterface" xmin="-15" xmax="15" ymin="-15" ymax="15" dx=
    "0.1" dy="0.1" rmin="0" rmax="20" thmin="-180" thmax="180" dr="0.1" dth=
    "1.0" GlobalePose="true" xlidar="0.0" ylidar="2.0" LidarCentred="true"
    lidarOutlayerMinRange="1.5" lidarSigma="0.25" DecayFactor="0.95" />

  <MapInterface type="MapInterface" roadFormat="txt" roadCoordinates="lambert93"
    roadFilePath="ClosedLoopMap.txt" buildingsFormat="gml" buildingsCoordinates=
    "lambert93" buildingsFilePath="Mairie_XII_short.gml" geoidFilePath="raf98.
    txt" ui="true" latRef="48.844441730555560" longRef="2.425018041666667" altRef
    ="126.244" />
</components>

<parameters>
  <plugins list="libCLD.so|libLot1Component.so" />
</parameters>
</pacpus>
```

Pour chaque capteur utilisé, un composant récupère les données au moyen d'un *thread* qui scrute l'arrivée de nouvelles données. L'algorithme est cadencé sur le lidar. C'est le composant dédié à la récupération des données lidar qui exécute les calculs de grille. Pour rendre indépendants les codes de l'algorithme des codes du composant lidar, ce dernier récupère automatiquement la fonction à utiliser située dans le composant principal.

Concernant le système de localisation, il a été développé pour s'interfacer avec le système d'odométrie CAN ou avec le système Span CPT hybridant GPS et centrale inertielle. Si on utilise le système Span CPT, alors la pose absolue fournie permet l'utilisation des informations cartographiques.

On trouve ci-dessous la liste des 9 composants Pacpus qui composent le programme :

Le composant principal :

- ComputingComponent

Ce composant principal est utilisé pour gérer les différents composants du pro-

gramme. Il initialise tous les autres composants et il contient la fonction principale qui fait les liens entre tous les composants.

Les composants qui récupèrent les données des capteurs :

– LidarInterface

Ce composant est chargé de récupérer les données lidar dans une mémoire partagée. C'est ce composant qui contient le thread de calcul principal, ce composant récupère la fonction principale à partir du ComputingComponent.

– CameraInterface

Ce composant récupère les images de la caméra.

– Lot1Component

Ce composant récupère les données du système de positionnement Span CPT et convertit la pose en structure "Lot1" définie dans le cadre du projet CityVIP.

– CANComponent

Ce composant récupère les données du bus CAN et il effectue le décodage des trames.

Les composants de calcul :

– CLDInterface

Ce composant implémente les fonctions de calcul de grilles. A l'initialisation, il instancie les structures de grilles, puis il possède une fonction de calcul qui, à partir du scan lidar courant et de l'égo-motion du véhicule, met à jour la grille. C'est ce composant qui réalise la création de la SG, les transformations de grille, l'affaiblissement et la fusion.

– OdometrieComponent

Ce composant utilise un composant CANComponent pour récupérer les données des capteurs odométriques à partir du bus CAN, puis il calcule la translation et la rotation entre la position courante et la dernière position reçue. Ce composant possède un unique thread.

– ProjectorComponent

Ce composant permet de calculer la projection de la grille dans l'image de la caméra.

– MapInterface

Ce composant permet le chargement des cartes ainsi que leur conversion dans un repère ENU. Il implémente aussi la méthode de rasterisation par GPU.

Exploitation de la carte en temps réel L'utilisation des cartes en temps réel est différente de celle présentée pour la méthode halo centrée. Pour éviter les approximations d'interpolation, on recalcule la PriorGrid, à chaque itération, en utilisant la pose absolue courante (Span CPT). La carte vectorielle est transformée dans le repère local puis, elle est rasterisée (c'est-à-dire projetée sur une image matricielle) à chaque étape. Nous avons utilisé des optimisations OpenGL pour réaliser cette étape en temps réel en utilisant le processeur graphique. Les sommets sont passés en paramètre à la carte graphique pour construire un ensemble de triangles. On affecte des couleurs différentes aux triangles de la chaussée et aux triangles du bâti.

A chaque instant, nous positionnons la caméra virtuelle à la verticale du repère véhicule en utilisant la pose du véhicule. Cette caméra est dirigée vers le bas comme illustré sur la figure 5.25 et placée en orientant le haut de l'image suivant le cap du véhicule. Nous utilisons un modèle de camera à projection orthogonale ce qui permet de positionner la caméra à une altitude quelconque sans influencer le résultat de la projection (voir figure 5.26). Le rendu de l'image permet de récupérer la grille dans le FrameBuffer¹ de la carte graphique. On définit la taille de l'image du FrameBuffer et les paramètres du Viewport² de telle manière à ce qu'elle soit compatible en taille et en résolution avec la CLD . La scène étant composée de quelques dizaines de milliers de triangles, le rendu est très rapide avec les cartes graphiques actuelles.

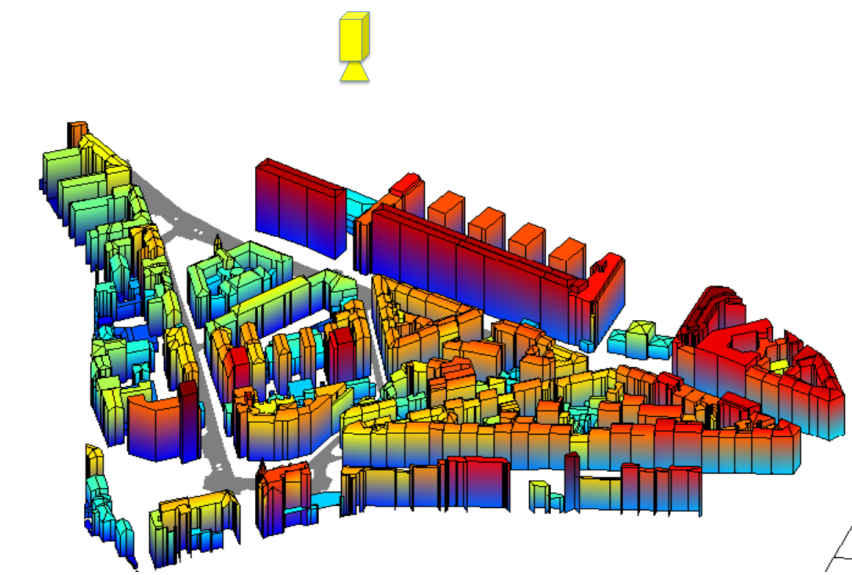


Figure 5.25.: Positionnement de la caméra virtuelle pour faire le rendu de la carte.

5.3.4. Méthode de validation expérimentale

La validation et l'analyse des performances d'un système de perception temps réel est assez difficile à réaliser. Dans sa thèse [Wang, 2004], C-C Wang a utilisé trois caméras pour effectuer la validation de ses résultats. Nous proposons une amélioration qui permet une validation qualitative en temps réel en projetant en transparence la carte locale dynamique estimée dans l'image de la scène. Cela permet une analyse visuelle dense. De cette façon, la justesse de l'approche est analysée par la correspondance

-
1. C'est la zone mémoire de la carte graphique qui contient l'image résultante à afficher.
 2. Il définit la façon dont les coordonnées sont mappées entre l'espace métrique et l'image.

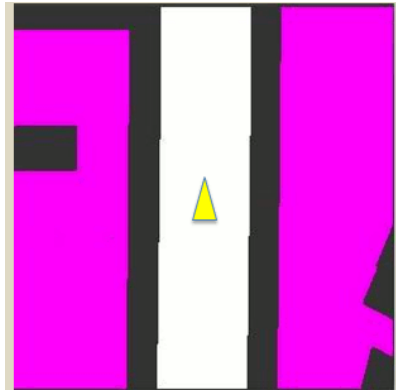


Figure 5.26.: Reprojection de la carte dans une grille pour créer la PriorGrid. Le blanc représente la chaussée, le violet le bâti et le noir le reste.

entre les résultats obtenus et la scène observée. On note cependant que les cellules masquées ne peuvent pas être évaluées de cette façon.

Pour mettre en œuvre cette méthode dans des conditions maximales, une caméra grand-angle a été installée derrière le pare-brise. Son champ de vision était presque identique à l'ouverture du lidar. Une procédure d'étalonnage qui a été préalablement réalisée [Rodriguez Florez *et al.*, 2008](procédure hors ligne) permet de déterminer la matrice de transformation relative entre la caméra et le repère véhicule ${}^C T_M$ et la matrice K des paramètres intrinsèques de la caméra en utilisant un modèle classique sténopé projectif supposant l'absence de distorsion. La projection est réalisée en calculant la matrice d'homographie H qui permet de passer du plan de la grille au plan image :

$$H = K \cdot {}^C T_M$$

Cela permet d'exprimer les coordonnées dans l'image des sommets de la grille de la façon suivante :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{ij} \sim H \cdot P_{i,j} \text{ with } j = 1 : 4$$

où $P_{i,j}$ représente les coordonnées 3D du j -ième sommet appartenant à la i -ième cellule de la grille. Comme la géométrie du système est fixe (grille et capteurs), ce calcul est réalisé une seule fois. Cela permet de reconstruire les polygones qui correspondent aux cellules dans l'image comme illustré sur la figure 5.27.

Finalement la couleur RGBA d'un polygone est déterminée par la valeur de la fonction de masse de la cellule. L'espace navigable est calculé comme étant l'intersection

de l'espace libre et de la contrainte de carte (qui représente la chaussée ici). Il est représenté en vert. Les cellules occupées et les cellules supportant du conflit qui sont situées sur la chaussée sont représentées respectivement en rouge et en bleu. Enfin, les cellules inconnues (masse sur Ω) et les cellules situées à l'extérieur de la chaussée sont transparentes.

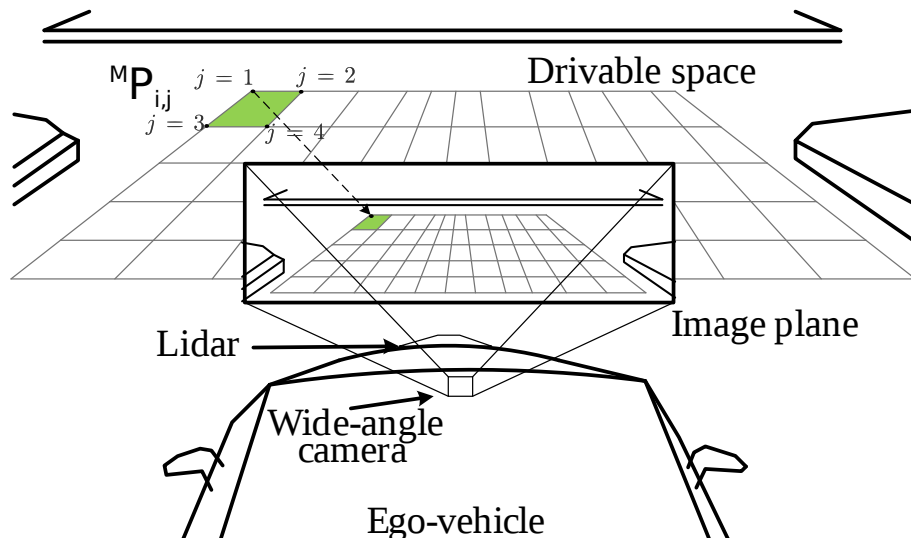


Figure 5.27.: Projection de la grille dans l'image de la camera

La qualité de l'analyse réalisée dépend fortement de la précision de la calibration des capteurs.

5.3.5. Résultats

De nombreux tests et démonstrations ont été réalisés. Les résultats présentés dans ce paragraphe proviennent de la journée de démonstration finale du projet CityVIP réalisée en septembre 2011 à Paris. La configuration utilisée comporte le système de localisation Span CPT, une caméra grand angle installée au niveau du pare brise. Le lidar Alasca XT qui était cadencé à 15 Hz et était orienté de façon à avoir deux nappes qui touchent le sol. Nous avons utilisé une grille qui s'étend à 15 mètres autour du véhicule avec une résolution de 0.1 m. Cette configuration permet de faire fonctionner le programme de façon confortable, puisque nous avons pu atteindre 60 Hz en rejeux accéléré. Afin d'illustrer les résultats de la méthode, les figures qui suivent montrent la fenêtre principale de l'application embarquée. Dans cette fenêtre, on trouve la CLD calculée et sa reprojction dans l'image de la caméra.

Sur la figure de droite, la représentation de la CLD est donnée en vue de dessus, le véhicule étant représenté par le triangle jaune au centre. Le blanc signifie que les cellules sont soit à l'extérieur de la chaussée ou soit à l'extérieur du domaine de la CLD. Sur les cellules à l'intérieur, le vert représente l'espace libre, le rouge repré-

sente les obstacles, le bleu représente le mouvement et le noir représente l'inconnu. L'intensité des couleurs représente la valeurs des masses sur les éléments.

