

ACADEMIE DE MONTPELLIER
UNIVERSITE MONTPELLIER II
– Sciences et Techniques du Languedoc –

THESE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE MONTPELLIER II

Discipline : **Génie Informatique, Automatique et Traitement du signal**

Formation doctorale : **Systèmes Automatiques et Microélectroniques**

Ecole doctorale : **Information, Structures, Systèmes**

présentée et soutenue publiquement

par

Geovany ARAUJO BORGES

Le 27 mai 2002

Titre :

**Cartographie de l'environnement et localisation robuste pour la
navigation de robots mobiles**

JURY

Mme. Marie-José ALDON	Chargée de recherche CNRS au LIRMM	Directrice de thèse
M. Michel DEVY	Directeur de recherche CNRS au LAAS	Rapporteur
M. Alain FOURNIER	Professeur à l'Université Montpellier II	Président
M. Dominique MEIZEL	Professeur à l'Université de Technologie de Compiègne	Rapporteur
M. El Mustapha MOUADDIB	Professeur à l'Université de Picardie Jules Verne	Examineur
M. Patrick RIVES	Directeur de recherche INRIA Sophia-Antipolis	Examineur

A mes parents bien-aimés, Drausio Rogério et Avany, qui n'ont jamais baissé les bras devant les difficultés de la vie pour pouvoir éduquer leurs enfants. Cette thèse est la vôtre.

Remerciements

Voici une page où je risque d'être encore plus bavard que dans le reste du manuscrit. Est-ce possible ?

Je souhaite remercier Michel Habib, directeur du LIRMM, Christian Landrault, directeur adjoint, et Gaston Cambon, ancien directeur, pour m'avoir accueilli au laboratoire pour préparer cette thèse. Tous mes remerciements à Pierre Dauchez, ancien responsable du Département de Robotique, qui a eu la gentillesse de retransmettre ma proposition de candidature aux autres membres du département. Merci à Etienne Dombre, actuel responsable du département, qui, avec sympathie et intérêt, a suivi mon travail au sein du groupe. Merci à la CAPES, organisme du gouvernement brésilien qui a financé ma bourse et respecté rigoureusement son engagement pour ma formation en France.

Je transmets ma reconnaissance aux membres du jury de soutenance, qui ont accepté d'analyser bénévolement ce travail, en portant un regard critique et pertinent, nécessaire à tout travail scientifique. Je remercie ainsi les rapporteurs M. Michel Devy et M. Dominique Meizel, le président du jury M. Alain Fournier, ainsi que les examinateurs M. El Mustapha Mouaddib et M. Patrick Rives.

Un travail de thèse, qui doit mettre en évidence les apports originaux de son auteur, requiert entre autres une orientation perspicace, critique et expérimentée du directeur de thèse. Comment remercier ainsi ma directrice de thèse, Mme. Marie José-Aldon, qui a su conduire ce travail avec rigueur et dévouement, même quand le moral était au plus bas. A chaque retour de conférence je me posais encore des questions, auxquelles elle a su donner de bonnes réponses. En dehors du plan technique, nous avons eu un rapport d'amitié et de respect. Pour tout cela, un grand merci Marie-Jo !

Aux membres du département, j'adresse mon respect et mes remerciements pour la confiance

qu'ils m'ont accordée. Mes vœux sincères de succès à David Andreu, Etienne Dombre, Didier Crestani, André Crosnier, Pierre Dauchez, Alain Fournier, Philippe Fraisse, David Guiraud, Bruno Jouvencel, Alain Liegeois, François Pierrot, Philippe Poignet, François Prunet, Olivier Strauss, Jérôme Vaganay, René Zapata. Jamais je n'oublierai la question qui m'a été posée par Pierre Dauchez et François Pierrot lors de notre voyage à San Francisco en 2000 : "*A quoi ça sert la localisation ? En avons nous réellement besoin ?*". Cette question m'a perturbé une journée entière, d'une part parce que mes réponses n'étaient pas assez convaincantes, mais, encore plus grave, parce que je n'avais jamais réfléchi à cela. Je pense y avoir apporté des réponses dans ce manuscrit.

Merci à ceux qui assurent le bon fonctionnement du laboratoire. Je remercie particulièrement Josette Durante, avec qui j'ai appris à parler vite (ceux qui étaient à la soutenance le savent), Nicole Olivet pour son esprit chaleureux, Nadine et Philippe Tilloy (1, 2, 3, zéro), Anne Bancel, Ghislaine Takessian, Marie Mejean, Freddy, Patrice et Martine Périquier qui a dû supporter ma présence permanente à la bibliothèque au début des travaux. Un grand merci à Thierry Gil, qui, avec ses compétences techniques, a conçu avec Michel Benoît la carte entrées-sorties PCI du robot Omni. Merci à ce dernier pour m'avoir permis d'éviter les pannes depuis avril 2000. Sans aucun doute, c'est à Michel Benoît que je dois la possibilité d'expérimenter sur une plate-forme mobile robuste. Ses compétences techniques ont permis la naissance du robot Omni. En plus, j'ai beaucoup apprécié la confiance qu'il m'a accordée.

Durant ces trois années et demi, voici les héros du LIRMM qui ont su créer un environnement très agréable et humain (et tout ça en maintenant des inter-corrélations de façon consistante) : Frédéric Comby (mon attaquant préféré) et Tiphaine, Julien Aragones (compagnon de bataille avec Omni), Sébastien Krut (incroyable, il faut trouver une solution à son problème), Frédéric Marquet (sa persévérance m'a presque convaincu d'apprendre l'occitan), Benoît Telle (cool), Jean-Damien Carbou (merci pour les *brigadeiros*), Olivier Company (mon remplaçant pour la nuit), Gilles Duchemin (c'est la classe pour *smatcher*), Vincent Creuze (capitaine Némó), Philippe Lucidarne (le gars du nord et ses robots), Philippe Rongier (temps moyen de réponse de l'ordre de *10 ms*), Christophe Caby (pas moi), Christine Coves (qui a pris Woodstock?), Yan Lecorre, Arnaud Lelevé, Philippe Baccou (tu as compris le coup?), Abraham Lopez (le planificateur, *arriba!*), Ivan Espinosa (*arriba! arriba!*), Tomas Salgado (*arriba! arriba! arriba!*), Daniel Roman (je ne peux plus), Andrés (*buenos dias*), Florian Pantaleão (et sa famille, bien sûr), Christophe Rabaud, Jérôme Albaric (ouf), Emmanuel Matéo, Sébastien Druon, Fabien Lydoire (ça marche?), Philippe Maurine, Jean-Max, Serge Bernard, Isabelle, et tous les autres (ça devient long) ...

En dehors de mon environnement de travail, j'ai eu le plaisir de faire la connaissance d'êtres formidables. Dans le début, une époque normalement difficile, j'ai beaucoup compté sur Suelo et Carmen Numazawa, un couple brésilien qui m'a accueilli après un mauvais départ dans l'auberge de jeunesse de Montpellier. Aux amis brésiliens, qui ont aidé à conserver un pied ici et l'autre là bas, de l'autre côté de l'atlantique : Rolf et Simoni, Ricardo et Sheila, Sandro et Rosane, Almir et Edinolia, Givago, Pricilla, Mirela et Alessandro, Marta, Alberto et Cristina. A mes amis parisiens de l'ENST : Fernando Rangel, Elizabeth et Yuri Catunda, pour leurs conseils et leur amitié. A nos amis français, pour les moments agréables et l'échange culturel : Hervé, Ghislaine et Stéphane. A Jean-Claude Aldon, pour sa franchise et son ouverture caractéristique du sud ; il nous a conquis.

A Jean Sallantin et Francine, un couple hors norme. A ma chère Colette, tu nous manques.

Je ne peux pas oublier mes amis ultra-atlantique : Maurício (*sócio*) et Luciana (hebdomadaire féminin), Madeiro (*tenso*), André Menezes, avec lesquels les choses ne vont jamais changer. Aux professeurs et amis Antonio Marcus Nogueira Lima et Gurdip Singh Deep, avec lesquels j'ai eu le plaisir de travailler pendant six ans et qui ont beaucoup contribué à ma formation scientifique à Campina Grande, au Brésil.

A mes parents, qui comme la dédicace l'exprime sont les grands responsables pour celui qui a écrit ce manuscrit, à ma sœur Rilena et à mon frère Arthur. Je risque de ne plus habiter à la maison. A ma chère belle-famille, Carlton, Iara, Christus et Juliana, qui m'ont accueilli ouvertement.

Et à toi ma petite Germana, en dernier, mais toujours la première, pour ton support et ton amour, *juntos*. Sans cela, qui serais-je ?

Table des matières

Table des matières	v
Table des figures	xiii
Liste des tableaux	xvii
1 Introduction générale	1
1.1 Définition du problème	1
1.2 Principales difficultés	3
1.3 Plate-forme mobile Omni	3
1.4 Architecture de localisation et de cartographie	4
1.5 Contributions	6
1.6 Plan du manuscrit	6
2 Etat de l’art : méthodes de cartographie et de localisation par cartes géométriques	9
2.1 Introduction	9
2.2 Segmentation de données et cartographie locale	11
2.3 Localisation	12
2.3.1 Autolocalisation	13

2.3.2	Localisation relative par estime	14
2.3.3	Localisation par mise en correspondance de cartes géométriques	14
2.4	Cartographie et localisation concurrentes	16
3	Acquisition et prétraitement des données multisensorielles	21
3.1	Introduction	21
3.2	Système multicapteur	22
3.3	Cohérence des images multicapteur	24
3.4	Processus d'acquisition	25
3.5	Prétraitement des données	28
3.5.1	Caractérisation des mesures du télémètre laser	29
3.5.2	Etude de l'influence du mouvement sur les images télémétriques	30
3.5.3	Procédures de prétraitement	33
3.6	Evaluation expérimentale	35
3.6.1	Expérience sur la correction du biais	35
3.6.2	Expérience sur la compensation du mouvement	35
3.6.3	Expérience sur la pseudo-synchronisation des images	36
3.7	Conclusion	36
4	Segmentation des images	39
4.1	Introduction	39
4.2	Segmentation d'images vidéo	40
4.2.1	Détection de contours verticaux	41
4.2.2	Détection des droites verticales	42
4.3	Segmentation d'images télémétriques	45
4.3.1	Indices d'intérêt dans les images télémétriques	45
4.3.2	Détection de points de cassure	47
4.3.3	Extraction de segments de droite	53
4.3.4	Noyaux pour l'extraction de segments	54
4.4	Simulations	61
4.5	Résultats expérimentaux	64
4.5.1	Segmentation d'images vidéo	66
4.5.2	Segmentation d'images télémétriques	66

4.6	Conclusions	71
5	Cartographie locale	73
5.1	Introduction	73
5.2	Construction de la carte locale	74
5.2.1	Traitement des indices télémétriques	75
5.2.2	Fusion avec les indices vidéo	79
5.3	Evaluation expérimentale	84
5.3.1	Contre-exemple de la détection d'arêtes photométriques	86
5.4	Conclusions	87
6	Localisation dynamique robuste	91
6.1	Introduction	91
6.2	Localisation relative à l'estime	92
6.2.1	Modèle cinématique du robot Omni	92
6.2.2	Odométrie	94
6.2.3	Gyrodométrie	94
6.2.4	Identification paramétrique	95
6.2.5	Modèle de propagation de l'erreur	97
6.3	Localisation référencée environnement	98
6.3.1	Principe et difficultés	98
6.3.2	Approches par filtrage stochastique	99
6.3.3	Discussion à propos des approches <i>FKE</i>	101
6.3.4	Nouvelles approches par optimisation non-linéaire	102
6.3.5	Analyse de complexité	112
6.4	Etudes en simulation	114
6.5	Résultats expérimentaux	115
6.5.1	Expérience dans le site <i>Premier étage extension - B</i>	118
6.5.2	Expérience dans le site <i>Couloir robotique - B</i>	122
6.6	Conclusion	125
7	Cartographie globale découplée	127
7.1	Introduction	127