Sur la figure de gauche, on trouve l'image de la caméra et la CLD reprojétée en transparence. Les zones inconnues ou hors de la route sont complètement transparentes.

Dans le premier exemple de la figure 5.28, un piéton traverse la route. On peut voir premièrement que le piéton est bien détecté, et deuxièmement, que l'utilisation de la carte permet de filtrer les zones en dehors de l'espace navigable. Finalement, la zone en vert représente la zone dans laquelle on peut naviguer.

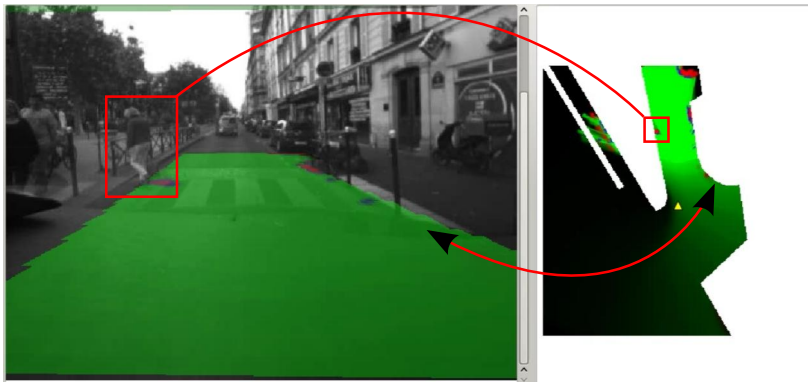


Figure 5.28.: Résultat de la construction de la CLD : un piéton traverse la route.

Sur l'exemple de la figure 5.29, on peut remarquer que les objets fixes sur la chaussée sont bien détectés, ce qui délimite d'avantage l'espace navigable. On peut remarquer quelques cellules bleues le long des véhicules à l'arrêt. Ceci est dû aux imprécisions de l'estimation du mouvement propre et aux erreurs produites par les phénomènes de discrétisation et de transformation des grilles. De plus, on peut voir clairement l'effet de l'affaiblissement temporel sur les cellule situées derrière le véhicule. Plus les cellules ont quitté le champ de vue du capteur depuis longtemps (et sont donc loin derrière le véhicule), plus ces cellules sont affaiblies, c'est-à-dire que la masse est progressivement transférée sur Ω (en noir).

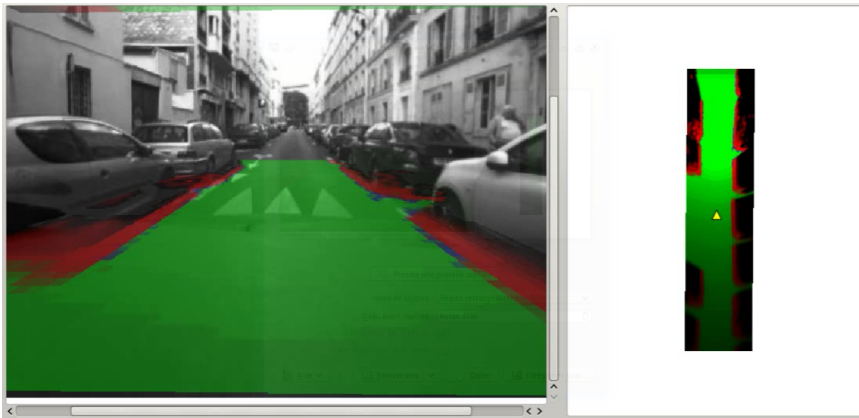


Figure 5.29.: Résultat de la construction de la CLD : des véhicules sont garés sur les bords de la route.

On peut remarquer sur l'exemple de la figure 5.30 que les objets mobiles, comme ici le vélo, génèrent un conflit important (cellules bleues) à l'avant et à l'arrière. Ils sont donc facilement identifiables par ce moyen. A cause de la limitation du nombre de canaux de couleur (RGB) dans les images, nous n'avons pas pu distinguer à l'affichage les deux types de conflit.

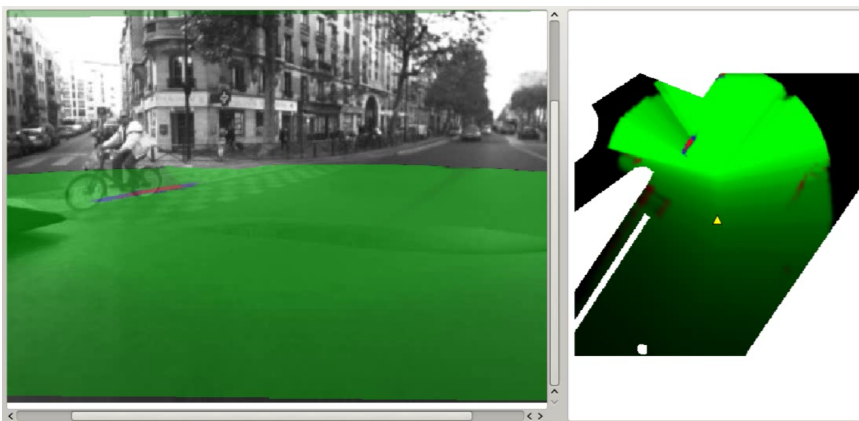


Figure 5.30.: Résultat de la construction de la CLD : détection du mouvement d'un cycliste

Sur l'exemple de la figure 5.31, on peut voir la bonne mise en correspondance de la contrainte de carte. Les deux terre-pleins cartographiés sont détectés statiques par

le télémètre. Les cellules rouges correspondantes sont quasiment toutes retirées par le filtrage de la carte car la pose et la carte sont ici de bonne qualité. Comme il reste quelques cellules rouges, on peut en déduire une erreur de l'ordre de 1 ou 2 cellules (environ 20 cm).

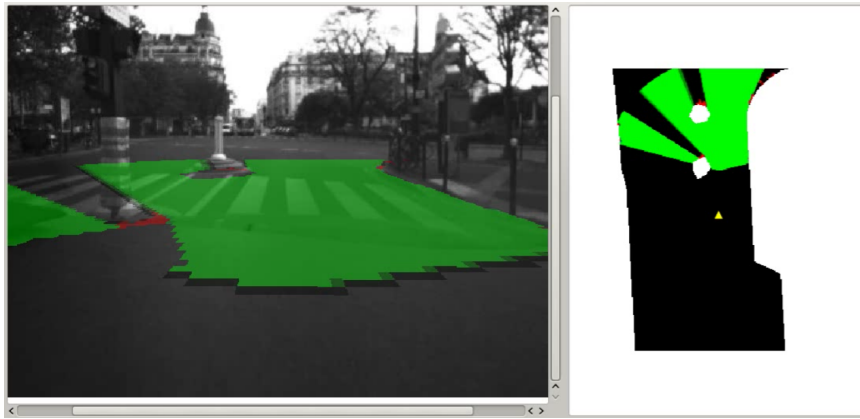


Figure 5.31.: Résultat de la construction de la CLD : la détection des terre-pleins centraux est quasiment retirée de la grille par la contrainte de carte.

Cependant, si l'estimation de la pose est de mauvaise qualité, la mise en correspondance entre la carte et la grille se dégrade. L'exemple de la figure 5.32 montre par exemple, ce qui se passe dans ce cas. Dans cette séquence réelle, la pose fournie par la centrale est biaisée de quelques mètres sur la gauche. Le bord gauche de la route dans la carte est donc décalé sur la droite à cause de cette erreur de positionnement. La détection des obstacles est toujours bonne car on peut remarquer que tous les piétons sont correctement détectés. Par contre, la zone navigable est coupée sur la gauche et étendue sur la droite (on ne peut pas le voir sur l'image).

Dans tout les exemples précédents, l'algorithme de détection du sol fonctionnait correctement car la surface était relativement plane et avec un angle de tangage modéré. On rappelle que les points des deux nappes inférieures qui touchent le sol ne sont pas introduits dans le modèle évidentiel d'obstacle. Lorsque les hypothèses de l'algorithme ne sont plus respectées les points du sol ne sont plus labellisés correctement et introduisent des cellules occupées au milieu de la route. Sur la figure 5.33, le ralentisseur ne respecte plus la condition de route plane. Le haut du dos d'âne est alors détecté comme un obstacle.

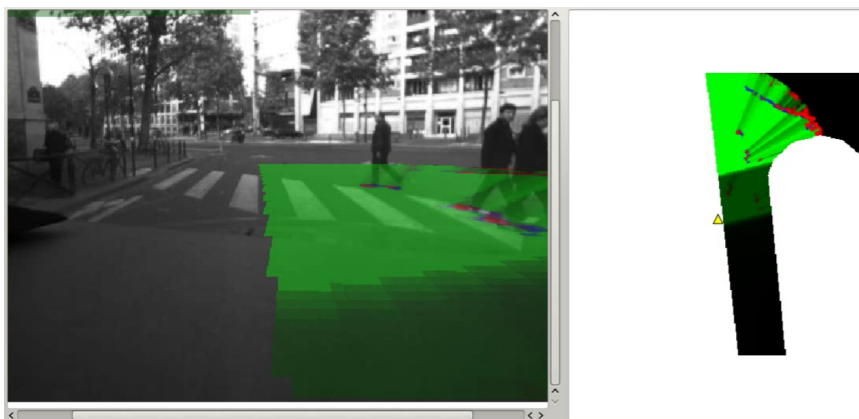


Figure 5.32.: Résultat de la construction de la CLD : effet d'une erreur de localisation

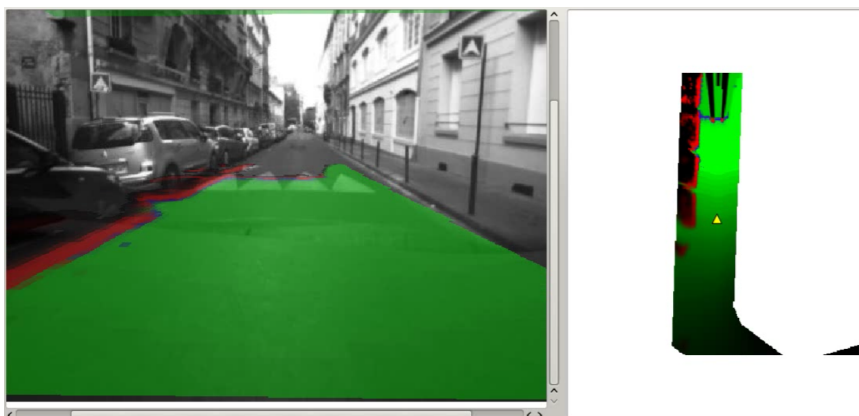


Figure 5.33.: Résultat de la construction de la CLD : effet d'un ralentisseur

5.4. Conclusion

Dans ce chapitre, nous avons présenté les principaux résultats expérimentaux de la thèse. Nous avons implémenté deux variantes (halo-centrée et égo-centrée) de l'algorithme de mise à jour de grilles de perception décrit dans le chapitre 2 et reposant sur les formalismes des fonctions de croyance et d'accumulation présentés dans le chapitre 3.

La méthode halo-centrée a été implémentée sous Matlab en utilisant les cadres de gestion d'incertitude d'accumulation et évidentiel. Les résultats de ces deux approches ont été comparés. L'apport du cadre évidentiel pour la détection d'objets en mouvement a été clairement démontré. La méthode détecte mieux les cellules en mouvement mais elle génère un peu plus de bruit du à la discrétisation spatiale et

aux erreurs de pose du véhicule.

La méthode égo-centrée a été implémentée en temps réel sur un véhicule prototype. La réalisation de nombreux tests en conditions réelles a permis d'évaluer les points faibles et les points forts de l'approche. Finalement, l'approche proposée a montré de bons résultats lorsque les conditions de fonctionnement sont conformes aux hypothèses. L'espace navigable est bien caractérisé en utilisant les contraintes d'espace libre et les contraintes de carte. Les objets mobiles sont bien détectés qu'ils soient petits (piétons) ou grand (bus). L'utilisation d'une fonction d'oubli introduit l'effet de rémanence recherché qui nous semble très important pour la navigation car la mémoire de perception est plus grande que le champ de vision du capteur.

Ayant eut accès aux cartes précises du projet CityVIP, nous nous sommes intéressés à l'injection d'informations cartographiques pour améliorer la perception dans les grilles. Nous avons proposé de projeter les cartes sur une grille métrique et d'utiliser cette grille soit, pour réaliser un filtrage de la grille de perception, soit pour la fusionner avec celle-ci. Nous montrons que cela permet de mieux caractériser les éléments de la grille.

Cette implémentation a conduit à l'amélioration de la plate-forme Pacpus et à la mise en œuvre de plusieurs bibliothèques de traitements et de calculs utilisées en robotique. De plus, la construction modulaire de cette application permettra sa réutilisation au sein de travaux futurs.