7.2	Aspects théoriques de la cartographie et localisation découplés	128
7.3	Fusion de structures cartographiques	128
7.3.1	Fusion de demi-plans	129
7.3.2	Fusion d'arêtes	130
7.3.3	Exemple simulé	130
7.4	Fusion de structures sous contraintes	132
7.4.1	Propagation de contraintes stochastiques	132
7.4.2	Une nouvelle approche pour la fusion de structures cartographiques	133
7.5	Cartographie globale découplée	136
7.5.1	Mise en correspondance et classification des structures	137
7.5.2	Inclusion de structures non supportées	137
7.5.3	Fusion des observations	138
7.5.4	Conservation de l'espace libre	141
7.5.5	Fusion de ressemblances	142
7.6	Etudes en simulation	144
7.6.1	Environnement de simulation	145
7.6.2	Résultats de la cartographie découplée sous contraintes	146
7.7	Résultats expérimentaux	147
7.7.1	Cartographie du site <i>Premier étage extension - B</i>	147
7.7.2	Cartographie du site <i>Halle mécatronique</i>	150
7.8	Conclusion	150
8	Conclusion générale et perspectives	153
	Bibliographie	158
	Annexes	171
A	Repères et transformations dans l'espace 2-D	173
A.1	Repères principaux du robot Omni	173
A.2	Transformation de coordonnées	174
A.3	Relation entre repères	175
A.3.1	Position du robot dans un repère de référence	175
A.3.2	Position du repère capteur dans le repère robot	175

B	Représentation et propagation de l'incertitude	177
B.1	Introduction	177
B.2	Processus gaussiens	177
B.3	Représentation de l'incertitude paramétrique	181
B.4	Propagation de l'incertitude	181
B.4.1	Propagation par linéarisation	181
B.4.2	Propagation par échantillonnage sélectif	184
B.5	Distance de Mahalanobis	185
C	Primitives géométriques 2-D	187
C.1	Modélisation	187
C.1.1	Points	187
C.1.2	Droites	187
C.1.3	Segments de droite	188
C.2	Changement de repère	188
C.2.1	Indices de type point	189
C.2.2	Indices de type droite	189
C.3	Relations	190
C.3.1	Angle de vue d'un point	190
C.3.2	Intersection de deux droites	190
D	Représentation géométrique d'environnements	193
D.1	Structures cartographiques	193
D.1.1	Structures de type demi-plan	193
D.1.2	Structures de type arête	195
D.2	Changement inverse de repère	195
D.3	Correspondance de structures cartographiques	196
D.3.1	Correspondance entre demi-plans	196
D.3.2	Correspondance entre arêtes	197
D.3.3	Notation pour les correspondances	198
D.4	Cartes d'environnement	198
D.4.1	Symbolique	198
D.4.2	Carte locale et carte globale	199

D.5	Mise en correspondance de cartes	200
E	Filtres stochastiques	201
E.1	Introduction	201
E.2	Consistance des estimées	202
E.3	Filtrage de systèmes linéaires	203
E.3.1	Filtre de Kalman	204
E.3.2	Intersection de covariances	205
E.3.3	Exemple d'application	208
E.3.4	Intersection de covariances modifiée	208
E.4	Filtrage de systèmes non-linéaires	210
E.4.1	Filtre de Kalman étendu	210
E.4.2	Filtre de Kalman étendu itératif	211
E.4.3	Limitations	211
F	Etalonnage du système multicapteur	215
F.1	Analyse géométrique	215
F.2	Simulation	217
F.3	Procédure d'étalonnage	218
G	Données des expériences	223
G.1	Introduction	223
G.2	Halle robotique - A	224
G.3	Halle robotique - B	225
G.4	Couloir robotique - A	226
G.5	Couloir robotique - B	226
G.6	Premier étage extension - A	227
G.7	Premier étage extension - B	227
G.8	Rez-de-chaussée extension - A	228
G.9	Rez-de-chaussée extension - B	229
G.10	Halle mécatronique	230
H	Tableaux de performance des méthodes de localisation	231

Notations	235
Index	239

Table des figures

1.1	Localisation d'un robot mobile par mise en correspondance de cartes d'environnement	2
1.2	Robot Omni.	4
1.3	Schéma général de l'architecture de localisation et de cartographie.	5
2.1	Représentations d'environnement.	11
3.1	(a) Une image caméra et (b) les repères attachés à ce capteur.	22
3.2	(a) Image de profondeur fournie par le télémètre laser. (b) Illustration du repère télémétrique. (c) Plan de balayage laser.	24
3.3	(a) Système multicapteur embarqué sur le robot Omni. (b) Configuration simplifiée des repères capteurs.	25
3.4	Séquence temporelle des événements associés à l'acquisition de données multicapteur.	26
3.5	Schéma de l'acquisition synchronisée de données multicapteur.	27
3.6	Histogramme du décalage temporel entre les instants de datation des images télémétriques et des images vidéo.	28
3.7	Caractérisation de l'erreur de mesure du télémètre laser : (a) $r_m - r$ et l'estimée $\hat{e}_b(r_m)$, et (b) les résidus $r_m - r - \hat{e}_b(r_m)$	30
3.8	Simulation d'images télémétriques avec le robot (a) en arrêt et (b,c,d) en mouvement.	31
3.9	Evaluation visuelle de la correction du biais de mesure du télémètre laser.	37
3.10	Résultats expérimentaux obtenus par application des procédures de prétraitement des images télémétriques.	37

3.11	Résultats expérimentaux obtenus par application de la procédure de synchronisation des images $\mathcal{L}$ et $\mathcal{I}$ ($t_{\mathcal{L}} - t_{\mathcal{I}} = -25\text{ ms}$).	38
4.1	Exemple d'extraction de droites dans des images vidéo : (a) une image réelle et (b) les droites verticales extraites.	41
4.2	Différentes phases de la détection de contours verticaux dans une image vidéo. . . .	43
4.3	Extraction de droites verticales utilisant accumulateur	44
4.4	Illustration des indices d'intérêt pour la segmentation d'image télémétriques.	47
4.5	Détection de points de cassure par seuillage adaptatif.	48
4.6	Détection de points de cassure utilisant l'approche basée sur le filtre de Kalman. . .	52
4.7	Illustration des noyaux classiques d'extraction de droites.	58
4.8	Evaluation des noyaux d'extraction de droites : environnement de simulation.	62
4.9	Exemples de la détection de droites verticales : (a)-(c) bons résultats et (d)-(g) mise en défaut de la méthode.	65
4.10	Contre-exemple où les points de cassure forts ne représentent pas les extrémités des surfaces balayées.	67
4.11	Evaluation expérimentale des détecteurs de points de cassure.	69
4.12	Evaluation expérimentale de la méthode d'extraction de segments dans les images télémétriques : comparaison des résultats obtenus avec les différents noyaux.	70
4.13	Temps de calcul des algorithmes de segmentation d'images télémétriques utilisant les différents noyaux pour l'extraction de segments. Ces résultats sont présentés en (a) pour l'approche adaptative de détection de points de cassure, et en (b) pour le détecteur basé sur le filtre de Kalman.	71
5.1	Construction d'une carte locale par traitement d'indices télémétriques. Les ellipsoïdes d'incertitude ont été agrandis pour une meilleure visualisation.	76
5.2	Détermination des paramètres des demi-plans.	77
5.3	Détermination des paramètres des arêtes de type bord.	78
5.4	Détermination des paramètres des arêtes de type coin.	79
5.5	Fusion d'indices vidéo dans la construction de cartes locales	80
5.6	Initialisation de l'incertitude d'une arête photométrique	82
5.7	Superposition des structures de toutes les cartes locales extraites durant une expérience suivant un trajet fermé. Les positions du robot correspondantes aux instants d'acquisition d'images comme aussi les images télémétriques sont aussi bien illustrés	85
5.8	Cas de figure concernant les arêtes photométriques : (a)-(b) cas montrant l'intérêt d'associer un sous-type aux arêtes ; (c)-(d) cas qui illustre les observations multiples.	87

5.9	Temps de calcul utilisé pour la segmentation des données multicapteur et la construction des cartes locales dans l'expérience précédente.	88
5.10	Contre-exemple de la détection d'arêtes photométriques.	89
6.1	Variables de configuration du robot Omni	93
6.2	Validation de l'identification de paramètres géométriques : écarts de position $X \times Y$ et d'orientation θ de l'odométrie en utilisant les paramètres mesurés λ_0 (courbes en -o-) et les paramètres estimés $\hat{\lambda}$ (courbes en -●-).	97
6.3	Cas de données incomplètes	109
6.4	Temps de calcul des différentes approches de localisation.	113
6.5	Etude de cas pour <i>MCP</i> Incomplet (partie I).	116
6.6	Etude de cas pour <i>MCP</i> Incomplet (partie II).	117
6.7	Conditions d'expérimentation dans le site <i>Premier étage extension - B</i>	119
6.8	Écarts et intervalles de confiance $\pm 3\sigma$ de localisation dans le site <i>Premier étage extension - B</i> (odométrie calibrée).	121
6.9	Conditions d'expérimentation dans le site <i>Couloir robotique - B</i>	122
6.10	Écarts et intervalles de confiance $\pm 3\sigma$ de localisation dans le site <i>Couloir robotique - B</i> (odométrie calibrée).	124
7.1	Détermination des paramètres géométriques pour la fusion de demi-plans.	130
7.2	Exemple de la fusion de structures.	131
7.3	Exemple de la prise en compte de contraintes dans la fusion de structures.	136
7.4	Environnement de simulation de l'Exemple 7.3	139
7.5	Exemple de l'évolution des arêtes en fonction de leur observation locale.	140
7.6	Diagramme de transition du type d'une arête globale (cercles) lors de la fusion avec des observations locales (arcs).	141
7.7	Détermination de la région d'espace libre (<i>REL</i>) associée à un demi-plan local.	141
7.8	Exemples d'application de l'étape de maintenance de l'espace libre.	143
7.9	Exemple de la fusion de ressemblances : (a) configuration originale, (b) carte globale mise à jour sans fusion de ressemblances, (c) structures ressemblantes et (d) leur fusion	144
7.10	Environnement de simulation	145
7.11	Résultats de simulation de la construction de cartes globales sous contraintes	148
7.12	Carte globale du site <i>Premier étage extension - B</i>	149

7.13	Résultats expérimentaux obtenus pour la cartographie du site <i>Premier étage extension - B.</i>	151
7.14	Carte globale du site <i>Halle mécatronique.</i>	152
7.15	Résultats expérimentaux obtenus pour la cartographie du site <i>Halle mécatronique.</i>	152
A.1	Repères principaux du robot Omni.	174
A.2	Positionnement entre repères.	175
B.1	Fonctions qui caractérisent la distribution gaussienne.	178
B.2	Ellipses de covariance de la distribution gaussienne bi-variable.	180
B.3	Exemple d'application de la méthode de propagation d'incertitude par linéarisation.	183
B.4	Exemple d'application de la méthode de propagation d'incertitude par échantillonnage sélectif.	185
C.1	Primitives géométriques dans l'espace 2-D.	188
D.1	Structures cartographiques.	194
D.2	Critères de correspondance géométrique entre demi-plans.	197
D.3	Simulation d'une carte.	199
D.4	Exemple réel de la mise en correspondance de cartes d'environnement.	200
E.1	Exemple d'utilisation des filtres stochastiques pour la fusion de données.	209
F.1	(a) Configuration simplifiée des repères du système multicapteur. (b) Projection des repères dans le plan de balayage laser.	216
F.2	Résultats de simulation de la transformation $\phi = g(u)$ pour différentes configurations $(x_C^L, y_C^L, \theta_C^L)$ du positionnement inter-capteurs.	218
F.3	Exemple de paire d'images utilisée pour extraire les mesures $\{\phi_m, u\}$ de la procédure d'étalonnage du système multicapteur. (a) Image vidéo et colonnes correspondantes à u (tracés verticaux). (b) Image télémétrique et directions des mesures ϕ_m (tracés pointillés).	220
F.4	Données d'étalonnage du système multicapteur : (a) les mesures $\phi_m \times u$ et de la transformation estimée $\hat{g}(u)$, et (b) les résidus $\phi_m - \hat{g}(u)$	221

Liste des tableaux

3.1	Paramètres du télémètre laser fournis par le fabricant IBEO GmbH.	23
4.1	Valeurs de $\bar{N}_l$, $\bar{\eta}_l$, $\bar{\varepsilon}_\rho$, et $\bar{\varepsilon}_\alpha$ pour la simulation avec $\sigma_r = 0,03 \text{ m}$	63
4.2	Valeurs de $\bar{N}_l$, $\bar{\eta}_l$, $\bar{\varepsilon}_\rho$, et $\bar{\varepsilon}_\alpha$ pour la simulation avec $\sigma_r = 0,05 \text{ m}$	64
7.1	Listes de correspondances	137
H.1	Mesures statistiques de localisation dans le site <i>Premier étage extension</i> - <i>B</i> et estime par odométrie calibrée.	232
H.2	Mesures statistiques de localisation dans le site <i>Premier étage extension</i> - <i>B</i> et estime par odométrie calibrée pessimiste.	232
H.3	Mesures statistiques de localisation dans le site <i>Premier étage extension</i> - <i>B</i> et estime par odométrie calibrée optimiste.	232
H.4	Mesures statistiques de localisation dans le site <i>Couloir robotique</i> - <i>B</i> et estime par odométrie calibrée.	233
H.5	Mesures statistiques de localisation dans le site <i>Couloir robotique</i> - <i>B</i> et estime par odométrie calibrée pessimiste.	233
H.6	Mesures statistiques de localisation dans le site <i>Couloir robotique</i> - <i>B</i> et estime par odométrie calibrée optimiste.	233

Introduction générale

1.1 Définition du problème

La recherche de techniques fiables de cartographie d'environnement et de localisation est fondamentale pour la conception de robots mobiles autonomes. La cartographie automatique permet la construction de cartes d'environnement adaptées à la localisation. Dans un robot dit autonome, la cartographie et la localisation servent de support à la planification et au suivi de chemins. Une recherche bibliographique nous a permis d'identifier un nombre très important d'approches pour la cartographie et la localisation. Plus spécifiquement, dans notre étude nous nous intéressons à la représentation 2-D d'environnements d'intérieur, sous forme d'une carte globale $\mathcal{M}^G$ composée de structures géométriques. Cette représentation est généralement construite automatiquement à partir de données acquises au cours d'une mission d'exploration. Lors de la réalisation de tâches de navigation globales (*e.g.*, aller de la position $\mathcal{A}$ à la position $\mathcal{B}$, représentées dans un repère absolu), les structures de $\mathcal{M}^G$ peuvent être mises à jour pour incorporer des changements récents dans l'environnement.