6. Conclusion et perspectives

Sommaire

6.1. Conclusion générale	151
6.2. Perspectives	153

6.1. Conclusion générale

Dans ce manuscrit, nous avons étudié plusieurs méthodes utilisant des grilles pour la perception de véhicules en milieu urbain. En se basant sur le formalisme des grilles d'occupation 2D pour décrire l'environnement, nous avons présenté et formalisé une méthode utilisant une architecture de type fusion de données pour mettre à jour de façon dynamique une grille de l'environnement. Nous avons également étudié différents cadres méthodologiques pour gérer les incertitudes dans les grilles d'occupation, à savoir l'accumulation spatiale, le filtrage bayésien et les fonctions de croyance. Ce dernier cadre évidentiel a été particulièrement étudié et formalisé pour la perception robotique. Nous avons mis en œuvre et testé cette approche sur des données réelles enregistrées durant le projet CityVIP à Paris. Enfin, nous l'avons intégrée à bord d'un véhicule expérimental du laboratoire Heudiasyc.

Les contributions principales de ce travail sont les suivantes.

Traitements sur les grilles d'occupation

Nous avons choisi d'utiliser une représentation de l'environnement avec des grilles. Cette représentation a l'avantage de représenter finement tout type d'environnement sans faire d'hypothèse à-priori, contrairement aux approches basées primitives. La fidélité (en anglais *precision*) est seulement limitée par la résolution de la grille. Ce type de représentation est cependant assez difficile à manipuler. En considérant le parallèle avec les images matricielles (comme Elfes l'avait d'ailleurs déjà proposé à la fin de [Elfes, 1989c]) et les avancées importantes dans le domaine du traitement numérique d'image (comme le parallélisme de calculs des nouvelles architectures matérielles), il nous a semblé important de développer le concept de grille métrique à partir de l'image matricielle, en ajoutant une notion métrique (spatiale) à la grille de l'image.

Intérêt du cadre évidentiel dans les grilles d'occupation

Le formalisme des fonctions de croyance est relativement récent et de plus en plus populaire en fusion de données. Ce formalisme est encore peu utilisé dans les traitements bas niveau de la fusion multi-capteurs car il est souvent considéré comme trop lourd pour les applications embarquées temps réel du fait de sa complexité calculatoire. Nous avons proposé d'utiliser ici ce formalisme avec un cadre de discernement simple et une approche reposant sur une modélisation inverse des capteurs extéroceptifs et une mise à jour de la grille par fusion. Ceci permet de limiter la complexité tout en apportant les avantages du cadre évidentiel, comme la différenciation entre les incertitudes d'ignorance et les incertitudes de conflit.

Ce formalisme est en outre très avantageux dans notre approche car il possède :

- des outils puissants pour la prise de décision qui permettent notamment de mesurer la quantité d'information contenue dans la grille de perception,
- différentes règles de combinaison qui correspondent à différents types de fusion et qui sont simples à mettre en œuvre,
- des mécanismes intéressants et faciles à mettre en place tels que l'affaiblissement ou les opérations de raffinement et de grossissement.

Nous avons comparé plusieurs opérateurs de fusion évidentielle et les avons aussi comparé à l'approche bayésienne, notamment quand la scène est dynamique.

Proposition de modèles de capteur lidar

Dans ces travaux, le lidar multi-écho et multi-nappe a été le capteur de perception principal. Après avoir étudié finement le fonctionnement de ce capteur et analysé beaucoup de données, nous avons proposé un modèle de capteur qui utilise une grille polaire faisant un compromis entre les contraintes de l'embarqué et la fidélité du modèle avec les mesures. Afin d'exploiter au maximum l'information fournie par ce capteur, il nous a semblé primordial que ce modèle de capteur prenne en compte la détection du sol et les masquages. Nous avons proposé un deuxième modèle plus raffiné prenant en compte les incertitudes de position des échos. Ce modèle qui repose sur la fusion de tous les échos n'est à l'heure actuelle pas compatible avec les contraintes temps-réel.

Proposition d'une architecture fusion modulaire multi-grille

Nous avons proposé une architecture de mise à jour par fusion. Elle est suffisamment générique pour être facilement implémentée dans différents repères et avec différents formalismes. L'approche repose sur la construction d'un ensemble de cartes (locales et dynamiques). Chaque carte est construite à partir d'une source précise et représente une contrainte particulière. Ainsi, chaque source d'information est fusionnée au sein d'une carte globale regroupant l'intersection de toutes ces contraintes. Cette approche est très modulaire ce qui favorise la mise en place de systèmes multi-capteurs

et aussi l'ajout d'information en utilisant d'autres sources comme de l'information géographique.

Implémentation et expérimentations

La mise en œuvre de l'approche proposée en utilisant un lidar a permis de l'évaluer qualitativement en vue de son utilisation au sein de systèmes d'aide à la conduite ou de robots autonomes. De plus, l'implémentation des travaux sur un démonstrateur temps-réel a permis de tester le système dans de très diverses conditions. Durant ces tests, nous avons pu vérifier grâce à la procédure de validation visuelle que nous avons proposée, que les résultats étaient très cohérents avec la réalité et que cette approche était compatible avec les contraintes embarquées. Nous avons également pu évaluer les limites de l'approche par des tests s'éloignant des conditions nominales de fonctionnement. Ces travaux d'implémentation ont permis plusieurs améliorations sur la plate-forme Pacpus.

6.2. Perspectives

On peut dresser plusieurs perspectives à ces travaux.

Validation

Les résultats présentés dans cette thèse sont encourageants, mais ils n'ont été évalués que de façon qualitative. Bien que la cohérence entre la grille de perception et l'environnement réel semble bonne au travers de l'image de la caméra, nous n'avons pas de résultat chiffré. Une validation quantitative du système permettrait de fournir ce type d'indicateur pour comparer notre approche avec d'autres. Pour cela l'utilisation de données réelles accompagnées d'une vérité terrain est à envisager. La comparaison entre les résultats de l'algorithme proposé et la vérité terrain nécessite néanmoins de trouver des métriques de mesure pertinentes. D'autre part, la réalisation d'un jeu de données de vérité terrain est complexe à réaliser car elle concerne simultanément plusieurs éléments (piéton, véhicule, etc...) et exige une précision spatiale et de synchronisation très importante. On peut envisager faire une validation sur simulateur dans un premier temps.

Prise en compte des incertitudes de localisation

Nous faisons l'hypothèse est que la pose du robot est connue avec une précision élevée devant la résolution de la grille. Or, cette hypothèse n'est pas vérifiée la plupart du temps. Cela a plusieurs conséquences, comme la génération de conflit sur les objets fixes due à une mauvaise cohérence durant la fusion de la grille. Dans un premier

temps, on peut utiliser la perception pour améliorer la localisation comme c'est le cas dans les systèmes de type SLAM qui exploitent explicitement la perception de l'environnement. Cela permettrait de ne pas recourir à l'utilisation d'un système de positionnement onéreux et améliorerait certainement le résultat de la fusion. On peut d'autre part, prendre en compte une connaissance de l'incertitude de localisation dans le modèle de fusion. Il faut pour cela trouver une représentation pertinente de l'incertitude spatiale dans le modèle de grille.

Dynamique de la scène

Dans l'approche présentée, nous utilisons un affaiblissement dans la grille pour prendre en compte la dynamique de la scène. L'utilisation de cet affaiblissement et d'opérateurs de fusion réactifs avec peu de latence permet de détecter correctement les cellules contenant des objets mobiles et de les mettre en évidence. En l'état actuel, ces cellules ne sont ni propagées ni segmentées. La segmentation des cellules mobiles est une perspective intéressante et facilement réalisable compte tenu des résultats de détection des objets mobiles. La propagation spatiale de ces cellules permettrait d'améliorer le système de perception, puisque le conflit pourrait être mis à profit pour corriger une erreur de prédiction. Cette propagation pourrait se faire soit au niveau de la grille soit au niveau d'une représentation avec des primitives associées à la grille.

Utilisation des cartes

Au cours de cette thèse, nous avons initié des travaux autour de l'injection d'information géographique a priori dans les grilles de perception. Ces travaux ont abouti à la proposition de deux méthodes : une méthode par filtrage et une méthode par fusion. Ces approches ne considèrent pas en l'état les erreurs de la carte et de la pose du véhicule. La prise en compte de ces incertitudes est nécessaire pour une utilisation intègre des cartes dans la perception. Dans l'approche par fusion, nous avons proposé un mécanisme de raffinement de l'état O du cadre de discernement qui pourrait aussi s'étendre à l'état F . Enfin, nous avons, pour l'instant, utilisé seulement les cartes du projet CityVIP. Une perspective pourrait être d'utiliser des cartes libres d'accès telles que les cartes OpenStreetMap ou GoogleMap qui sont moins précises mais qui couvrent une grande partie du territoire.

Implémentation multi-capteurs

Lors des expérimentations, nous avons constaté qu'un capteur unique limite la perception. L'utilisation de capteurs extéroceptifs supplémentaires serait bénéfique tant au niveau de l'intégrité du système (en offrant des redondances) qu'au niveau de la complémentarité (en couvrant un champ différent ou en utilisant une autre modalité

de perception). Comme l'architecture de fusion proposée permet d'ajouter de nouvelles sources, on peut facilement ajouter une nouvelle modalité de perception après avoir développé un modèle de capteur sur grille évidentielle adaptée. Par exemple, dans le cadre du projet de recherche collaborative PRETIV, Philippe Xu étudie un système perception utilisant un système stéréo-vision et un lidar. Il a développé un algorithme pour fusionner ces différentes sources dans le plan image assez proche des travaux de cette thèse.

Utilisation sur un véhicule autonome

Bien que ces travaux soient destinés à être utilisés dans des applications de robotique en milieu très contraint, l'approche présentée n'a pas encore été testée à bord d'un véhicule complètement autonome. Une perspective logique serait donc d'implémenter cette approche dans le système de navigation d'un véhicule. Un ensemble de travaux en cours au laboratoire Heudiasyc vise à utiliser ces travaux dans divers projets. Dans un premier temps, nous envisageons d'utiliser cette approche pour implémenter dans un système de navigation d'un robot *wifibot* [Wif, 2003]. Dans un second temps, ces travaux pourront être utilisés pour la navigation de véhicules automobiles automatisés du laboratoire comme base de calcul à des stratégies de planification de trajectoire à court terme.

A. Opérations sur les grilles

Nous avons vu comment utiliser les grilles pour estimer une fonction définie sur un domaine spatial 2D. Dans cette partie nous allons montrer comment, on peut appliquer les transformations nécessaires pour propager l'information contenue dans les grilles. Comme le montre la figure A.1, ces opérations peuvent être regroupées en deux catégories : les opérations mono-grilles, qui sont de simples transformations de grille, ou les transformations multi-grilles, qui regroupent les opérations de combinaison de grilles.

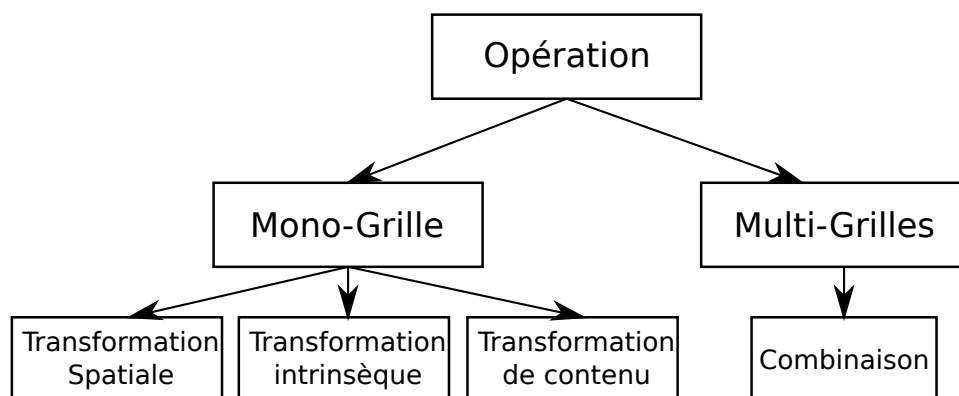


Figure A.1.: Classification d'opérations classiques sur les grilles

A.1. Transformation spatiale

A.1.1. Principe du traitement de l'antécédent

On définit une transformation géométrique de grille comme une transformation dans laquelle on modifie la position des informations contenues dans la grille (cf. figure A.2).

Dans ces transformations, on calcule une correspondance fonctionnelle entre deux coordonnées, les coordonnées d'entrée et de sortie. Les coordonnées d'entrée sont les coordonnées des centres de la grille d'entrée C^{In} , qui supportent l'information dans les cellules. Le résultat de la transformation géométrique appliquée à la grille forme les coordonnées de sortie. Comme les coordonnées de sortie C^{Out} ne correspondent pas à la grille de sortie (voir figure A.3), on utilise les fonctions d'interpolation

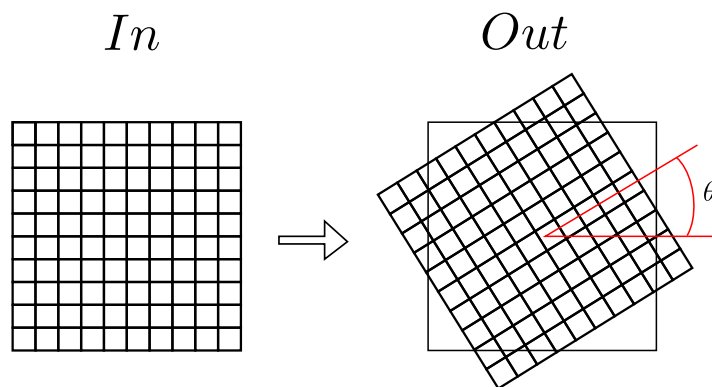


Figure A.2.: Exemple de transformation géométrique (ici une rotation d'angle θ).