Considérons un robot mobile qui évolue sur sol plan dans un milieu d'intérieur, conformément à l'illustration de la Figure 1.1. Nous avons dans cette figure deux repères 2-D d'intérêt : le repère global $\mathcal{R}^G : \mathcal{X}^G \times \mathcal{Y}^G$, attaché à l'environnement, et le repère robot (ou repère local) $\mathcal{R}^R : \mathcal{X}^R \times \mathcal{Y}^R$ attaché au robot. Ces repères sont reliés par $\mathbf{z} = (x, y, \theta)^T$, où (x, y) sont les coordonnées de l'origine de $\mathcal{R}^R$ dans le repère $\mathcal{R}^G$, et θ est l'angle entre les axes $\mathcal{X}^R$ et $\mathcal{X}^G$. Donc, $\mathbf{z}$ définit la position du robot dans l'environnement. Pour l'instant, nous considérons la localisation et la cartographie comme des problèmes séparés, définis de la façon suivante :

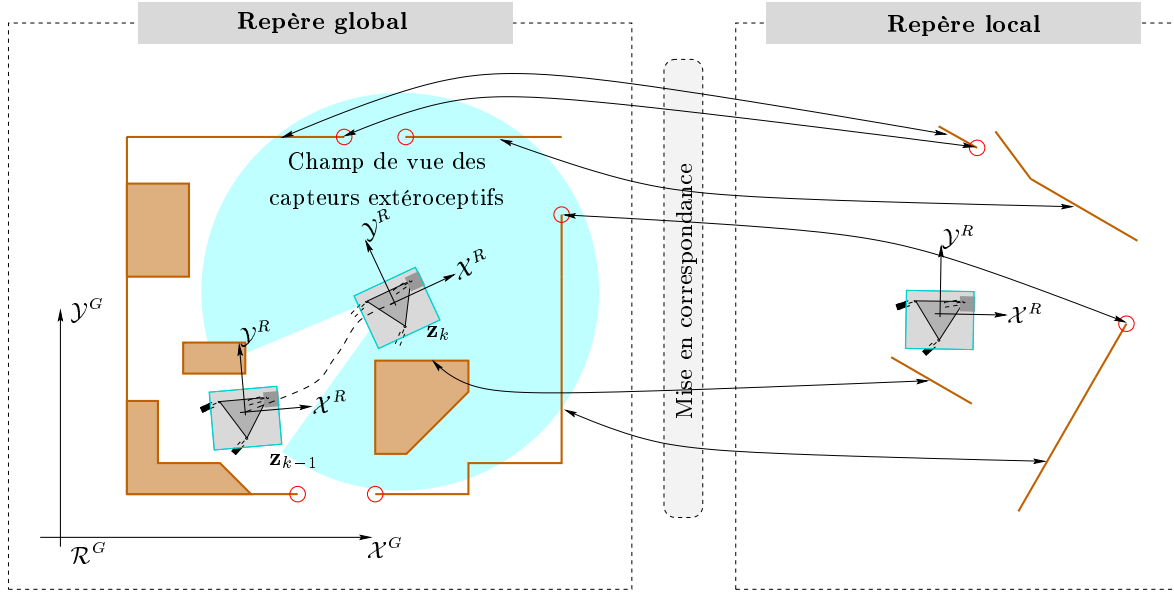


FIG. 1.1 – Localisation d'un robot mobile par mise en correspondance de cartes d'environnement

Localisation. Le problème de la localisation consiste en mettre à la disposition du système de navigation du robot une estimée $\hat{\mathbf{z}}$ consistante de $\mathbf{z}$ à chaque instant. Cela se fait à partir des mesures des *capteurs proprioceptifs* et des *capteurs extéroceptifs*. Les capteurs proprioceptifs fournissent des mesures du mouvement propre du robot (*e.g.*, des vitesses, des accélérations), et présentent en général une fréquence d'échantillonnage importante. Les capteurs extéroceptifs permettent d'acquérir des informations locales sur l'environnement autour du robot. On distingue deux modes de localisation :

- *Localisation relative à l'estime* : Dans la localisation relative, une estimée initiale de la position $\hat{\mathbf{z}}_{k-1}$ est intégrée à chaque pas d'échantillonnage en utilisant des mesures fournies par les capteurs proprioceptifs. L'intégration repose sur le modèle cinématique du robot. Les imprécisions de ce modèle et l'incertitude des mesures entraînent une croissance sans limite de l'erreur de positionnement ;
- *Localisation par mise en correspondance de cartes géométriques (recalage)* : Les capteurs extéroceptifs permettent de construire une représentation locale de l'environnement lorsque le robot se trouve à la position $\mathbf{z}_k$ (cf. Figure 1.1). Cette représentation, ou carte locale $\mathcal{M}^R$, possède un certain nombre de structures communes avec la carte globale $\mathcal{M}^G$. Ces structures, dites en *correspondance*, sont sensées représenter le même objet de l'environnement réel. Les correspondances étant reliées entre elles par des fonctions analytiques de la position du robot, cela nous permet d'estimer $\mathbf{z}_k$.

Une des limitations de la localisation par recalage est que la fréquence d'estimation que nous pouvons atteindre avec cette méthode est en général bien inférieure à celle de la localisation à l'estime. Cela a conduit à des solutions hybrides, où les deux modalités de localisation s'alternent. Ainsi, entre deux localisations par recalage aux instants discrets $k-1$ et k , l'estimée $\hat{\mathbf{z}}_{k-1}$ est mise à jour par estime.

Cartographie : Le problème de la cartographie consiste alors en mettre à jour la carte $\mathcal{M}^G$ à partir des observations locales $\mathcal{M}^R$ à la position $\mathbf{z}$. Malgré la simplicité apparente du problème, la cartographie est loin d'être un problème facile à résoudre.

1.2 Principales difficultés

Dans le contexte de cette thèse, nous nous sommes intéressés à des aspects théoriques et pratiques de la cartographie et de la localisation, en portant notre attention sur un certain nombre de difficultés. Les problèmes suivants ont été pris en considération :

- La cartographie et la localisation sont fortement liées au système sensoriel utilisé pour la capture d'informations concernant l'environnement. Ces capteurs étant sujets à des bruits de mesure, il est nécessaire de modéliser leur incertitude. Il faut ensuite mettre en œuvre des outils mathématiques appropriés pour propager cette incertitude à tous les niveaux du processus de localisation ;
- La localisation joue un rôle essentiel pour la cartographie et vice-versa. Un échec de la localisation peut engendrer des erreurs irrécupérables pour la cartographie. Alors, la carte $\mathcal{M}^G$ n'est plus cohérente avec l'environnement réel. De la même façon, une mise à jour erronée de la carte globale a des effets négatifs sur la localisation qui utilise cette carte ;
- Des erreurs de construction de la carte locale $\mathcal{M}^R$ peuvent se propager tant vers l'estimation de position que vers la mise à jour de la carte globale $\mathcal{M}^G$. A titre d'exemple, citons deux sources d'erreur agissant sur $\mathcal{M}^R$: l'incohérence des données extéroceptives avec l'environnement local, et l'extraction de fausses structures géométriques lors de la segmentation de ces données.

1.3 Plate-forme mobile Omni

Les évaluations expérimentales menées dans notre étude ont été faites sur une plate-forme mobile, le robot Omni, illustré sur la Figure 1.2. La conception mécanique de ce robot a été réalisée pendant la thèse de Le Corre [134][133], au LIRMM. Cependant, cette plate-forme a subi un certain nombre de modifications mécaniques et électroniques, avant d'être mise à notre disposition.

Omni est un robot omnidirectionnel, doté de trois roues motrices et directrices, qui lui permettent de naviguer facilement dans des environnements encombrés. Son architecture informatique est composée d'un ordinateur central Pentium II 300 MHz, et de cartes *transputer* qui servent d'interface entre l'ordinateur et certains capteurs. Son architecture logicielle [29] a été développée sous *Windows NT*®¹, complété de l'extension temps réel *RTX*®².

Cette plate-forme est dotée des capteurs suivants :

- **Capteurs proprioceptifs** :

¹Windows NT est une marque enregistrée de Microsoft, Inc.

²RTX est une marque enregistrée de Venturcom, Inc.

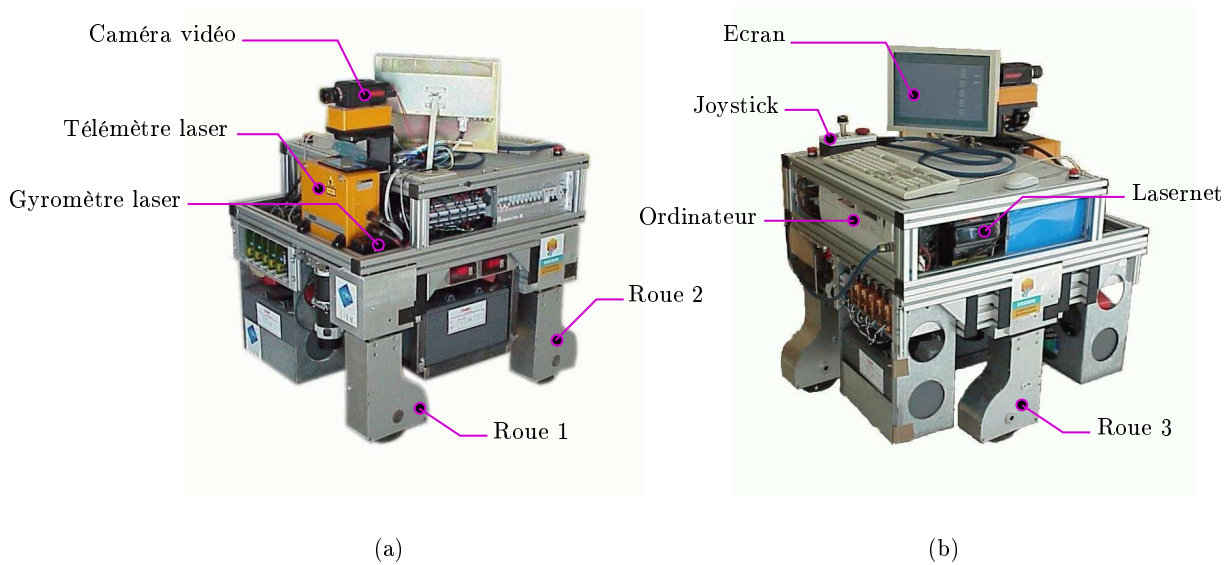


FIG. 1.2 – Robot Omni.

- trois codeurs absolus, utilisés pour mesurer les angles de direction de chaque roue ;
- six codeurs incrémentaux, utilisés pour mesurer les rotations de chacun des axes ;
- un gyromètre laser à fibre optique, qui fournit des mesures précises de variation du cap.
- **Capteurs extéroceptifs** :
 - une caméra vidéo en noir et blanc ;
 - un télémètre laser à balayage horizontal, pour capturer des images de profondeur 2-D de l'environnement ;
 - un goniomètre laser (lasernet), utilisé pour la détection de balises artificielles codées [2]. Cet instrument n'a pas été utilisé dans notre étude.