détaillées par la suite. Le contenu des cellules est ainsi modifié par l'application d'une transformation spatiale.

En pratique, pour être sûr que toutes les cellules de la grille de sortie soient bien affectées, on réalise l'opération inverse (voir figure A.3). On balaye les centres de la grille d'arrivée et on calcule les coordonnées correspondantes dans la grille de départ en utilisant la fonction inverse. Pour évaluer la valeur associée à chaque cellule (qui on le rappelle est rattachée au centre), nous allons réaliser une interpolation d'ordre variable entre les valeurs des C_{ij}^{Out} après transformation et les valeurs des C_{ij}^{In} avant transformation. Si le contenu est un vecteur, toutes les composantes sont interpolées et ce, de manière indépendante entre elles.

A.1.2. Interpolation multivariée

L'interpolation multivariée est utilisée pour calculer la valeur d'une fonction en un point donné connaissant la valeur de cette fonction en un ensemble d'autres points du voisinage [Larsson et Fornberg, 2005].

C'est le cas lorsqu'on applique une transformation spatiale sur la grille qui ne respecte pas la discrétisation de la grille. Dans ce cas, les centres $C_{ij} = \begin{bmatrix} x \\ y \end{bmatrix}$ des cellules transformées dans le repère \mathcal{R}_M ne respectent plus la condition d'alignement (équation A.1) :

$$\begin{aligned} x_{C_{ij}} &= {}^0x - i \cdot \Delta_u \\ y_{C_{ij}} &= {}^0y - j \cdot \Delta_v \end{aligned} \quad (\text{A.1})$$

L'interpolation d'ordre 0 consiste en une approximation par plus proche voisin (*Nearest Neighbour interpolation*). Soit une grille qui subit une transformation géomé-

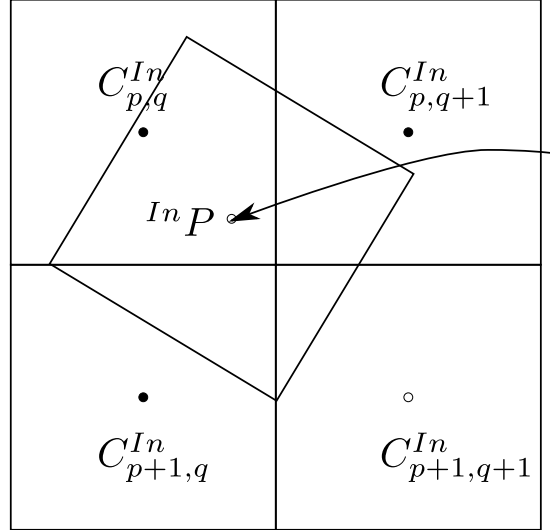


Figure A.3.: Interpolation d'une cellule : le centre de la cellule transformée n'est pas aligné sur la grille. ^{In}P est le vecteur de coordonnées de l'antécédent de C_{ij}^{Out} . Elles sont réelles et exprimées dans le repère de la grille G^{In} . Les points C^{In} sont les centres des cellules adjacentes. Leurs coordonnées sont indicées de façon discrète dans le repère de la grille G^{In} .

trique f , l'algorithme¹ A.1 permet d'évaluer la valeur d'une cellule après transformation. La fonction prend en paramètres d'entrée la grille G^{In} soit $(I^{In}, C^{In}, \mathcal{D}^{In})$, ainsi que le point ^{In}P . ^{In}P correspond à l'antécédent du centre d'une cellule de la grille G^{Out} . Elle retourne alors la valeur de la cellule de sortie.

Algorithme A.1 $I_{ij}^{Out} \leftarrow \text{Interpolation_NN}(I^{In}, C^{In}, \mathcal{D}^{In}, ^{In}P)$

```

if  $^{In}P \in \mathcal{D}^{In}$  then
     $\begin{bmatrix} p \\ q \end{bmatrix} = \text{round\_cell}(^{In}P, C^{In})$ 
     $I_{ij}^{Out} \leftarrow I_{pq}^{In}$ 
else
     $I_{ij}^{Out} \leftarrow \text{Default\_value}$ 
end if

```

L'interpolation d'ordre 0 est simple à réaliser mais ne prend pas en compte la position du point à l'intérieur de la cellule ce qui induit des approximations importantes. Pour améliorer les résultats, on peut augmenter l'ordre d'interpolation.

De même que pour l'interpolation d'ordre 0, l'interpolation d'ordre 1 permet de calculer la valeur d'une cellule qui ne correspond pas au maillage. Elle consiste

1. La fonction `round_cell` renvoie les indices de la cellule dont le centre C^{In} est le plus proche du point P^{In}

cependant en une interpolation bi-linéaire (*bi-linear interpolation*). On utilise le modèle de l'équation A.2 pour réaliser cette interpolation. Pour chaque point interpolé, les 4 points adjacents permettent de déterminer les différents α_{pq} .

$$I(x, y) = \sum_{p=0}^1 \sum_{q=0}^1 \alpha_{pq} x^p y^q \quad (\text{A.2})$$

Comme décrit dans l'algorithme² A.2, l'interpolation s'effectue, en pratique, comme deux interpolations linéaires successives.

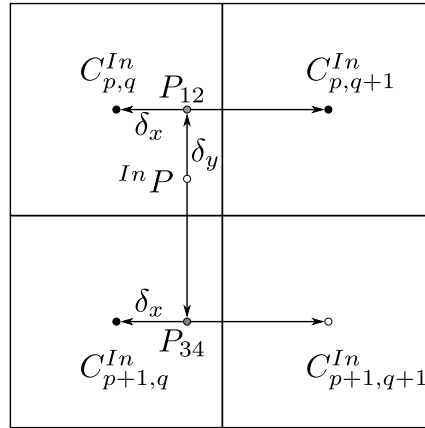


Figure A.4.: Interpolation bilinéaire de la valeur d'une cellule

Algorithme A.2 $I_{ij}^{Out} \leftarrow \text{Interpolation_BL}(I^{In}, C^{In}, \mathcal{D}^{In}, {}^{In}P)$

```

if ( ${}^{In}P \in \mathcal{D}^{In}$ ) then
   $\begin{bmatrix} p \\ q \end{bmatrix} = \text{floor\_cell}({}^{In}P, C^{In})$ 
   $\delta_x = \frac{{}^{In}P.x - C_{pq}^{In}.x}{C_{p+1,q}^{In}.x - C_{pq}^{In}.x}$ 
   $\delta_y = \frac{{}^{In}P.y - C_{pq}^{In}.y}{C_{p,q+1}^{In}.y - C_{pq}^{In}.y}$ 
   $I_{12} = (1 - \delta_x) \cdot I_{pq}^{In} + \delta_x \cdot I_{p+1,q}^{In}$ 
   $I_{34} = (1 - \delta_x) \cdot I_{pq+1}^{In} + \delta_x \cdot I_{p+1,q+1}^{In}$ 
   $I_{ij}^{Out} = (1 - \delta_y) \cdot I_{12} + \delta_y \cdot I_{34}$ 
else
   $I_{ij}^{Out} \leftarrow \text{Default\_value}$ 
end if

```

2. La fonction floor_cell renvoie les indices de la cellule dont le centre C^{In} est le plus proche du point P^{In} arrondi vers le bas

A.1.3. Erreurs d'interpolation successives

Dans certain cas, la même grille subit plusieurs interpolations successives (cas d'un processus séquentiel par exemple). Or, à chaque interpolation, on commet une erreur d'approximation sur la grille.

Pour mettre en avant ce phénomène, on peut considérer l'exemple suivant. Soit le signal 1D discret $S_i(x)$ défini sur la figure A.5a. Ce signal est échantillonné toutes les unités suivant x .

On réalise une translation de ce signal pour obtenir le signal suivant $S_f(x) = S_i(x - 10)$, cf. figure A.5b.

Cependant, la translation est réalisée en une succession de translations élémentaires de pas $dx = 0.5$. Le processus est décrit par l'algorithme A.3.

Algorithme A.3 $S_t(x) \leftarrow$ Translation ($S(x), p$)

```
 $S_0(x) = S_i(x)$   
for  $i = 1 \rightarrow 20$  do  
   $S_i = \text{interpolate}(S_{i-1}(x - 0.5))$   
end for  
 $S_t(x) = S_{20}(x)$ 
```

Si on utilise une interpolation de type plus proche voisin, le signal final S_t n'est pas déformé, mais il n'est pas positionné au bon endroit à cause des erreurs d'arrondi qui à chaque itération vont légèrement décaler le résultat. Ceci est visible sur l'exemple de la figure A.5c.

Si on utilise une interpolation de type linéaire, le signal final est bien localisé en position mais il est étalé comme le montre la figure A.5d. Ce phénomène est comparable à un filtrage de type passe bas : il est visible aux endroits où il y a des gradients importants et a tendance à flouter la grille. La figure A.5e montre l'erreur commise qui crée un effet de flou sur la grille.

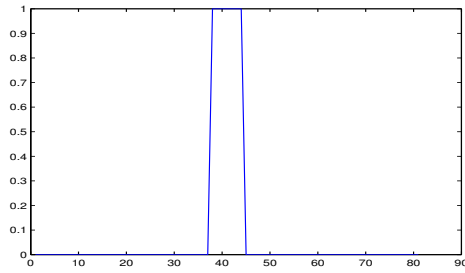
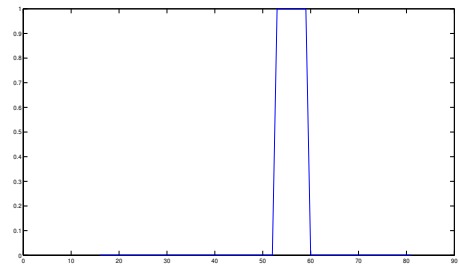
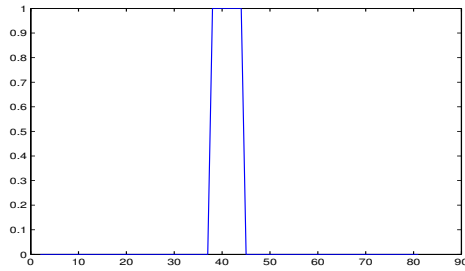
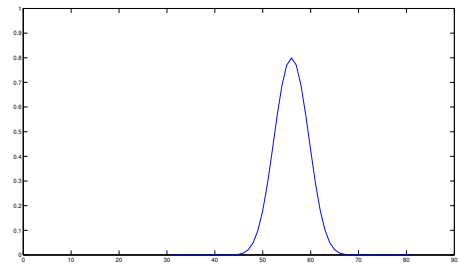
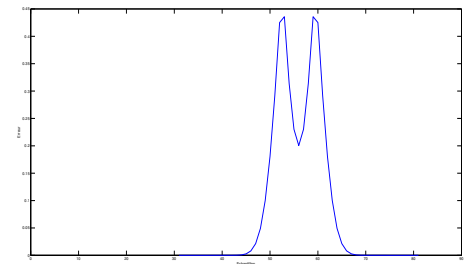
(a) Signal initial : $S_i(x)$, x est l'abscisse(b) Signal final désiré : $S_f(x)$ (c) Résultat du processus utilisant une interpolation plus proche voisin, le signal obtenu S_t est décalé par rapport S_t (d) Résultat du processus utilisant une interpolation linéaire, le signal obtenu S_t est très différent du signal attendu S_t .(e) Différence entre le résultat attendu et le résultat obtenu en utilisant l'interpolation linéaire $S_f(x) - S_t(x)$

Figure A.5.: Approximations dues à l'interpolation

A.1.4. Ré-échantillonnage de grille

L'opération de ré-échantillonnage de grille redéfinit complètement un nouveau maillage pour la grille avec des pas de discrétisation différents $\Delta^{Out} = (\Delta_u^{Out}, \Delta_v^{Out})$. Si le nouveau pas d'échantillonnage est plus grand que le précédent, on parle de sous-échantillonnage et il y a perte d'une partie de l'information. Dans le cas contraire, on parle de sur-échantillonnage. Ce sur-échantillonnage n'ajoute pas d'information, mais il augmente la limite de Shannon pour des traitements ultérieurs (la grille pourra contenir des informations avec une fréquence spatiale supérieure).

Enfin, de manière générale, la dimension de la grille ré-échantillonnée est modifiée

pour correspondre à un multiple du nouveau pas d'échantillonnage (nombre entier de cellules).