Une thèse, préparée actuellement par Julien Aragonès, concerne la commande de ce robot. Les algorithmes développés permettent au robot de suivre une trajectoire définie par une courbe de Bézier en utilisant des points de contrôle [5]. Cette méthode permet de considérer des estimées incertaines de position absolue du robot et de corriger la trajectoire tout en préservant la continuité du mouvement. Son intégration avec le système de localisation autonome développé dans cette thèse permet aujourd'hui au robot Omni de se déplacer dans des environnements déjà cartographiés.

1.4 Architecture de localisation et de cartographie

L'architecture de notre système de localisation et de cartographie est présentée sur la Figure 1.3. En fait, cette architecture est semblable à celle de tout système de localisation utilisant des cartes géométriques. Son originalité réside dans le contenu des différents modules :

- *Module de localisation relative*, où nous avons employé deux techniques de localisation : (i) l'odométrie, qui utilise seulement les données codeurs, et (ii) la gyrodométrie qui associe les

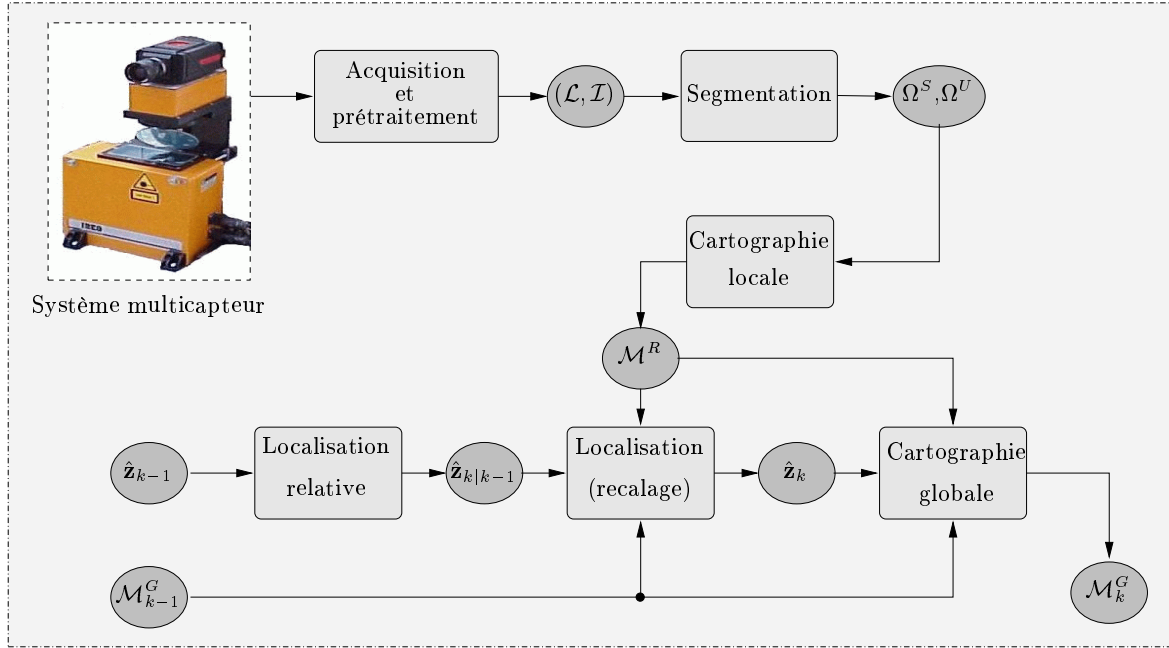


FIG. 1.3 – Schéma général de l'architecture de localisation et de cartographie.

mesures du gyromètre laser à l'odométrie ;

- *Module d'acquisition et de pré-traitement de données multicapteur*, qui réalise l'acquisition des données de la caméra vidéo et du télémètre laser, représentées respectivement par les structures $\mathcal{L}$ et $\mathcal{I}$. Avant d'être utilisées, ces données subissent un pré-traitement qui (i) compense la déformation de $\mathcal{L}$ due au mouvement du robot et (ii) réalise une pseudo-synchronisation entre $\mathcal{L}$ et $\mathcal{I}$. Ces procédures sont nécessaires pour que ces données soient cohérentes avec l'environnement local ;
- *Module de segmentation*, qui crée des listes Ω^S et Ω^U contenant les primitives géométriques extraites des données multicapteur. A ce stade, la représentation des primitives incorpore déjà des mesures d'incertitude ;
- *Module de cartographie locale*, qui génère la carte locale $\mathcal{M}^R$ par fusion de primitives des listes Ω^S et Ω^U ;
- *Module de localisation par recalage*, qui peut utiliser différentes techniques pour l'estimation de position à partir des structures en correspondance entre les cartes $\mathcal{M}^R$ et $\mathcal{M}_{k-1}^G$ (carte globale mise à jour dans le cycle de cartographie précédent). Comme un certain temps s'est écoulé depuis l'acquisition de données multicapteur, l'estimée de position est actualisée avant d'être mise à la disposition du système de navigation du robot ;
- *Module de cartographie globale*, où la carte globale $\mathcal{M}_{k-1}^G$ est mise à jour à partir de la carte locale $\mathcal{M}^R$ et de l'estimation récente de la position, pour donner $\mathcal{M}_k^G$.

Nous remarquons dans cette architecture que la cartographie et la localisation sont des modules indépendants. Cela implique que, contrairement au cas des approches couplées [52] [61][71][141], nous ne maintenons pas les corrélations entre la position du robot et les structures de la carte

globale.

Des validations expérimentales extensives ont été menées dans des environnements complexes (*i.e.*, encombrés, avec obstacles mobiles et conditions d'illumination variables, etc.). Notre objectif étant la vérification de la précision et de la robustesse des approches proposées, nous avons utilisé les estimations de localisation absolue fournies par la gyrodométrie comme vérité terrain. De plus, toutes les fonctionnalités de localisation et de cartographie présentées dans ce manuscrit sont aussi intégrées sur le robot Omni [30].

1.5 Contributions

Notre étude s'est focalisée sur les objectifs suivants :

- I. Proposition d'une architecture de localisation dynamique [33]. Dans cette architecture, des procédures spécialisées permettent :
 - de minimiser l'influence du mouvement du robot dans la déformation des images télémétriques ;
 - de compenser l'incohérence temporelle entre une image télémétrique et une image vidéo acquises de façon asynchrone par le système multicapteur ;
 - de ramener l'estimée de position dans le temps courant $k + \Delta T_l$, où ΔT_l est le temps écoulé entre l'acquisition de données et la fin des procédures de localisation.
- II. Conception d'un nouvel algorithme d'extraction de droites dans des images de profondeur 2-D [32] ;
- III. Etude d'une solution optimale et robuste au problème de la localisation de robots mobiles par mise en correspondance d'indices géométriques [31][35] [36][37] ;
- IV. Conception et implantation d'une approche découplée pour la construction de cartes d'environnements [34]. Cette approche s'appuie par des outils mathématiques récents pour assurer la consistance stochastique et la cohérence des structures de la carte, tout en préservant une complexité de calcul réduite.

1.6 Plan du manuscrit

Cette thèse est organisée de la façon suivante. Le chapitre 2 présente un état de l'art des principales solutions apportées aux problèmes de la cartographie et de la localisation des robots mobiles dans des environnements d'intérieur. Nous décrivons brièvement les principales méthodologies développées, en mettant en lumière leurs avantages et leurs inconvénients. Cependant, une analyse bibliographique plus approfondie des outils mathématiques exploités dans ce travail est donnée dans les chapitres suivants. Dans le chapitre 3 nous présentons le processus d'acquisition de données du système multicapteur, ainsi que des procédés de prétraitement de ces données. La segmentation d'images capteur pour l'extraction de primitives géométriques est traitée dans le chapitre 4. Le chapitre 5 présente la cartographie locale, c'est à dire l'ensemble des algorithmes

employés pour la construction de cartes géométriques locales à partir des primitives géométriques fournies par la segmentation. Le chapitre 6 présente nos approches de localisation dynamique robuste, qui sont comparées expérimentalement à des approches classiques par filtrage stochastique. Notre méthodologie découplée de cartographie d'environnement est décrite dans le chapitre 7.

Des très nombreux exemples et résultats expérimentaux illustrent les chapitres 3-7. Nous avons aussi effectué des comparaisons avec d'autres approches existantes. Les conclusions et les perspectives de ce travail sont présentées dans le chapitre 8. Nous donnons en annexe l'ensemble des outils, procédés et données expérimentales qui ont servi de support à ce travail.

Etat de l'art : méthodes de cartographie et de localisation par cartes géométriques

2.1 Introduction

Dans le milieu industriel, des véhicules automatiques ou semi-automatiques sont utilisés avec succès pour le transport de matériaux. Les techniques de navigation employées requièrent l'installation de balises artificielles [2][166] ou l'implantation de systèmes de guidage au sol [38][156]. Néanmoins, compte tenu de l'expansion des domaines d'application de la robotique de service (*e.g.*, surveillance, transport, domotique), de la robotique en milieu hostile (*e.g.*, nucléaire, sous-marin, spatial), ou encore de la robotique ludique, la conception de systèmes de navigation plus autonomes, compatibles avec des environnements complexes, mal connus *a priori* et évolutifs, devient nécessaire. Le robot doit pouvoir choisir et adapter en ligne sa trajectoire, de façon à réaliser des tâches plus complexes de navigation.

Dans ce cas, la construction et la mise à jour d'une représentation précise de l'environnement est indispensable, notamment sous forme d'une carte. Si l'on exclut les modèles CAO qui sont des représentations figées et pas nécessairement exactes de l'environnement, les représentations réalistes sont construites par le biais d'outils mathématiques à partir de données extéroceptives acquises généralement par les capteurs du robot. Ces représentations ou cartes appartiennent à deux catégories distinctes [206] :

- *Les représentations topologiques*, qui maintiennent des relations entre les différentes structures de l'environnement [58][126][127]. Ces cartes sont généralement constituées d'un graphe, où les nœuds correspondent aux structures géométriques (*e.g.*, lignes, points) ou symboliques (*e.g.*, portes, murs), et les connections indiquent les trajets (graphe de Voronoï [58]) ou des actions (cartes cognitives [126]) qui relient les zones d'observation des structures ;
- *Les représentations métriques*, qui préservent la configuration absolue des structures de la carte par rapport à un repère de référence. Les représentations métriques les plus courantes sont les grilles d'occupation [76][162][163], et les cartes stochastiques [165][195][196]. D'autres formalismes ont été proposés en utilisant des réseaux de neurones auto-organisés [124][102] ou des méthodes de régression en composantes principales [125][214].

On sait que, chez l'homme, la navigation se fait en utilisant une représentation topologique de l'environnement, *i.e.*, par cartes cognitives [151]. Etant donné qu'elles utilisent des relations entre les structures, les représentations topologiques ne requièrent pas la connaissance de leur emplacement absolu dans l'environnement. La localisation se traduit pourtant par un ensemble de procédures de reconnaissance de lieux. Cela n'est pas le cas pour la représentation métrique où un recalage des structures dans le repère de référence est nécessaire. Les structures étant détectées par des capteurs installés sur le robot (repère local), pour effectuer leur recalage on a besoin de connaître la position du véhicule dans le repère de référence (repère global). Lorsqu'on s'intéresse simultanément à la mise à jour de la représentation de l'environnement et à la localisation du robot, on est confronté au problème de *cartographie et localisation concurrentes* (*CLC*). Dans d'autres cas, la localisation peut être aussi traitée comme un problème à part, si une carte d'environnement est fournie au préalable.

A titre d'illustration, nous présentons sur la Figure 2.1 une grille d'occupation et une carte stochastique géométrique du site *Halle robotique - A*, que nous avons construites à partir des mesures présentées dans l'annexe G.2. La grille d'occupation (Figure 2.1(a)) est en fait une répartition de l'environnement en cellules à chacune desquelles est associée une probabilité d'occupation. Ce modèle donne une image de l'espace libre dont la résolution dépend de la taille des cellules, et est bien adapté au problème de la génération de trajectoires et de la navigation dans des environnements non structurés. Cela n'est pas le cas avec la carte géométrique illustrée sur la Figure 2.1(b) où seules les frontières des obstacles sont représentées sous forme de structures géométriques.