On dispose des points $C_{ij}^{In} = \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix}$ qui vérifient

$$\begin{aligned} x_{ij} &= {}^0x - i \cdot \Delta_u^{In} \\ y_{ij} &= {}^0y - j \cdot \Delta_v^{In} \end{aligned} \quad (\text{A.3})$$

On définit un nouveau maillage $C_{ij}^{Out} = \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix}$:

$$\begin{aligned} x_{ij} &= {}^0x - i \cdot \Delta_u^{Out} \\ y_{ij} &= {}^0y - j \cdot \Delta_v^{Out} \end{aligned} \quad (\text{A.4})$$

Pour calculer les valeurs I_{ij}^{Out} à partir des valeurs I_{ij}^{In} et des centres, on utilise une fonction, qui peut être une interpolation ou un filtre médian, par exemple.

A.1.5. Transformation solide

Il s'agit d'appliquer une translation et/ou une rotation à une grille. Soit \mathcal{R}_{In} et \mathcal{R}_{Out} deux repères distincts de \mathbb{E}^2 . Comme défini dans la section 2.2.2, on définit ${}^{Out}H_{In}$ la matrice de transformation homogène associée à la transformation de \mathcal{R}_{In} à \mathcal{R}_{Out} .

$${}^{Out}H_{In} = \begin{bmatrix} r_{xx} & r_{xy} & t_x \\ r_{yx} & r_{yy} & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.5})$$

On rappelle que la transformation ponctuelle s'effectue de la façon suivante :

$$\forall P \in \mathbb{E}^2, {}^{Out}P = {}^{Out}H_{In} \cdot {}^{In}P \quad (\text{A.6})$$

Comme le décrit l'algorithme A.4, la transformation est appliquée à chaque cellule de la grille indépendamment. Pour chaque cellule de la nouvelle grille, on calcule les coordonnées du centre de la cellule par la transformation inverse et on calcule ensuite la valeur du contenu en utilisant les fonctions d'interpolation définies dans la section A.1.2.

A.2. Transformation intrinsèque

Ces transformations modifient les caractéristiques intrinsèques des grilles, telles que le repère, la résolution ou le domaine.

Algorithme A.4 $I^{Out} \leftarrow \text{Transformation_Solide}(I^{In}, C^{In}, \mathcal{D}^{In}, C^{Out}, {}^{Out}H_{In})$

```

1: compute  ${}^{Out}H_{In}^{-1}$ 
2: for all  $i, j \mid G_{ij}^{Out} \in G^{Out}$  do
3:    $\begin{bmatrix} x_{Out} \\ y_{Out} \\ 1 \end{bmatrix} \leftarrow C_{ij}^{Out}$ 
4:    $\begin{bmatrix} x_{In} \\ y_{In} \\ 1 \end{bmatrix} = {}^{Out}H_{In}^{-1} \cdot \begin{bmatrix} x_{Out} \\ y_{Out} \\ 1 \end{bmatrix}$ 
5:    $I_{ij}^{Out} \leftarrow \text{Interpolation\_BL} \left( I^{In}, C^{In}, \mathcal{D}^{In}, \begin{bmatrix} x_{In} \\ y_{In} \\ 1 \end{bmatrix} \right)$ 
6: end for

```

A.2.1. Changement de repère

Le changement de repère de grille consiste simplement à définir un nouveau repère de grille \mathcal{R}_2 . Pour que la grille soit géométriquement compatible avec \mathcal{R}_2 , il faut que les axes de celui-ci soient colinéaires aux axes de \mathcal{R}_1 . On limitera donc le changement de repère à une translation pure.

Dans ce cas, le changement de repère consiste à calculer les coordonnées du point $I = \begin{bmatrix} {}^0x & {}^0y & 1 \end{bmatrix}^T$ dans le nouveau repère, le contenu des cellules n'est pas modifié.

A.2.2. Transformation polaire / cartésienne

On peut définir la transformation d'une grille d'un repère polaire en un repère cartésien comme un *warping* défini par les relations suivantes :

$$\forall \mathbf{P} P = \begin{bmatrix} \rho \\ \theta \end{bmatrix}, \mathbf{C}_P = \mathcal{W}(\mathbf{P} P) \quad (\text{A.7})$$

Ce *warping* est explicité dans l'équation suivante :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \rho \cdot \cos \theta \\ \rho \cdot \sin \theta \end{bmatrix}. \quad (\text{A.8})$$

De la même façon que pour la transformation solide, on calcule dans un premier temps les correspondances de coordonnées et on calcule alors les valeurs des cellules en utilisant les fonctions d'interpolation.

Algorithme A.5 $I_{ij}^{Cart} \leftarrow \text{Pol2Cart}(I^{Pol}, C^{Pol}, \mathcal{D}^{Pol}, C_{ij}^{Cart})$

```

1: for all  $i,j \mid G_{ij}^{Cart} \in G^{Cart}$  do
2:    $\begin{bmatrix} x \\ y \end{bmatrix} = C^{Cart}$ 
3:    $\begin{bmatrix} \rho \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \text{atan}\left(\frac{y}{x}\right) \end{bmatrix}$ 
4:    $I_{ij}^{Cart} \leftarrow \text{Interpolation\_BL}\left(I^{Pol}, C^{Pol}, \mathcal{D}^{Pol}, \begin{bmatrix} \rho \\ \theta \end{bmatrix}\right)$ 
5: end for

```

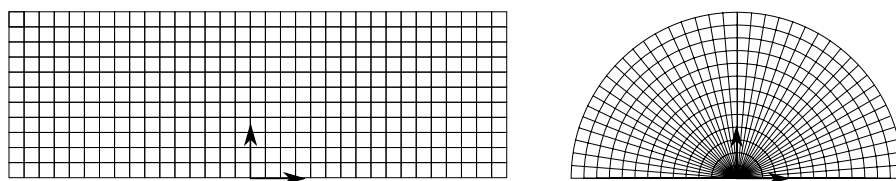


Figure A.6.: Représentation de la grille polaire, à gauche dans son repère polaire et à droite dans un repère cartésien après avoir réalisé la transformation.

La figure A.6 montre la grille polaire avant et après transformation.

On se limite ici à présenter la transformation polaire \rightarrow cartésien car c'est celle dont nous aurons besoin par la suite avec le capteur lidar.

A.2.3. Agrandissement et troncature

Les opérations d'agrandissement et de troncature correspondent à un changement de domaine pour la grille. Cela a pour effet d'ajouter ou de retirer un certain nombre de lignes et/ou de colonnes dans la grille. Dans le cas d'un agrandissement, les valeurs des cellules ajoutées sont fixées à une valeur par défaut qui en général correspond à l'ignorance.

A.3. Transformation de contenu

Les transformations de contenu sont les transformations dans lesquelles on modifie directement la valeur contenue dans la cellule. Ces opérations regroupent par exemple les filtres appliqués à l'image. On présente ci-après la seule opération dont nous aurons besoin dans la suite de ce manuscrit.

A.3.1. Normalisation

Dans certain cas, à cause des approximations de calcul numérique, des conditions de normalisation du contenu des cellules ne sont plus forcément vérifiées. Un étape de normalisation A.9 comme suit est alors utilisée.

$$I_j^i(k) = \frac{I_j^i(k)}{\sum_l I_j^i(l)}, \forall k \in [1 n] \quad (\text{A.9})$$

A.4. Combinaison

Les opérations de combinaison de grilles comprennent toutes les opérations qui utilisent plusieurs grilles en entrée. Généralement, ces opérations nécessitent que les grilles d'entrée soient cohérentes spatialement, c'est-à-dire que chaque cellule d'une grille correspond exactement à une autre cellule dans l'autre grille. Pour cela, il faut que les maillages des deux grilles soient identiques (même résolution, même domaine).

Ainsi, on réalise au préalable un ensemble de transformations (géométriques ou intrinsèques) sur les grilles pour les rendre compatibles.

A.4.1. Fusion statique de deux grilles

La fusion de grilles consiste à combiner spatialement l'information des deux grilles distinctes pour créer une nouvelle grille. On peut combiner les grilles soit pour augmenter la quantité d'information contenue dans la grille fusionnée, soit pour augmenter l'intégrité de la grille fusionnée.

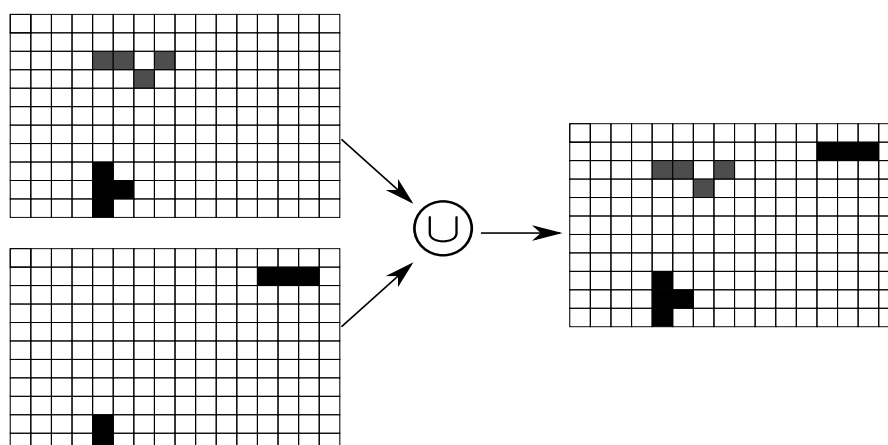
Dans l'approche proposée ici, on combine l'information de chaque cellule. Il faut en plus des conditions précédentes que chaque cellule de la première grille corresponde spatialement à une cellule de la seconde grille.

Un opérateur de fusion de grille revient alors à une fusion de cellule à cellule et consiste à appliquer cette fusion sur chaque paire de cellules de manière indépendante.

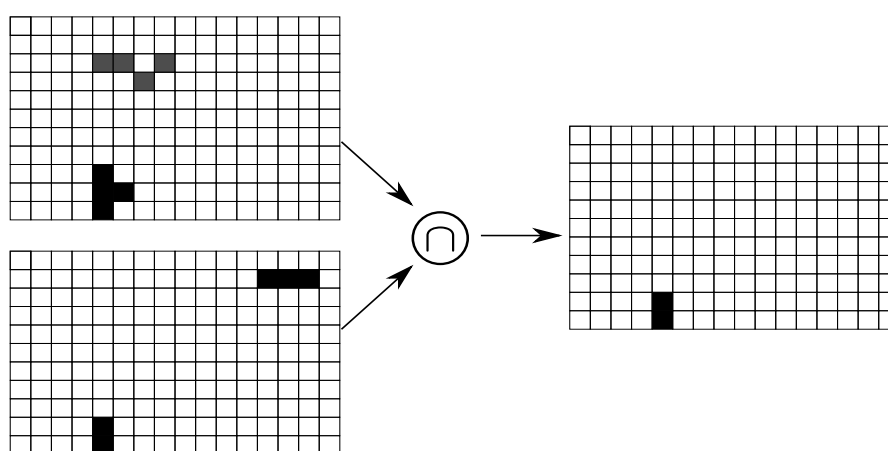
La figure A.7 illustre des mécanismes de fusion élémentaires.

A.4.2. Inter-corrélation

De façon identique à l'inter-corrélation d'images, l'inter-corrélation de grilles est une fonction qui mesure la similarité entre les grilles, dans le but de mesurer, par exemple, un déplacement ou de reconnaître un élément particulier, [Konolige et Chou, 1999].



(a) Fusion disjonctive de 2 grilles.



(b) Fusion conjonctive de 2 grilles.

Figure A.7.: Fusion cellule à cellule de deux grilles. Les cellules noires correspondent à une information de valeur maximum

Considérons, par exemple, que l'on a deux grilles G^1 et G^2 séparées par une translation.

Le principe de la corrélation est le suivant :

- on fait subir un décalage d'un certain nombre de cellules (p, q) à une des 2 grilles,
- on fait le produit 2 à 2 entre les deux grilles (on fait le produit cellule à cellule sur tout les cellules qui coïncident),
- on somme tous ces produits.

Il existe plusieurs algorithmes pour calculer la corrélation, les principaux sont détaillés dans le tableaux A.1. Certains travaillent avec des valeurs centrées et/ou moyennées, cf. [Sutton *et al.*, 2009]. La liste ci-dessous présente les formules de calcul des principaux algorithmes. On préfère souvent utiliser la formule *ZNCC* (Zero mean Normalized Cross-Correlation) qui retourne une valeur dans l'intervalle $[-1, 1]$ indépendante de la moyenne de la grille.