De très nombreuses solutions ont été proposées pour résoudre le problème de la localisation et de la *CLC* avec une grille d'occupation. Nous citons par exemple les travaux menés par les équipes de Thrun, Fox et Burgard [44][79][202][203], qui ont abouti à un impressionnant système de navigation implanté sur les robots RHINO [42] et MINERVA [200]. Etant donné la complexité de l'estimation bayésienne, l'utilisation de ces méthodes en temps réel, dans un processus de localisation, nécessite la mise en œuvre de mécanismes de filtrage de données et de sélection de cellules [42][43]. Pour la cartographie, les auteurs annoncent des temps inférieurs à une heure [203]. D'autres travaux ont porté sur la fusion de cartes topologiques et de grilles d'occupation pour l'aide à la navigation [201][206]. Des travaux comparatifs ont été menés par Raschke et Borenstein [180], Schiele et Crowley [186], et Gutmann *et al.* [87]. Une version floue des grilles d'occupation a été proposée par Oriolo *et al.* [172] [173].

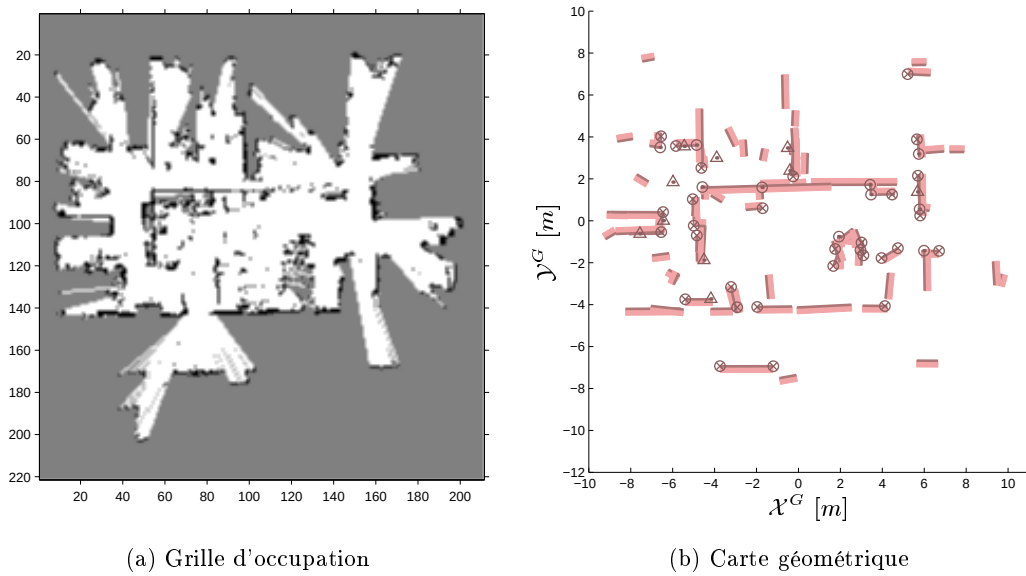


FIG. 2.1 – Représentations d'environnement.

Nous avons focalisé cet état de l'art sur les méthodes de localisation et de cartographie directement liées au sujet de notre étude, c'est à dire utilisant les cartes géométriques. Avant de continuer, nous faisons un bref rappel sur les notations. Dans la section 1.1, nous avons introduit les deux instances principales des cartes géométriques 2-D : la carte globale $\mathcal{M}^G$ associée au repère global, et la carte locale $\mathcal{M}^R$, attachée au repère robot. Ces cartes sont constituées de primitives géométriques qui sont généralement des segments de droite et/ou des points. Nous avons aussi défini $\hat{\mathbf{z}}$ comme l'estimée de la configuration $\mathbf{z} = (x, y, \theta)^T$ du robot dans le repère global.

Ainsi, nous commençons par une présentation des principales méthodes pour l'extraction de primitives géométriques, qui servent de support à la construction de cartes locales d'environnement. Ensuite, nous abordons le problème de la localisation seule, suivie des approches pour la cartographie et localisation concurrentes.

2.2 Segmentation de données et cartographie locale

Les méthodes d'extraction de primitives géométriques sont plus au moins adaptées au type de capteur utilisé pour percevoir l'environnement. Pour les véhicules navigant sur sol plan, il s'agit généralement de capteurs de distance tels que les télémètres laser à balayage 2-D, ou les ceintures de capteurs à ultrason montées sur la périphérie du robot. Quelques travaux utilisent également des indices extraits d'images vidéo.

Certaines méthodes de segmentation d'images de distance ont été publiées dans des travaux concernant la localisation, la segmentation étant traitée en deuxième plan. Par exemple, pour exploiter des mesures fournies par une ceinture de capteurs à ultrason, [6], [98] et [110] présentent

des heuristiques pour détecter des obstacles présentant des formes géométriques caractéristiques (bords, coins, murs). Le même genre de capteur est utilisé par Leonard *et al.* [141]. Rives *et al.* [184] utilisent des mesures fournies par deux capteurs à ultrason pour détecter à la fois la distance et l'angle d'orientation de surfaces balayées par les capteurs.

Un des avantages principaux d'un télémètre laser est son excellente résolution angulaire, bien supérieure à celle des capteurs à ultrason, qui permet d'acquérir un grand nombre de mesures de l'environnement autour du robot. Néanmoins, l'extraction de formes géométriques dans les images télémétriques n'est pas une tâche facile, même avec des techniques dites robustes. Plusieurs méthodes de segmentation utilisent la transformée de Hough [73][190] pour extraire des lignes infinies. Celles ci peuvent être ensuite délimitées de façon à obtenir des segments de droite, comme dans [8] [78][107]. Néanmoins, des approches moins robustes sont également utilisées pour l'extraction de droites [51][73][75][191][210][227]. Nous avons proposé dans [32] une approche basée sur les C-moyennes floues prototypées [123]. Une méthode d'extraction de segments de droites dans des images acquises en mouvement est présentée par Pears [176]. La prise en compte de mouvement lors de l'acquisition est faite par un filtre stochastique. Des primitives de type point extraites d'images télémétriques sont exploitées dans [51] et [193]. L'extraction de primitives de formes moins conventionnelles est le côté original de [128] et [194] qui, en plus des primitives géométriques classiques, ont inclus des groupements de mesures correspondant à des artefacts exploitables pour la localisation.

Parmi les travaux basés sur l'exploitation d'images vidéo, on peut citer [8], [50] et [161] qui utilisent des segments de droites correspondant à des contours verticaux. Ces indices sont assez intéressants lorsqu'on travaille dans l'espace de navigation 2-D. Par contre, des points de l'espace 3D fournis par un système de stéréovision sont plus intéressants pour la localisation 3-D [189].

La majeure partie des travaux que nous venons de citer ne s'intéressent pas à une représentation explicite de la géométrie de l'environnement local. Par exemple, dans [8], les primitives géométriques extraites d'une part par un télémètre laser et d'autre part par une caméra vidéo sont utilisées comme des mesures indépendantes les unes des autres par le système de localisation. Néanmoins, certaines approches construisent formellement une carte locale. C'est souvent le cas lorsque les mesures sont fournies par un système multicapteur. C'est le cas des travaux de Neira *et al.* [170] et Castellanos *et al.* [50]. Leur travail, ainsi que celui de Bulata et Devy [41], se caractérisent principalement par une modélisation haut niveau des primitives qui conduit à identifier des *structures cartographiques* (cf. annexe D.1).

2.3 Localisation

Les approches présentées ici sont classées en fonction du type de problème auquel elles sont destinées : l'autolocalisation, la localisation à l'estime et la localisation par mise en correspondance de cartes géométriques. Ces deux derniers problèmes sont traités dans ce manuscrit dans la section 6. Pour cela, nous n'entrons pas dans le détail des algorithmes. Une présentation plus étendue des capteurs et des méthodes peut être trouvée dans [1][26].

2.3.1 Autolocalisation

L'autolocalisation est un problème primordial pour la navigation référencée environnement. Il concerne l'ensemble des procédures utilisées pour déterminer la position du robot sans connaissance *a priori* de $\mathbf{z}$. On rencontre ce problème dans deux cas : lors de l'initialisation, ou lorsque le robot se perd au cours de l'exécution d'une tâche (cf. section 1.1). Dans ces cas, on doit déterminer une estimée de la position du robot seulement à partir des mesures extéroceptives et de la carte globale $\mathcal{M}^G$.

Certaines des approches d'autolocalisation utilisent directement les mesures capteur, sans extraire de primitive géométrique. Une solution originale par analyse d'intervalles a été proposée par Lévêque *et al.* [146], et expérimentée avec des mesures sonar. L'ensemble des mesures effectuées à une position inconnue $\mathbf{z}$ constitue un domaine borné D , étant donné l'incertitude des capteurs. En disposant d'équations qui relient la position $\mathbf{z}$ à des mesures possibles de chaque indice de la carte $\mathcal{M}^G$, on obtient alors par analyse d'intervalles l'ensemble borné Z de positions qui a pu générer D . Une autre solution utilisant un arbre d'hypothèses pour le même genre de capteur a été présentée par Lim et Leonard [147]. Etant donné toutes les mesures disponibles à la position courante, on construit un arbre d'hypothèses obtenues à partir de toutes les combinaisons deux-à-deux des mesures de distance. Les hypothèses sont alors soumises à des contraintes de mesure, et regroupées dans l'espace de configuration en utilisant une technique de *clustering*. Les meilleures hypothèses sont alors données par les centres des groupements.

Ces approches, bien qu'appliquées à des mesures d'ultrasons, sont en fait adaptées à n'importe quel capteur de distance. Cependant, lorsque nous avons des capteurs de haute précision tel qu'un télémètre laser ou une caméra vidéo, les primitives géométriques extraites des mesures peuvent aussi être exploitées pour l'autolocalisation. Les approches adaptées à ce type d'information effectuent en général une mise en correspondance globale. Ceci dit, on cherche des primitives de la carte locale et de la carte globale correspondant à un même objet de l'environnement. Un nombre considérable de solutions a été proposé pour des robots équipés de caméras vidéo [11][138][171][198]. Par exemple, [62] utilise le formalisme de Dempster-Shafer pour estimer la position absolue à partir de correspondances de segments de droite déterminées en considérant des attributs invariants par rapport à la position d'observation. Mouaddib [164] propose une méthode de recherche exhaustive sur un arbre d'interprétation pour trouver des correspondances. Une réduction considérable de la longueur de l'arbre d'interprétation peut être obtenue en divisant la carte globale en régions de visibilité pour chaque primitive [192].

En utilisant des primitives extraites de données capteur dans le repère local, [12] et [183] emploient des techniques de filtrage multimodal pour maintenir des hypothèses sur la position absolue du robot. Au départ, les hypothèses sont construites par des combinaisons entre primitives capteur appariées avec celles de la carte globale. En fait, les hypothèses sont les modes de la distribution *a posteriori* de l'estimée $\hat{\mathbf{z}}$. Au fur et à mesure que le robot se déplace, les hypothèses moins fiables disparaissent, et d'autres fusionnent, de façon à ce qu'il ne reste que les hypothèses de vraisemblance maximale. L'idée de représenter la distribution de $\hat{\mathbf{z}}$ par un ensemble d'échantillons est la caractéristique essentielle des approches récentes de localisation de type Monte Carlo [69][106][109][204]. Elles utilisent des procédés simples d'échantillonnage d'importance pour mettre

à jour les échantillons lors de l'arrivée des mesures. Cependant, leur temps de convergence est assez important.

2.3.2 Localisation relative par estime

Sur les robots mobiles, l'estime permet d'actualiser $\hat{\mathbf{z}}$ par intégration du modèle cinématique. Ce modèle relie $\mathbf{q}$ et $\dot{\mathbf{q}}$, les paramètres de configuration des axes du robot, à $\dot{\mathbf{z}}$. Cette intégration se fait à des instants discrets, en utilisant des mesures de capteurs proprioceptifs pour approximer $\mathbf{q}$ et $\dot{\mathbf{q}}$.