Type de corrélation	Acronyme	Formule
Corrélation croisée	CC	$\sum_{ij} I_{ij}^1 \cdot I_{i+p, j+q}^2$
Corrélation croisée centrée	ZCC	$\sum_{ij} (I_{ij}^1 - \bar{I}^1) \cdot (I_{i+p, j+q}^2 - \bar{I}^2)$
Corrélation croisée normalisée	NCC	$\frac{\sum_{ij} I_{ij}^1 \cdot I_{i+p, j+q}^2}{\sqrt{\sum_{ij} I_{ij}^1 ^2 + \sum_{ij} I_{i+p, j+q}^2 ^2}}$
Corrélation croisée centrée et normalisée	$ZNCC$	$\frac{\sum_{ij} (I_{ij}^1 - \bar{I}^1) \cdot (I_{i+p, j+q}^2 - \bar{I}^2)}{\sqrt{\sum_{ij} I_{ij}^1 - \bar{I}^1 ^2 + \sum_{ij} I_{i+p, j+q}^2 - \bar{I}^2 ^2}}$

Table A.1.: Récapitulatif des principaux algorithmes de corrélation utilisés

On peut étendre ce principe à d'autres transformations que la translation mais cela est plus complexe, le coup calculatoire devient très important et la précision diminue à cause des interpolations.

B. Affaiblissement contextuel

Nous démontrons ici l'utilisation de l'affaiblissement contextuel pour créer une fonction de masse de masquage.

L'affaiblissement contextuel est proposée par Mercier dans [Mercier *et al.*, 2005], il permet de transférer une part de croyance d'un état sur l'ignorance. Il est utilisé pour traduire le manque de fiabilité dans une source. Il se formalise comme suit.

Soit considère une fonction de masse m définit sur le cadre de discernement Ω . on souhaite appliquer un affaiblissement contextuel à la fonction de masse m . L'affaiblissement contextuel se base sur la création d'une fonction de masse définie de la façon suivante :

$$\mathbf{m}_\alpha = \begin{bmatrix} \emptyset & F & O & \Omega \\ 1 - \sum_i \alpha_i & \alpha_f & \alpha_o & \alpha_{of} \end{bmatrix}$$

Cette fonction de masse est alors fusionner de façon disjonctive avec la fonction de masse initiale pour appliquer l'affaiblissement.

$$m = m \odot m_\alpha$$

L'affaiblissement contextuel appliqué dépend des différents coefficients α . En construisant une fonction de masse m_α particulière, on peut réaliser un affaiblissement différents pour chaque état. Si on place une masse non nulle sur α_{of} , on affaiblit les deux états O et F de la même manière. Si on place une masse non nulle α_f , on affaiblit l'état opposé O et, pour affaiblir l'état F , on place la masse sur α_o .

Nous rappelons ici le problème posé dans le chapitre 4, nous considérons n distributions de fonctions de masse m_k qui correspondent chacune à un écho laser (dans l'exemple, $n = 2$).

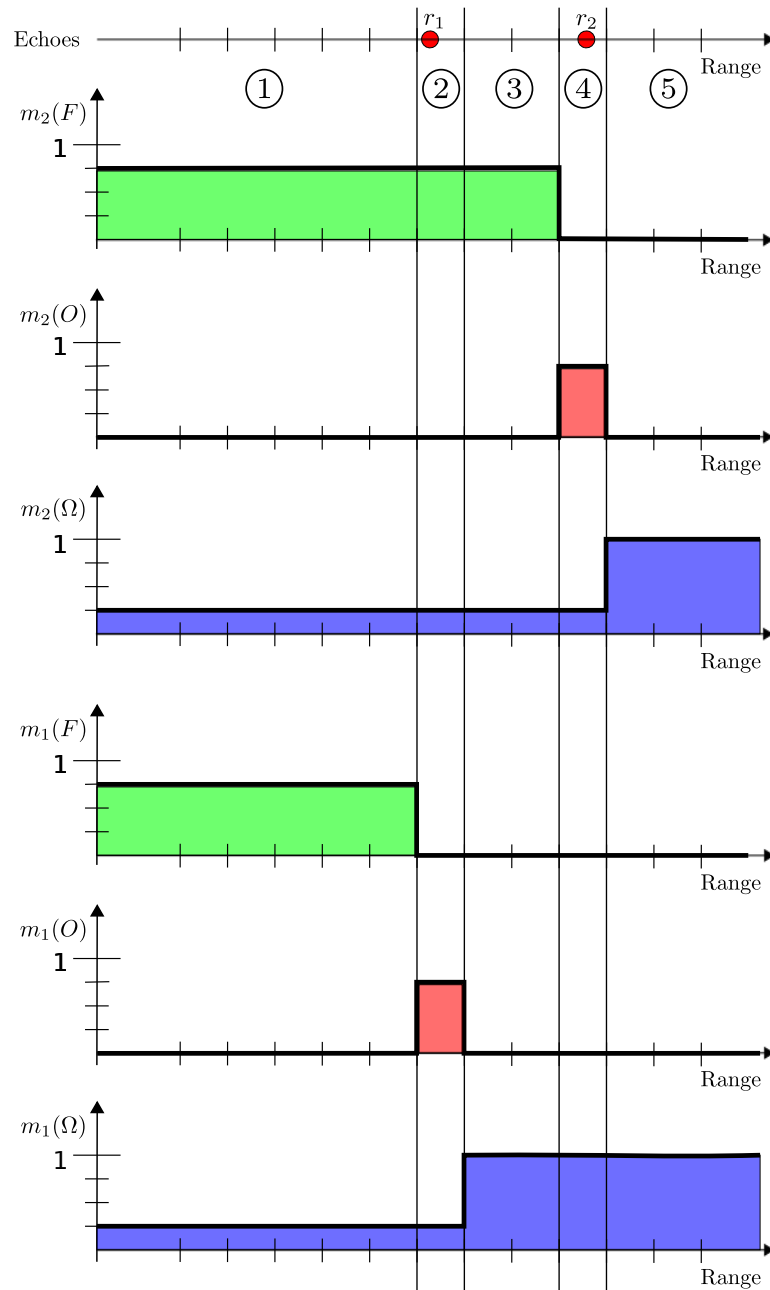


Figure B.1.: Profils du modèle de capteur pour deux mesures élémentaires

D'après les hypothèses de la section 4.3.2, le masquage du point 2 par le point 1 est considéré de la façon suivante. Si les points 1 et 2 sont mesurés c'est que les zones 2 et 4 sont occupées. La zone 1, située devant le point le plus proche, est libre car les deux échos ont pu traverser cette zone. La zone 5 est inconnue car elle est située derrière tous les obstacles, elle est donc masquée. Enfin, la zone 3 est aussi inconnue car elle est située derrière le premier écho, l'obstacle qui produit le premier écho masque donc cette zone.

La technique utilisée consiste à affaiblir totalement la masse sur l'état F pour les cellules situées à partir du premier écho dans le même secteur angulaire. Dans notre cas, on veut donc affaiblir la masse $m_2(F)$ dans les zones 2, 3, 4 et 5, puisque cette information n'est pas fiable d'après la première source.

Pour détecter à partir de quel cellule il faut réaliser l'affaiblissement, on utilise la règle de fusion disjonctive qui permet de traduire l'information commune entre toutes les sources (dans notre cas, l'espace est libre dans la zone 1). Nous avons introduit une distribution de fonction de masse m_f .

$$m_f = \bigoplus_k m_k.$$

On rappelle que la distribution m_f sera soit une fonction de masse vide pour les cellules situées derrière le premier impact (c.f. secteur 2, 3, 4, et 5 sur la figure), soit une fonction de masse avec comme seul élément focal F pour les cellules avant le premier écho.

Les cellule ou $m_f(F) > 0$ sont les cellules pour lesquelles toute les distribution m_k on une part de croyance pour l'état libre.

Comme $m_f(F)$ est le résultat d'un produit de nombres plus petits que 1, on normalise cette masse à 1.

$$m_f(F) = \frac{m_f(F)}{\max(m_f(F))} \text{ si } \max(m_f) \neq 0$$

$$m_f(\Omega) = 1 - m_f(F)$$

La figure B.2 montre le résultat après la fusion et la normalisation.

Dans le cas qui nous intéresse, pour mettre en place cet affaiblissement, on crée une distribution de fonctions de masse de masquage m_m à partir de m_f . Elle est construite de façon à affaiblir la masse sur F des bba de tous les échos (m_k) dans les cellules où $m_f(\Omega) = 1$ et à ne pas affecter les bba dans la zone 1.

Pour affaiblir la masse sur F dans les zones 2, 3, 4 et 5, on pose

$$m_m(O) = \alpha_o = 1$$

Or dans ces zone, $m(\Omega) = 1$ et $m_f(F) = 0$

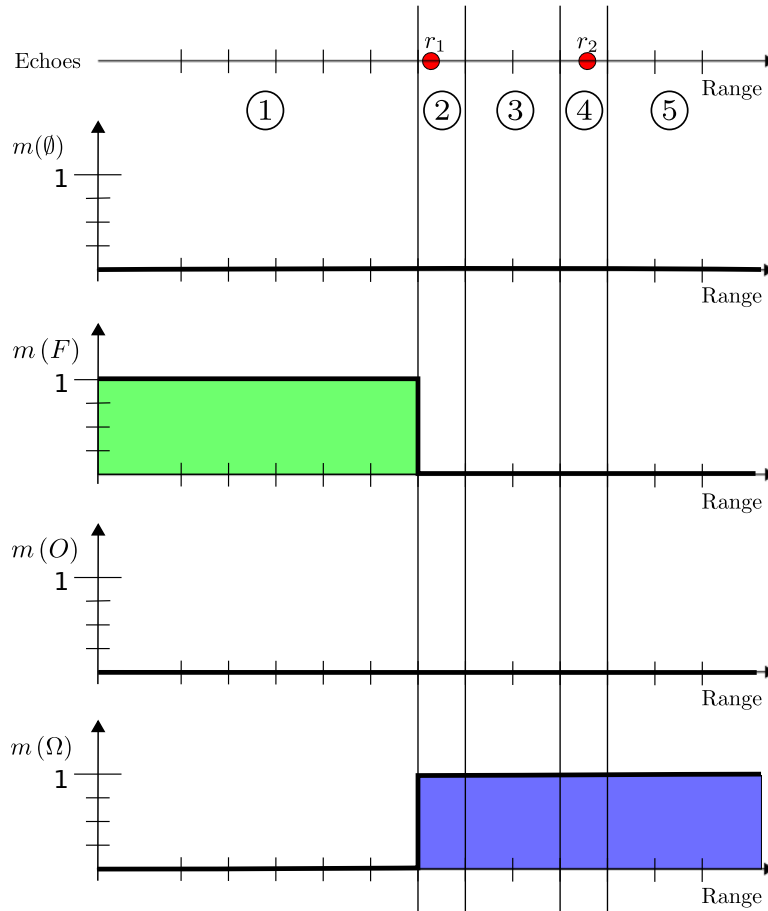


Figure B.2.: Résultat de la construction de m_f à partir des deux bba

$$m_m = \begin{bmatrix} \emptyset & F & O & \Omega \\ m_f(F) = 0 & 0 & m_f(\Omega) = 1 & 0 \end{bmatrix}$$

Par contre dans la zone 1, on ne fait pas d'affaiblissement :

$$\begin{aligned} \alpha &= \alpha_f = \alpha_o = 0 \\ m_m(\emptyset) &= 1 - \sum \alpha = 1 \end{aligned}$$

et comme $m_f(F) = 1$ et $m_f(\Omega) = 0$ dans cette zone :

$$m_m = \begin{bmatrix} \emptyset & F & O & \Omega \\ m_f(F) = 1 & 0 & m_f(\Omega) = 0 & 0 \end{bmatrix}$$

Donc finalement comme $m_f(\Omega)$ est nulle dans la zone 1 et $m_f(F)$ dans les autres zones, on peut écrire pour toutes les zones :

$$m_m = \begin{bmatrix} \emptyset & F & O & \Omega \\ m_f(F) & 0 & m_f(\Omega) & 0 \end{bmatrix}$$

Enfin, on réalise l'affaiblissement en fusionnant de manière disjonctive chaque bba avec la fonction de masquage. Pour créer une bba qui prend en compte ces effets :

$$m_{mk} = m_k \oplus m_m$$

A la fin, chaque bba affaiblie se voit modifiée de façon à ce que si elle avait des cellules libres situées après le premier écho de ce secteur angulaire, ces cellules deviennent inconnues.

La fusion conjonctive des fonctions de masse affaiblies permet de créer un modèle de capteur cohérent ne prenant pas en compte la dépendance des points entres eux.

C. Détection des points du sol

Nous décrivons ici le principe de l'algorithme de détection des points lidar situés sur le plan du sol qui a été utilisé dans ces travaux. Cette algorithme a été proposé par M. Fayad au cours de sa thèse [Fayad, 2009].

L'objectif de la méthode est de calculer l'angle de tangage α à partir d'un ensemble de points des nappes basses (1 et 2) du lidar compte-tenu de l'angle φ entre les nappes et de la divergence ϵ du laser. La configuration du système est décrite dans la figure C.1.

Dans cette approche, on considère les distances projetées sur le plan yz . En faisant l'hypothèse que le point est sur la route, la distance peut être calculée comme :

$$d = \frac{h}{\sin(\gamma)}$$

Pour chaque point de la première nappe qui ont des chances d'être sur la route (on considère la partie centrale du scan dans un intervalle $[\theta_{min} \ \theta_{max}]$), on réalise le processus suivant :

1. Dans un premier temps, on calcule la distance $d1$ pour un point de la nappe 1. Cette distance est la distance théorique de l'écho (projeté sur le plan yz) si l'angle de tangage est nul.
2. Puis on calcule $d2_{p-}$ et $d2_{p+}$ comme les bornes de l'intervalle qui devrait contenir la seconde nappe si celle-ci touche le sol. Ces distances sont calculées à partir de $d1$ puisque l'on connaît l'angle séparant les deux nappes et la divergence du laser ($\varphi - \epsilon$ et $\varphi + \epsilon$).
3. Les points de la première nappe et de la seconde nappe ne correspondant pas au même angle d'azimut θ , on recherche les deux points qui encadrent le point de la première nappe considéré. On calcule les distances $d2_{\theta-}$ et $d2_{\theta+}$ en faisant l'hypothèse que ces points sont sur le sol ce qui permet de calculer le $d2_{int}$ correspondant par interpolation latérale.
4. Enfin, on vérifie si le point interpolé $d2_{int}$ est compris dans l'encadrement $d2_{p-}$ et $d2_{p+}$. Si c'est le cas, on labellise les trois points considérés dans ($d1$, $d2_{\theta-}$ et $d2_{\theta+}$) comme appartenant à la route) et on conserve la distance du point de la nappe basse pour le filtrage)
5. Si le nombre de points détectés est supérieur à un certain seuil, on estime l'angle de tangage.

Finalement, on corrige les coordonnées x et z des points du sol, c'est-à-dire les points dont la coordonnée en z est comprise dans un plan incliné d'angle α dans un intervalle $[h - sl \quad h + sl]$ où sl est un seuil de confiance.

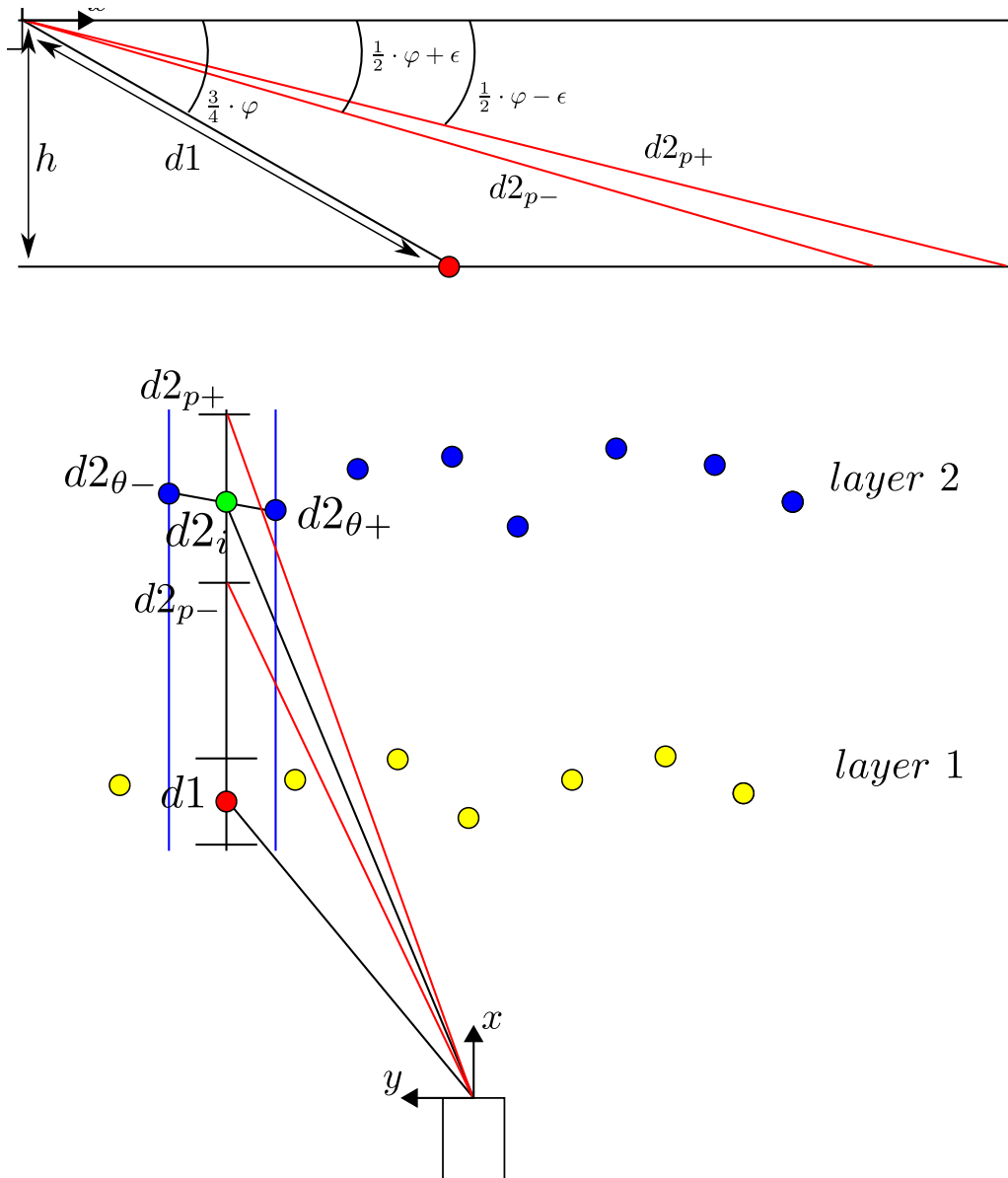


Figure C.1.: Détection du plan du sol. Sur la figure du haut, une vue de profil. Sur la figure du bas, une vue de haut.

Bibliographie

- [Wif, 2003] (2003). Wifibot : <http://www.wifibot.com/>.
- [OSM, 2004] (2004). OpenStreetMap : www.openstreetmap.org.
- [Aitken, 2005] AITKEN, V. (2005). Evidential Mapping for Mobile Robots with Range Sensors. In *2005 IEEE Instrumentation and Measurement Technology Conference Proceedings*, volume 3, pages 2180–2185.
- [Baba, 2007] BABA, A. (2007). *Cartographie de l'Environnement et Suivi Simultané de Cibles Dynamiques par Un Robot Mobile*. Thèse de doctorat, Université de Toulouse.
- [Bar-Shalom, 2000] BAR-SHALOM, Y. (2000). *Multitarget-Multisensor tracking : Applications and Advances*. Artech House.
- [Bayes et Price, 1763] BAYES, M. et PRICE, M. (1763). An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, 53:370–418.
- [Berger, 1985] BERGER, J. (1985). *Statistical decision theory and Bayesian analysis*. Springer-Verlag, New York, 2nd édition.
- [Berlin, 2007] BERLIN, T. (2007). Spirit of Berlin : An Autonomous Car for the DARPA Urban Challenge Hardware and Software Architecture.
- [Bonnifait *et al.*,] BONNIFAIT, P., BOURON, P., CRUBILLE, P. et MEIZEL, D. Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, volume 2, pages 1597–1602.
- [Borenstein et Koren, 1991] BORENSTEIN, J. et KOREN, Y. (1991). Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 7(4):535–539.
- [Bourgault *et al.*, 2002] BOURGAULT, F., MAKARENKO, A., WILLIAMS, S., GROCHOLSKY, B. et DURRANT-WHYTE, H. (2002). Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 1, pages 540–545.
- [Bradski et Kaehler, 2008] BRADSKI, G. et KAEHLER, A. (2008). *Learning OpenCV*, volume 16. O'Reilly.
- [Buehler *et al.*, 2009] BUEHLER, M., IAGNEMMA, K. et SINGH, S., éditeurs (2009). *The DARPA Urban Challenge : Autonomous Vehicles in City Traffic*, George

- Air Force Base, Victorville, California, USA*, volume 56 de *Springer Tracts in Advanced Robotics*. Springer.
- [Cherubini et Chaumette, 2011] CHERUBINI, A. et CHAUMETTE, F. (2011). Visual navigation with obstacle avoidance. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1593–1598.
- [Coué et al., 2006] COUÉ, C., PRADALIER, C., LAUGIER, C., FRAICHARD, T. et BESSIÈRE, P. (2006). Bayesian Occupancy Filtering for Multitarget Tracking : An Automotive Application. *The International Journal of Robotics Research*, 25(1):19–30.
- [Delmas, 2011] DELMAS, P. (2011). *Génération active des déplacement d'un véhicule agricole dans son environnement*. Thèse de doctorat, Université de Clermont Ferrand.
- [Dempster, 1968] DEMPSTER, A. P. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, 30:205–247.
- [Denoeux, 2006] DENOEU, T. (2006). The cautious rule of combination for belief functions and some extensions. *9th International Conference on Information Fusion*, pages 1–8.
- [Dubois et Prade, 1990] DUBOIS, D. et PRADE, H. (1990). Consonant approximations of belief functions. *International Journal of Approximate Reasoning*, 4(5-6):419–449.
- [Durrant-whyte, 1988] DURRANT-WHYTE, H. (1988). Sensor Models and Multisensor Integration. *The International Journal of Robotics Research*, 7(6):97–113.
- [Elfes, 1989a] ELFES, A. (1989a). A tessellated probabilistic representation for spatial robot perception and navigation. In *Proceedings of the NASA Conference on Space Telerobotics*, pages Volume 2 p 341–350. JPL, California Inst. of Tech.
- [Elfes, 1989b] ELFES, A. (1989b). *Occupancy grids : A probabilistic framework for robot perception and navigation*. Thèse de doctorat, Carnegie Mellon University.
- [Elfes, 1989c] ELFES, A. (1989c). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.
- [Elfes, 1990] ELFES, A. (1990). Occupancy Grids : A Stochastic Spatial Representation for Active Robot Perception. In *Proceedings of the Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-90)*, pages 136–146, New York, NY. Elsevier Science.
- [Fayad, 2009] FAYAD, F. (2009). *Contribution au développement de méthodes de perception pour un véhicule routier*. Thèse de doctorat, Université de technologie de Compiègne.
- [Fayad et Cherfaoui, 2007] FAYAD, F. et CHERFAOUI, V. (2007). Tracking objects using a laser scanner in driving situation based on modeling target shape. In *2007 IEEE Intelligent Vehicles Symposium*, pages 44–49, Istanbul.

- [Fayad et Cherfaoui, 2008] FAYAD, F. et CHERFAOUI, V. (2008). Detection and Recognition confidences update in a multi-sensor pedestrian tracking system. *In Int. Conf on Information Processing and Management of Uncertainty in knowledge-Based Systems, IPMU 2008, Malaga*, pages 409–416.
- [Ferreira et al., 2010] FERREIRA, J. a. F., LOBO, J. et DIAS, J. (2010). Bayesian real-time perception algorithms on GPU. *Journal of RealTime Image Processing*, 6(3):171–186.
- [Gate, 2009] GATE, G. (2009). *Reliable Perception of Highly Changing Environment*. Thèse de doctorat, Ecole Nationale Supérieure Des Mines De Paris.
- [Gidel et al., 2008] GIDEL, S., CHECCHIN, P., BLANC, C., CHATEAU, T. et TRAS-SOUDAINE, L. (2008). Parzen method for fusion of laserscanner data : Application to pedestrian detection. *2008 IEEE Intelligent Vehicles Symposium*, pages 319–324.
- [Hähnel et al., 2005] HÄHNEL, D., THRUN, S., WEGBREIT, B. et BURGARD, W. (2005). Towards lazy data association in SLAM. *Robotics Research*.
- [Herrmann et al., 2010] HERRMANN, D., KAMPFANS, T. et LANGETEPE, E. (2010). Exploring Simple Triangular and Hexagonal Grid Polygons Online.
- [Himmelsbach et al., 2009] HIMMELSBACH, M., LUETTEL, T. et WUENSCH, H.-J. (2009). Real-time object classification in 3D point clouds using point feature histograms. *In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 994–1000.
- [Himmelsbach et al., 2008] HIMMELSBACH, M., MÜLLER, A., MUELLER, A., LUETTEL, T. et WUENSCH, H.-J. (2008). LIDAR-based 3D Object Perception. *In Proceedings of 1st International Workshop on Cognition for Technical Systems*, Munich.
- [Jensfelt, 2001] JENSFELT, P. (2001). *Approaches to Mobile Robot Localization in Indoor Environments*. Thèse de doctorat, Royal Institute of Technology.
- [Julier et Uhlmann, 2004] JULIER, S. J. et UHLMANN, J. K. (2004). Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422.
- [Kalman, 1960] KALMAN, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems 1. *Journal Of Basic Engineering*, 82(Series D):35–45.
- [Kamel Mekhnacha, Yong Mao, David Raulo, 2009] KAMEL MEKHNACHA, YONG MAO, DAVID RAULO, C. L. (2009). The Fast Clustering-Tracking Algorithm in the Bayesian Occupancy Filter Framework. *In HAHN, H., KO, H. et LEE, S., éditeurs : Multisensor Fusion and Integration for Intelligent Systems*, volume 35, pages 201–219. Springer Berlin Heidelberg.