La technique d'estime la plus courante pour les robots à roues est l'odométrie qui utilise seulement les mesures des codeurs incrémentaux. Borenstein et Feng [28] classent les différentes sources d'erreur agissant sur l'odométrie en (i) erreurs systématiques et (ii) erreurs non-systématiques. Les erreurs systématiques sont liées principalement à la connaissance approximative de λ (*i.e.*, les paramètres géométriques du robot), et à une modélisation cinématique approximative (*e.g.*, en ne considérant pas le non-alignement des axes). Les erreurs non-systématiques sont dues à des phénomènes extérieurs, comme par exemple la navigation sur des sols non plans, la collision avec des obstacles, le dérapage des roues, etc. La façon la plus répandue de représenter l'action de ces erreurs sur $\hat{\mathbf{z}}$ est de supposer que l'incertitude sur cette variable est de distribution gaussienne, de moyenne nulle et de covariance $\mathbf{\Lambda}_{\hat{\mathbf{z}}}$. Pour des robots à traction différentielle, il existe différents modèles plus ou moins sophistiqués pour mettre à jour $\mathbf{\Lambda}_{\hat{\mathbf{z}}}$ en même temps que $\hat{\mathbf{z}}$ [56] [216]. Pour les robots omnidirectionnels, de cinématique plus complexe, la solution la plus simple consiste à discrétiser le modèle cinématique par linéarisation au premier ordre, comme dans la section 6.2.2, car les stratégies de correction sont trop dépendantes de la forme du modèle cinématique [25].

La fusion avec des données proprioceptives fournies par d'autres capteurs tels que des gyroscopes [174][27], des compas magnétiques [26] ou des centrales inertielles [17] permet d'améliorer la précision de l'estime. Chung *et al.* [60] proposent une procédure de calibrage d'un gyromètre laser pour augmenter la précision. Von der Hardt *et al.* [215] étend cette démarche et propose l'identification en ligne de λ et des paramètres intrinsèques des capteurs. Pour améliorer l'estime d'un robot omnidirectionnel à roues suédoises (soumises à des glissements importants), Nagatani *et al.* [168] réalisent la fusion du déplacement calculé par odométrie avec celui correspondant au flot optique estimé à partir d'images vidéo.

2.3.3 Localisation par mise en correspondance de cartes géométriques

La localisation par mise en correspondance de cartes géométriques est souvent utilisée avec l'estime dans le cadre de la localisation hybride. Dans cette stratégie, l'estime assure la mise à jour de $\hat{\mathbf{z}}$ et $\mathbf{\Lambda}_{\hat{\mathbf{z}}}$ à des fréquences importantes. Même si les approches présentées dans la section 2.3.2 contribuent à réduire l'incertitude sur l'estime, elles ne pourront pas empêcher sa croissance au cours du temps. Ainsi, la localisation par mise en correspondance de cartes géométriques est employée pour corriger l'estimation de position. Pour cela, les méthodes présentées se servent soit de mesures extéroceptives brutes, soit de primitives géométriques extraites par segmentation de

ces mesures.

2.3.3.1 Méthodes utilisant des mesures extéroceptives brutes

Disposant de mesures télémétriques et d'une carte globale constituée de segments de droite, Cox [63] propose une approche itérative de correction de $\hat{\mathbf{z}}$. A chaque itération, les mesures télémétriques sont associées à des segments de la carte, et un facteur $\Delta\hat{\mathbf{z}}$ de correction est calculé de façon à minimiser les écarts d'association au sens des moindres carrés. Ce facteur est rajouté à $\hat{\mathbf{z}}$, et une nouvelle itération recommence jusqu'à ce que $\Delta\hat{\mathbf{z}}$ soit inférieur à un seuil. Gutmann *et al.* [88][89] ont suivi le même raisonnement et proposé des procédures semblables basées sur l'optimisation itérative. En fait, l'association des mesures étant effectuée à chaque itération, ces approches présentent une certaine robustesse vis à vis de la présence d'obstacles. Néanmoins, il existe toujours le risque que ces algorithmes restent bloqués dans des minima locaux. Une autre solution intéressante utilise un des premiers concepts pour la localisation par cartes, l'histogramme d'angles [220].

2.3.3.2 Méthodes utilisant primitives géométriques

Les méthodes de localisation de cette section cherchent à corriger (ou à estimer) la position du robot à partir de primitives géométriques correspondantes extraites du modèle global $\mathcal{M}^G$ et de la carte locale $\mathcal{M}^R$. Pour cela, il faut d'abord résoudre le problème de la mise en correspondance de ces primitives. Sans connaissance *a priori* sur la position du robot, même si elles sont aptes à résoudre ce problème, les approches d'autolocalisation présentées dans la section 2.3.1 présentent une complexité de calcul trop élevée. Une solution rapide proposée par Neira et Gonzalez [181] utilise la transformée de Hough pour calculer une signature des droites de l'environnement et estimer ensuite le cap du robot par une technique de vote. Cependant, il existe des travaux qui utilisent la transformée de Hough pour la segmentation seulement [8][78][107]. La mise en correspondance reste alors un problème local (cf. annexe D.5), que l'on résout en utilisant l'estime pour fournir par prédiction une estimée initiale de la position. Pour ces travaux, le recalage s'effectue à l'aide d'un filtre stochastique, le filtre de Kalman étendu (*FKE*).

En formulant le recalage comme un problème d'estimation stochastique non-linéaire, de très nombreux travaux ont été réalisés durant cette décennie. Nous avons ainsi des résultats en utilisant le *FKE* parallèle [40] [105][139], le *FKE* séquentiel [8][51][108][170][178]. [51], [170] et [178] utilisent la formulation information du filtre de Kalman. Le *FKE* séquentiel est la méthode la plus utilisée aujourd'hui pour résoudre le problème de la localisation seule, en raison des simplifications dans le processus de correction de position. Pour pouvoir estimer l'état de systèmes non-linéaires génériques, le *FKE* utilise des modèles linéarisés au premier ordre autour de la prédiction. Cela ajoute une difficulté supplémentaire au réglage de l'algorithme. En effet, des erreurs d'approximation peuvent avoir lieu à cause d'une mauvaise prédiction ou de l'approximation au premier ordre. Pour compenser ces erreurs, le *FKE* itératif [122] et le *FKE* adaptatif [111] ont été proposés. Néanmoins, d'autres complications, souvent liées aux distributions des différentes variables du modèle, pénalisent la formulation stochastique. Pour pallier les problèmes d'inconsistance d'estimation dus à

des corrélations non modélisables, Xu et Negahdaripour [223] appliquent la formulation étendue du filtre intersection de covariances [115]. Antoniali et Oriolo [4] proposent un filtre multi-hypothèses afin de fournir une estimation consistante en présence de formes multimodales de la densité de probabilité de l'estimée $\hat{\mathbf{z}}$, comme nous le verrons dans la section 6.3.3. Les approches récentes de localisation de type Monte Carlo, utilisées aussi pour l'autolocalisation, ont été évaluées dans [69][106][109] [204]. Ces approches ont fait preuve d'une certaine robustesse, d'une précision et d'une facilité d'implantation.

2.4 Cartographie et localisation concurrentes

La *cartographie et localisation concurrentes (CLC)*, désignée aussi par *cartographie et localisation simultanées*, est peut être le problème le plus connu de la recherche en robotique mobile [144]. Il est défini ainsi :

Partant d'une position connue, un robot mobile commence à naviguer dans un environnement initialement inconnu. Il utilise ses capteurs extéroceptifs pour acquérir des mesures sur l'environnement à des instants différents. A partir de ces mesures, il doit construire une représentation de l'environnement sous forme d'une carte globale $\mathcal{M}^G$. En même temps, il doit être capable de se localiser par rapport à la carte $\mathcal{M}^G$ de façon à pouvoir y intégrer les mesures extéroceptives représentées dans son repère local.

Une des premières solutions complètes au problème *CLC* est celle utilisant des cartes stochastiques, proposée par Smith, Self et Cheeseman [195][196]. Une carte stochastique est composée de N primitives géométriques, modélisées par des paramètres $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N$ connus dans le repère global. Les mesures effectuées par le robot étant représentées dans son repère local, elles sont fortement couplées à la position $\mathbf{z}$ du robot. Ainsi, dans le cadre de la *CLC* par cartes stochastiques, on constitue le vecteur d'état augmenté $\mathbf{x}^T = \begin{bmatrix} \mathbf{z}^T & \mathbf{p}_1^T & \mathbf{p}_2^T & \dots & \mathbf{p}_N^T \end{bmatrix}$. Le couplage entre les différentes composantes de ce vecteur d'état est représenté par sa matrice de covariance

$$\Lambda_{\mathbf{x}} = \begin{pmatrix} \Lambda_{\mathbf{z}} & \Lambda_{\mathbf{z},\mathbf{p}_1} & \dots & \Lambda_{\mathbf{z},\mathbf{p}_N} \\ \Lambda_{\mathbf{p}_1,\mathbf{z}} & \Lambda_{\mathbf{p}_1} & \dots & \Lambda_{\mathbf{p}_1,\mathbf{p}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{\mathbf{p}_N,\mathbf{z}} & \Lambda_{\mathbf{p}_N,\mathbf{p}_1} & \dots & \Lambda_{\mathbf{p}_N} \end{pmatrix},$$

où $\Lambda_{\mathbf{z}}, \Lambda_{\mathbf{p}_1}, \dots, \Lambda_{\mathbf{p}_N}$ sont les covariances des composantes de $\mathbf{x}$ et les termes extra diagonaux sont les inter-corrélations de ces variables. Etant donné la nature non-linéaire du problème, l'estimation de $\mathbf{x}$ et de sa matrice de covariance est obtenue de façon élégante et récursive par le filtre de Kalman étendu. Ce processus s'exécute à des instants discrets k , en deux phases, suivant le type de mesures disponibles [140] :

- La phase de prédiction par estime, qui calcule $\hat{\mathbf{x}}_{k|k-1}$ et $\Lambda_{\hat{\mathbf{x}}_{k|k-1}}$ à partir de mesures proprioceptives ;
- La phase de correction, où des primitives géométriques sont extraites des mesures extéroceptives. Ensuite, des correspondances sont établies avec les primitives de la carte globale, ce

qui permet de corriger la prédiction et d'obtenir les estimées $\hat{\mathbf{x}}_k$ et $\mathbf{\Lambda}_{\hat{\mathbf{x}}_k}$.

Bien entendu, dès que de nouvelles observations sont validées, elles sont rajoutées à la carte globale. Ainsi, la cartographie se fait de façon incrémentale. Par la suite, des nombreux travaux ont été réalisés suivant ce principe, comme par exemple [40][39][153] [154][49][61][66][77][141][182][169]. Dans [67], Davison et Murray ont appliqué la *CLC* avec un système de vision stéréoscopique. Ce travail utilise la motorisation du système capteur pour qu'il pointe dans la direction des primitives qui portent le plus d'information pour la localisation. Castellanos *et al.* [159] évaluent l'influence de l'utilisation de différents capteurs extéroceptifs. En réalité, la cartographie est confrontée à des multiples correspondances. Dans ces cas, négliger seulement une des correspondances peut biaiser le processus d'estimation. Avec un traitement indépendant des correspondances multiples, on peut faire croître de façon importante le nombre d'ambiguïtés dans la carte. Pour résoudre ce problème, Williams *et al.* [222] proposent de fusionner les primitives ambiguës de la carte, sans affecter les autres composantes qui sont reliées par les inter-corrélations.

Une des questions clefs de cette approche est l'importante complexité de calcul $O(N^3)$, due à la maintenance d'une matrice $\mathbf{\Lambda}_{\hat{\mathbf{x}}_k}$ de grande dimension. Leonard et Durrant-Whyte [140] proposent une stratégie de navigation qui permet d'éliminer les inter-corrélations de la carte. Cette stratégie requiert que le robot s'arrête, pour qu'un nombre important de mesures soit effectué au même endroit. Cela était justifié, étant donné la puissance de calcul disponible à l'époque, car la correction seule demandait quelques secondes [165]. Des travaux récents publiés par Victorino *et al.* [212][213] proposent une stratégie de commande basée sur capteur pour contraindre la navigation sur un graphe de Voronoï et améliorer ainsi la robustesse de l'ensemble localisation et cartographie. En ce qui concerne la complexité, Guivant et Nebot [86] l'ont réduit en simplifiant les équations de correction. Une solution efficace et élégante pour des environnements vastes est le découpage de la carte en sous-cartes ayant un minimum de recouvrement. Une telle solution a été proposée par Leonard et Feder [142][143]. Les auteurs présentent aussi une procédure pour garantir la consistance de l'état du véhicule lors du changement de carte.

Une étude expérimentale sur l'importance du maintien des inter-corrélations est présentée dans [52]. Les auteurs concluent que, sans considérer les inter-corrélations, on a des résultats optimistes sur l'estimée $\hat{\mathbf{x}}_k$; c'est à dire une sous-estimation de l'incertitude sur la position du robot et des primitives de la carte globale. Cela est maintenant un résultat connu et accepté par la communauté scientifique. Dans [158], les auteurs ne signalent pas de problème de divergence. Cela est peut être dû au fait que l'expérience réalisée dans [52] était de très courte durée (trajet de quatorze mètres). S'appuyant sur le formalisme de la *CLC* avec un filtre de Kalman étendu, Dissanayake *et al.* [71] montrent, qu'en théorie, la carte globale converge vers une carte relative. L'erreur finale de chaque primitive par rapport à l'environnement réel dépend de son incertitude lors de la première observation. Les auteurs sont convaincus de leur résultat. Néanmoins, nous avons quelques remarques à faire. Tout d'abord, la preuve est établie en supposant des modèles linéaires. Ainsi, ils arrivent à des conclusions connues depuis longtemps pour le filtre de Kalman linéaire [104], par exemple, que l'erreur d'estimation finale dépend de l'incertitude d'observation initiale. Les auteurs ne font aucune remarque sur les problèmes liés à la linéarisation qu'ils appliquent au filtre de Kalman, et qui peuvent biaiser l'estimée ou la faire diverger [104]. De plus, les expériences réalisées sont de courte

durée, dans un environnement réel de petite taille, et avec des balises bien espacées les unes des autres. Notre argumentation est confirmée par Julier et Uhlmann [116], qui démontrent de façon analytique un contre-exemple à la théorie de la *CLC* par filtre de Kalman étendu. Leur résultat est illustré par une simulation où la divergence est obtenue après un grand nombre de cycles de mise à jour. De plus, les auteurs affirment que des solutions de compensation parfois utilisées, comme l'augmentation artificielle des matrices d'incertitude [15], ne suffisent pas à garantir la convergence de la méthode. En conséquence, d'autres travaux ont été réalisés dans le but de préserver la consistance de la carte. Par exemple, [185] propose un filtre multi-hypothèses, et Chong et Kleeman [57] utilisent le filtre de Kalman étendu itératif et le *Julier-Uhlmann-Durrant-Whyte filter* [114][117].

Déjà en 1989, Moutarlier et Chatila [165] ont remarqué que la *CLC* par filtre de Kalman était très sujet à des divergences. Ils ont ainsi proposé une procédure connue sous le nom de recalage-fusion. Cette procédure corrige la prédiction de position en utilisant les correspondances, et ensuite elle met à jour la carte en utilisant l'estimée de position corrigée. D'autres approches découplent complètement le *CLC*, en traitant chaque structure de la carte de façon indépendante [131][132]. Cela revient à négliger les inter-corrélations et conduit à une carte d'environnement inconsistante si l'on utilise le filtre de Kalman étendu [52].

Dans des cas où des inter-corrélations existent entre les différentes variables d'un modèle linéaire, mais sont inconnues, le filtre intersection de covariances est capable de fournir une estimation consistante [115] (voir annexe E.3.2 pour un descriptif). Cette caractéristique fait de ce filtre un bon outil pour la cartographie découplée. Une version étendue (*i.e.*, pour des systèmes non-linéaires) est utilisée pour la *CLC* dans [207]. [91] propose une reformulation moins pessimiste de ce filtre. Néanmoins, les caractéristiques de consistance du filtre linéaire ne sont plus valables pour cette version étendue. En effet, à cause du processus de linéarisation, on peut encore propager de façon inconsistante les mesures d'incertitude (cf. exemple B.2).

Dans la suite de cette thèse (chapitre 7) nous proposerons une approche découplée au problème de cartographie et localisation concurrentes. Nous traiterons la localisation et la mise à jour de la carte séparément. La localisation s'effectue avec nos méthodes robustes décrites dans le chapitre 6. Pour la mise à jour, nous proposons une formulation utilisant seulement le filtre intersection de covariances linéaire. Cependant, nous avons vu dans cette section que la *CLC* est un problème avec des non-linéarités inhérentes. Ainsi, pour éviter la propagation inconsistante d'incertitudes, nous utilisons la technique d'échantillonnage sélectif [114] dans toutes opérations de changement de repère. De plus, nous avons exploité une représentation haut niveau des primitives de la carte, ce qui implique des contraintes entre certaines structures. Nous avons ainsi introduit une stratégie de propagation stochastique de ces contraintes. Des résultats expérimentaux illustrent les apports de notre méthode.

Un dernier point que nous voudrions soulever sur la *CLC* porte sur l'importance de la localisation dans ce problème. Si l'estimation de position est biaisée, n'importe quelle méthode présentée dans cette section peut diverger, y compris la nôtre. Cela peut être encore plus critique si le robot navigue à des vitesses importantes, lorsqu'un petit nombre d'observations successives peut ne pas suffire pour obtenir une estimée fiable d'une primitive. Ainsi, les méthodes présentées dans [72][84] peuvent être utilisées pour corriger la position du robot de façon plus globale, avant qu'elle soit

utilisée pour la cartographie.

Acquisition et prétraitement des données multisensorielles

3.1 Introduction

La construction de cartes locales est un processus de grande importance dans notre architecture de localisation par cartes géométriques. Ce processus a comme données des images de profondeur $\mathcal{L}$, acquises par le télémètre laser, et des images d'intensité $\mathcal{I}$ fournies par la caméra vidéo. L'acquisition de ces données n'est pas une tâche triviale lorsque la construction de cartes locales se fait avec des indices de haut niveau. Divers problèmes liés à l'acquisition doivent être pris en considération pour permettre ensuite une modélisation la plus exacte possible de l'environnement.

Ce chapitre présente les procédures d'acquisition et de prétraitement des données du système multicapteur. Ces procédures ont comme objectif principal d'assurer la cohérence spatiale et temporelle des images de profondeur et d'intensité. Après avoir présenté le système multicapteur, nous décrivons la méthodologie d'acquisition synchronisée de données. Cette méthodologie a comme objectif de pallier le décalage temporel entre les instants d'acquisition de $\mathcal{L}$ et $\mathcal{I}$. En ce qui concerne l'image télémétrique, nous proposons des solutions pour corriger le biais de mesure, la déformation de l'image due au mouvement du robot, et pour assurer leur cohérence temporelle. Les expériences d'évaluation menées avec des données réelles sont présentées dans la dernière section.

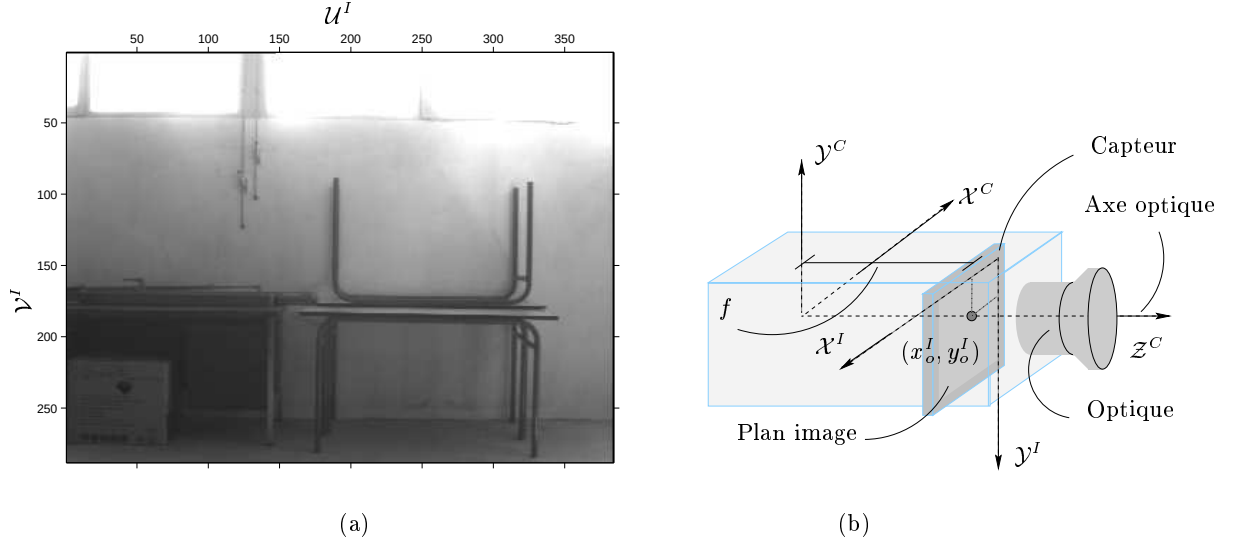


FIG. 3.1 – (a) Une image caméra et (b) les repères attachés à ce capteur.

3.2 Système multicapteur

Dans la section 1.3, nous avons introduit le robot Omni, doté de deux capteurs extéroceptifs destinés à la modélisation d'environnements : une caméra vidéo et un télémètre laser. La caméra vidéo fournit des images d'intensité en niveaux de gris (cf. Figure 3.1(a)) notées $\mathcal{I}$, telles que :

$$\mathcal{I} = \{p(u, v) | u = 1, \dots, M; v = 1, \dots, N\}. \quad (3.1)$$

Dans $\mathcal{I}$, $p(u, v)$ est une fonction discrète monotone croissante de l'énergie lumineuse reçue par la cellule de coordonnées (u, v) dans le *plan image* pendant la *période d'intégration* du capteur. Les cellules du plan image sont aussi appelées *pixels*. Au plan image est attaché un repère métrique 2-D $\mathcal{R}^I : \mathcal{X}^I \times \mathcal{Y}^I$. Les coordonnées métriques d'un pixel sont reliées à ses coordonnées image (u, v) par des constantes multiplicatives h_u et h_v . Ainsi, le pixel (u, v) dans l'image $\mathcal{I}$ a comme coordonnées métriques (x^I, y^I) , avec $x^I = h_u \cdot u$ et $y^I = h_v \cdot v$ dans l'espace Euclidien de $\mathcal{R}^I$.

Nous nous intéressons à la géométrie de ce capteur, décrite sur la Figure 3.1(b). Nous considérons un modèle géométrique simplifié de type *pin-hole* [101]. Nous supposons que l'axe optique de la caméra intercepte de façon orthogonale le plan image au point (x_o^I, y_o^I) . La distance entre (x_o^I, y_o^I) et le foyer du système optique est f , la *distance focale*. Le foyer est l'origine du repère métrique 3-D de la caméra : $\mathcal{R}^C : \mathcal{X}^C \times \mathcal{Y}^C \times \mathcal{Z}^C$. La Figure 3.1(b) montre la relation entre le repère image $\mathcal{R}^I$ et le repère métrique $\mathcal{R}^C$ attaché à la caméra. Les axes $\mathcal{X}^C$ et $\mathcal{X}^I$ sont parallèles et de sens contraires, comme le sont aussi $\mathcal{Y}^C$ et $\mathcal{Y}^I$.

Le télémètre laser fournit une image de profondeur 2-D qui représente le profil des obstacles autour du capteur dans le *plan de balayage* (Figure 3.2(a)). Cette image est formée par des mesures discrètes, nommées *points de balayage*. Le fonctionnement de ce capteur est décrit sur la Figure 3.2(b). Le système de mesure de distances est basé sur l'émission d'un train d'impulsions laser qui,

TAB. 3.1 – Paramètres du télémètre laser fournis par le fabricant IBEO GmbH.

Paramètre	Valeur fabricant	Description
$r_{\max}$	30 m	Portée maximale des mesures de distance du ladar.
$\phi_{\min}$	-135°	Angle minimum de balayage (programmable).
$\phi_{\max}$	135°	Angle maximum de balayage (programmable).
$\Delta\phi$	$0,6^\circ$	Résolution angulaire de ϕ (programmable)
T	0,125 s	Période d'un tour complet de 360° .
σ_r	0,03 m	Ecart-type des mesures de distance r .

quand il rencontre un obstacle, est réfléchi dans la direction d'émission. Le temps d'aller-retour des impulsions est mesuré par une horloge à très haute fréquence. En théorie, la distance r de l'obstacle au capteur est directement proportionnelle au temps de vol du train d'impulsions. En pratique, la lumière ambiante influence la mesure de distance [211].