- [Khatib et Groen, 2009] KHATIB, O. et GROEN, F. (2009). *The DARPA Urban Challenge*, volume 56 de *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg.
- [Kneip et al., 2009] KNEIP, L., TACHE, F., CAPRARI, G. et SIEGWART, R. (2009). Characterization of the compact Hokuyo URG-04LX 2D laser range scanner.

- [Konolige et Chou, 1999] KONOLIGE, K. et CHOU, K. (1999). Markov Localization using Correlation. In *IJCAI '99 Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1154 – 1159.
- [Kurdej, 2011] KURDEJ, M. (2011). Construction et combinaison de grilles d'occupation à partir de cartes vectorielles. Rapport technique, Université de Technologie de Compiègne, Compiègne.
- [Kurdej et al., 2012] KURDEJ, M., MORAS, J., CHERFAOUI, V. et BONNIFAIT, P. (2012). Map-Aided Fusion Using Evidential Grids for Mobile Perception in Urban Environment. In DENOEUX, T. et MASSON, M.-H., éditeurs : *International Conference on Belief Functions*, volume 164 de *Advances in Intelligent and Soft Computing*, pages 343–350. Springer.
- [Larsson et Fornberg, 2005] LARSSON, E. et FORNBERG, B. (2005). Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Computers & Mathematics with Applications*, 9810751.
- [Lawrenz, 1997] LAWRENZ, W. (1997). *CAN System Engineering : From Theory to Practical Applications*. Springer.
- [Levinson et Thrun, 2010] LEVINSON, J. et THRUN, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *2010 IEEE International Conference on Robotics and Automation*, pages 4372–4378.
- [Levitt et Lawton, 1990] LEVITT, T. S. et LAWTON, D. T. (1990). Qualitative Navigation for Mobile Robots. *Artif. Intell.*, 44(3):305–360.
- [Mercier et al., 2005] MERCIER, D., QUOST, B. et DENOEUX, T. (2005). Contextual Discounting of Belief Functions. *ECSQARU05 8th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, 3571 LNAI:552–562.
- [Miyasaka et al., 2009] MIYASAKA, T., OHAMA, Y. et NINOMIYA, Y. (2009). Egomotion estimation and moving object tracking using multi-layer LIDAR. In *2009 IEEE Intelligent Vehicles Symposium*, pages 151–156.
- [Montemerlo et al., 2008] MONTEMERLO, M., BECKER, J., BHAT, S., DAHLKAMP, H., DOLGOV, D., ETTINGER, S., HAEHNEL, D., HILDEN, T., HOFFMANN, G., HUHNKE, B., JOHNSTON, D., KLUMPP, S., LANGER, D., LEVANDOWSKI, A., LEVINSON, J., MARCIL, J., ORENSTEIN, D., PAEFGEN, J., PENNY, I., PETROVSKAYA, A., PFLUEGER, M., STANEK, G., STAVENS, D., VOGT, A. et THRUN, S. (2008). Junior : The Stanford entry in the Urban Challenge. *Journal of Field Robotics*, 25(9):569–597.
- [Montemerlo et al., 2002] MONTEMERLO, M., THRUN, S., KOLLER, D. et WEGBREIT, B. (2002). FastSLAM : A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598.
- [Moras et al., 2010] MORAS, J., CHERFAOUI, V. et BONNIFAIT, P. (2010). A lidar perception scheme for intelligent vehicle navigation. In *2010 11th International*

- Conference on Control Automation Robotics and Vision*, numéro December, pages 1809–1814.
- [Moras *et al.*, 2011a] MORAS, J., CHERFAOUI, V. et BONNIFAIT, P. (2011a). Credibilist occupancy grids for vehicle perception in dynamic environments. *In 2011 IEEE International Conference on Robotics and Automation*, pages 84–89, Shanghai.
- [Moras *et al.*, 2011b] MORAS, J., CHERFAOUI, V. et BONNIFAIT, P. (2011b). Moving Objects Detection by Conflict Analysis in Evidential Grids. *In 2011 IEEE Intelligent Vehicles Symposium IV*, pages 1122–1127.
- [Moras *et al.*, 2012] MORAS, J., RODRIGUEZ FLOREZ, S. A., DREVELLE, V., DHERBOMEZ, G., CHERFAOUI, V. et BONNIFAIT, P. (2012). Drivable Space Characterization using Automotive Lidar and Georeferenced Map Information. *In 2012 IEEE Intelligent Vehicles Symposium*.
- [Nashashibi *et al.*, 2008] NASHASHIBI, F., KHAMMARI, A. et LAURGEAU, C. (2008). Vehicle recognition and tracking using a generic multisensor and multialgorithm fusion approach. *International Journal of Vehicle Autonomous Systems*, 6(1/2): 134.
- [Opencl *et al.*, 2011] OPENCL, K., GROUP, W. et MUNSHI, A. (2011). OpenCL Specification. *ReVision*, 91(2):1–385.
- [Optics, 2006] OPTICS, G. B. (2006). Gaussian Beam Optics 2. *In Technology*, pages 1–14. Melles Griot, CVI Melles Griot.
- [Pagac *et al.*, 1998] PAGAC, D., NEBOT, E. et DURRANT-WHYTE, H. (1998). An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4):623–629.
- [Paromtchik *et al.*, 2011] PAROMTCHIK, I. E., PERROLLAZ, M. et LAUGIER, C. (2011). Fusion of telemetric and visual data from road scenes with a lexus experimental platform. *In 2011 IEEE Intelligent Vehicles Symposium IV*, numéro Iv, pages 746–751.
- [Petrovskaya et Thrun, 2009] PETROVSKAYA, A. et THRUN, S. (2009). Model Based Vehicle Tracking in urban environments. *In Proceedings 2009 IEEE International Conference on Robotics and Automation*.
- [Rey et Carlos, 2006] REY, U. et CARLOS, J. (2006). Dynamic gridmaps : comparing building techniques. *Mathware & Soft Computing*, 7.
- [Ristic *et al.*, 2004] RISTIC, B., ARULAMPALAM, S. et GORDON, N. (2004). *Beyond the Kalman Filter : Particle Filters for Tracking Applications*. Artech House.
- [Rodriguez Florez *et al.*, 2008] RODRIGUEZ FLOREZ, S. A., FREMONT, V. et BONNIFAIT, P. (2008). Influence of Intrinsic Parameters over Extrinsic Calibration between a Multi-Layer Lidar and a Camera. *In IEEE IROS08 2nd Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, page 34.
- [Rodriguez Florez *et al.*, 2009] RODRIGUEZ FLOREZ, S. A., FREMONT, V. et BONNIFAIT, P. (2009). An experiment of a 3D real-time robust visual odometry for

- intelligent vehicles. *In 2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6.
- [Royer *et al.*, 2005] ROYER, E., BOM, J., DHOME, M., THUÏLOT, B., LHULLIER, M. et MARMOITON, F. (2005). Outdoor autonomous navigation using monocular vision. *In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1253–1258, Edmonton, Canada.
- [Rusu et Cousins, 2011] RUSU, R. B. et COUSINS, S. (2011). 3D is here : Point Cloud Library (PCL). *2011 IEEE International Conference on Robotics and Automation*, pages 1–4.
- [Shafer, 1976] SHAFER, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press, Princeton.
- [Shannon, 1948] SHANNON, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(July 1928):379–423.
- [Shenoy, 1989] SHENOY, P. P. (1989). A valuation-based language for expert systems. *International Journal of Approximate Reasoning*, 3(5):383–411.
- [Smets, 1990] SMETS, P. (1990). Constructing the pignistic probability function in a context of uncertainty. *Uncertainty in artificial intelligence*, pages 29–39.
- [Smets, 1993] SMETS, P. (1993). Belief functions : The disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9(1):1–35.
- [Smets et Kennes, 1994] SMETS, P. et KENNES, R. (1994). The transferable belief model. *Artificial Intelligence*, 66(2):191–234.
- [Steux et El, 2010] STEUX, B. et EL, O. (2010). TinySLAM : a SLAM Algorithm in less than 200 lines C-Language Program. *In 11th. Int. Conf. Control, Automation, Robotics and Vision*, numéro December, pages 1975–1979.
- [Sutton *et al.*, 2009] SUTTON, M. A., ORTEU, J. J. et SCHREIER, H. W. (2009). *Image Correlation for Shape, Motion and Deformation Measurements- Basic Concepts, Theory and Applications*. Springer Science.
- [Tay *et al.*, 2008] TAY, M. K., MEKHNACHA, K., YGUEL, M., COUÉ, C., PRADALIÉ, C. et LAUGIER, C. (2008). The Bayesian occupation filter. *In BESSIÈRE, P., LAUGIER, C. et SIEGWART, R., éditeurs : Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*, pages 77–98. Springer Berlin Heidelberg, springer t édition.
- [Thrun, 2000] THRUN, S. (2000). Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *The International Journal of Robotics Research*, 19(11):972–999.
- [Thrun, 2003] THRUN, S. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127.
- [Thrun *et al.*, 1998] THRUN, S., BURGARD, W. et FOX, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 31:29–53.

- [Urmson *et al.*, 2007] URMSON, C., ANHALT, J., BAGNELL, D., BAKER, C., BITTNER, R., DOLAN, J., DUGGINS, D., FERGUSON, D., GALATALI, T., GEYER, C., GITTLEMAN, M., HARBAUGH, S., HEBERT, M., HOWARD, T., KELLY, A., KOHANBASH, D., LIKHACHEV, M., MILLER, N., PETERSON, K., RAJKUMAR, R., RYBSKI, P., SALESKY, B., SCHERER, S., WOO-SEO, Y., SIMMONS, R., SINGH, S., SNIDER, J., STENTZ, A., WHITTAKER, W. R., ZIGLAR, J., STRUBLE, J. et TAYLOR, M. (2007). Tartan Racing : A Multi-Modal Approach to the DARPA Urban Challenge. *Defense*, 94(4):386–387.
- [Vu *et al.*, 2011] VU, T.-D., BURLET, J. et AYCARD, O. (2011). Grid-based localization and local mapping with moving object detection and tracking. *Information Fusion*, 12(1):58–69.
- [Wang, 2004] WANG, C.-c. (2004). *Simultaneous Localization, Mapping and Moving Object Tracking*. Thèse de doctorat, Carnegie Mellon University.
- [Wurm *et al.*, 2007] WURM, K. M., STACHNISS, C., GRISETTI, G. et BURGARD, W. (2007). Improved Simultaneous Localization and Mapping using a Dual Representation of the Environment. In *Proceedings of the 3rd European Conference on Mobile Robots*.
- [Xie *et al.*, 2010] XIE, J., NASHASHIBI, F., PARENT, M. et FAVROT, O. G. (2010). A real-time robust global localization for autonomous mobile robots in large environments. In *11th International Conference on Control Automation Robotics and Vision*, numéro December, pages 1397–1402.
- [Xu et Collins, 2010] XU, H. et COLLINS, J. J. (2010). Mobile robot localization using feature based fuzzy map matching. In *2010 11th International Conference on Control Automation Robotics & Vision*, numéro December, pages 1968–1974.
- [Yager, 1987] YAGER, R. R. (1987). On the Dempster-Shafer framework and new combination rules. *Information sciences*, 41(2):93–138.
- [Yager, 2008] YAGER, R. R. (2008). Entropy and Specificity in a Mathematical Theory of Evidence. In YAGER, R. et LIU, L., éditeurs : *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 de *Studies in Fuzziness and Soft Computing*, pages 291–310. Springer Berlin / Heidelberg.
- [Zadeh, 1984] ZADEH, L. A. (1984). Book Review : A Mathematical Theory of Evidence. *AI Magazine*, 5(3):81–83.
- [Zhao *et al.*, 2009] ZHAO, H., CHIBA, M., SHIBASAKI, R., SHAO, X., CUI, J. et ZHA, H. (2009). A Laser-Scanner-Based Approach Toward Driving Safety and Traffic Data Collection. *IEEE Transactions on Intelligent Transportation Systems*, 10(3): 534–546.

Nomenclature

ABS	Anti-lock Braking System
AI	Accumulation level
CAN	Controller Area Network
CLD	Carte Locale Dynamique
DARPA	Defense Advanced Research Projects Agency
ECEF	Earth Centred Earth Fixed
EKF	Extended Kalman Filter
ENU	East, Nord, Up
EPS	Electronic Stability Program
GPS	Global Positioning System
IGN	Institut Géographique National
JPDA	Joint Probabilistic Data Association
LIDAR	LIght Detection And Ranging
MEMS	Micro Electro-Mechanical Systems
MG	MapGrid
MHT	Multiple Hypothesis Tracking
MNT	Modèles Numériques de Terrain
MOT	Moving Object Tracking
NN	Nearest Neighbour
PACPUS	Perception et Assistance Pour une Conduite PIUs Sûre
PG	PriorGrid
RTK	Real Time Kinematic

SBAS	Satellite-Based Augmentation System
SG	ScanGrid
SLAM	Simultaneous Localization And Mapping
SLAMMOT	Moving Object Tracking and Moving Object Tracking
UKF	Unscented Kalman Filter