Le balayage dans un plan se fait à l'aide d'un miroir rotatif incliné à 45° tournant à une vitesse constante dans le sens inverse des aiguilles d'une montre autour d'un axe orthogonal au plan de balayage $\mathcal{X}^L \times \mathcal{Y}^L$. L'axe $\mathcal{Z}^L$ du repère 3-D $\mathcal{R}^C : \mathcal{X}^L \times \mathcal{Y}^L \times \mathcal{Z}^L$ coïncide avec l'axe de rotation du miroir. Un codeur absolu permet de mesurer de façon assez précise l'angle correspondant à la direction de chaque mesure du capteur. Soit ϕ_n l'angle de balayage correspondant à la n -ième mesure de distance r_n . La première mesure est obtenue dans la direction $\phi_{\min}$, et les mesures consécutives sont effectuées avec un pas angulaire constant $\Delta\phi$ jusqu'à la dernière mesure dans la direction $\phi_{\max}$. L'ensemble des mesures télémétriques forment une image de profondeur 2-D notée :

$$\mathcal{L} = \{(r_n, \phi_n) | n = 1, \dots, N; \phi_1 = \phi_{\min}; \phi_N = \phi_{\max}; \phi_n - \phi_{n-1} = \Delta\phi\}. \quad (3.2)$$

Le champ angulaire du capteur est $\phi_{\max} - \phi_{\min}$. Chaque paire (r_n, ϕ_n) correspond aux coordonnées polaires du point d'intersection du faisceau laser avec la surface de l'obstacle dans la direction ϕ_n . Ainsi, (3.2) correspond à la représentation polaire d'une image de profondeur. Dans ce travail, nous considérons que les mesures angulaires ϕ_n sont exactes. Hélas, la même considération ne peut pas être appliquée aux mesures de distance r_n . Les erreurs de mesure étant fortement dépendantes de phénomènes physiques divers et de la résolution de l'horloge haute-fréquence du capteur, nous supposons que ces erreurs suivent une distribution $\mathcal{N}(0, \sigma_r^2)$. Pour l'instant nous admettons l'hypothèse d'une distribution de moyenne nulle. Dans la section 3.5.1 de ce mémoire, nous élargirons cette hypothèse en considérant le cas d'une distribution de moyenne variable, qui correspond au biais de mesure.

Un bref récapitulatif des paramètres du télémètre laser ainsi que de ses valeurs nominales est donné dans le Tableau 3.1.

Nous revenons maintenant à l'ensemble multicapteur du robot Omni, dont la photo est montrée sur la Figure 3.3(a). La caméra vidéo est fixée au dessus du télémètre laser. Cette configuration permet de centrer approximativement les champs des deux capteurs autour d'un axe vertical. Par ailleurs, elle simplifie la procédure d'étalonnage du système multicapteur proposée dans le cadre de cette thèse (cf. annexe F). En première approximation, nous supposons que les axes $\mathcal{Z}^L$ et $\mathcal{Y}^C$

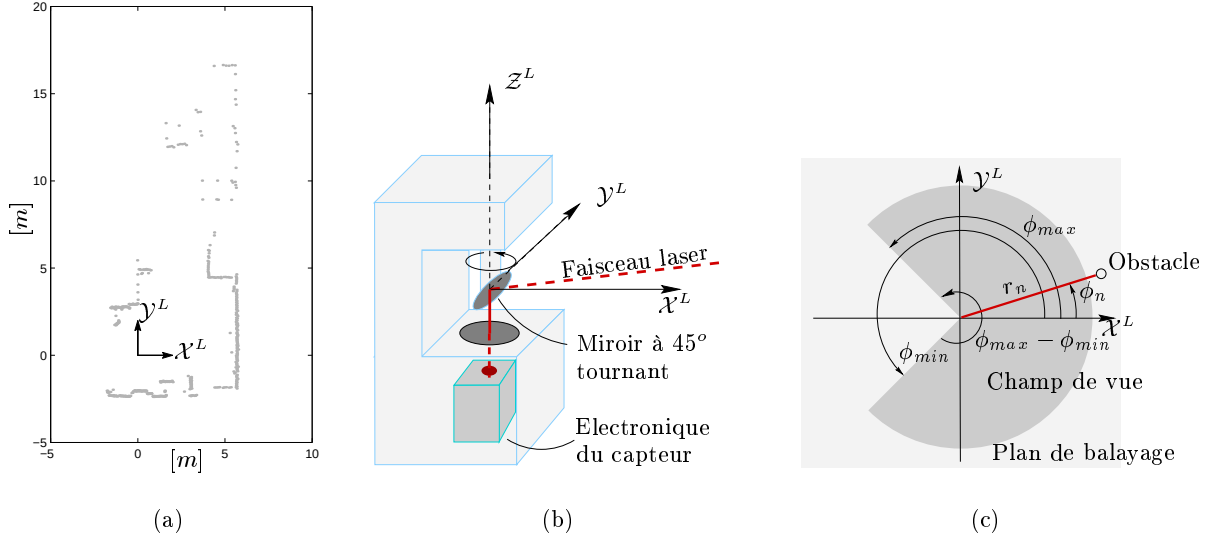


FIG. 3.2 – (a) Image de profondeur fournie par le télémètre laser. (b) Illustration du repère télé-métrique. (c) Plan de balayage laser.

sont parallèles (cf. Figure 3.3(b)).

3.3 Cohérence des images multicapteur

Les cartes locales sont une représentation géométrique de l'environnement local dans lequel le robot navigue. Elles sont construites par fusion d'indices géométriques extraits d'une paire d'images $(\mathcal{L}, \mathcal{I})$, supposées cohérentes avec l'environnement local à l'instant d'acquisition. Pour cela, il faut que les deux conditions suivantes soient satisfaites :

- I. *Condition de cohérence temporelle* : La différence entre les instants de fin d'acquisition de $\mathcal{L}$ et $\mathcal{I}$, respectivement $t_{\mathcal{L}}$ et $t_{\mathcal{I}}$, doit être négligeable. Le non respect de cette condition signifie que les mesures obtenues par les deux capteurs pour un même objet de l'environnement correspondent à des points de vue différents. Si le décalage temporel $\|t_{\mathcal{L}} - t_{\mathcal{I}}\|$ est trop important, les images ne peuvent pas être fusionnées ;
- II. *Condition de cohérence morphologique* : $\mathcal{L}$ doit être compatible avec la forme de l'environnement local. Le respect de cette condition est fondamental pour la construction de cartes locales précises.

En pratique, plusieurs phénomènes rendent complexe l'utilisation des systèmes multicapteur : (i) les mesures sont perturbées par des phénomènes physiques de nature aléatoire, (ii) elles ne sont délivrées de manière synchrone car les capteurs ont des fréquences d'acquisition différentes et temps de traitement des données dépend de leur nature, (iii) le mouvement des capteurs perturbe les mesures et cela d'autant plus que le temps d'acquisition est élevé. Ainsi, les conditions de

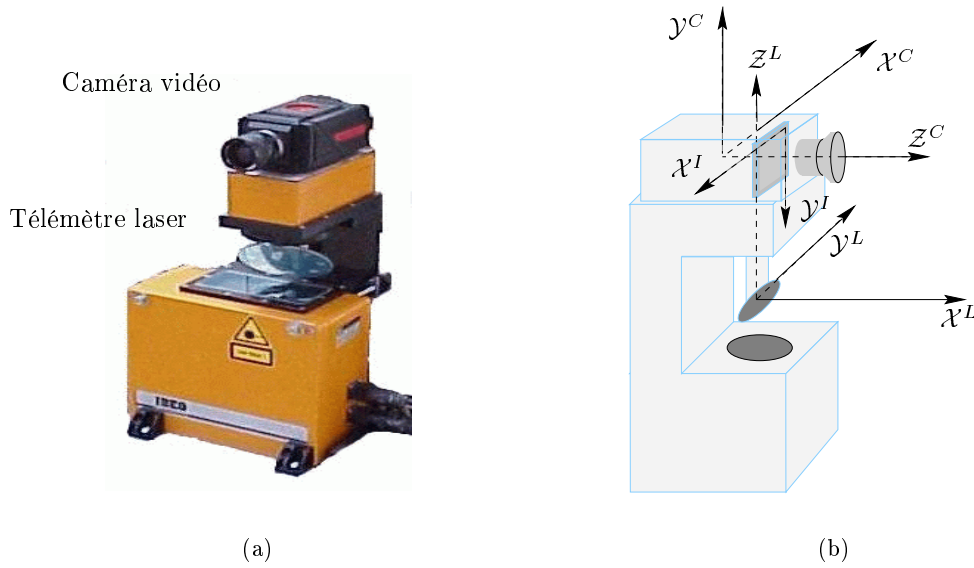


FIG. 3.3 – (a) Système multicapteur embarqué sur le robot Omni. (b) Configuration simplifiée des repères capteurs.

cohérence ne sont généralement pas respectées. Nous proposons ici de minimiser l'influence de ces phénomènes.

3.4 Processus d'acquisition

La fusion de données issues d'une image télémétrique et d'une image vidéo implique que ces images correspondent à l'observation instantanée de l'environnement local lorsque le robot se trouve à une position donnée. Cependant, elles ne sont généralement pas acquises de façon instantanée, et il n'y a pas de synchronisation des deux capteurs. Par conséquent, nous avons implanté une méthodologie permettant de réaliser une pseudo-synchronisation de ces images.

Avant de présenter cette méthodologie, nous allons décrire la façon dont les images sont acquises et datées par l'ordinateur du robot Omni :

- *Acquisition* : Le télémètre laser opère en mode continu, en envoyant une image à l'ordinateur du robot chaque 125 ms. En fait, le télémètre est interrogé périodiquement par l'ordinateur central dans le but de détecter l'instant de disponibilité d'une image, c'est à dire, juste après le balayage dans la dernière direction, à $\phi_{\max}$. L'image est alors transférée vers l'ordinateur central. Une fois reçue par l'ordinateur, elle n'est prise en considération par le système que si une requête a été faite par une autre tâche avant son arrivée. La caméra vidéo génère aussi des images de façon continue. Ces images arrivent dans la carte de numérisation, et elles ne sont stockées en mémoire que si cette carte reçoit une requête de l'ordinateur.
- *Datation* : La datation des images se fait à l'instant où elles arrivent dans l'ordinateur central en utilisant le temps absolu à l'arrivée. En fait, en considérant que le transfert de données

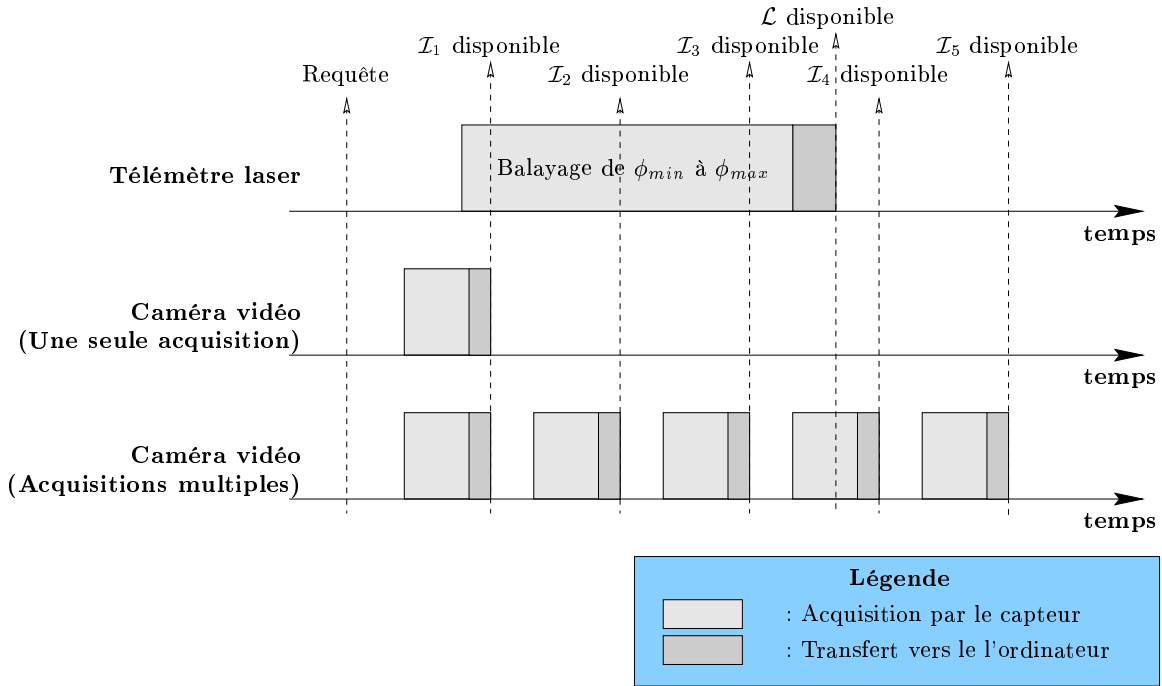


FIG. 3.4 – Séquence temporelle des événements associés à l'acquisition de données multicapteur.

s'effectue très rapidement, nous pouvons attacher le repère temporel à l'instant d'acquisition du dernier point de balayage (resp. *pixel*) de l'image télémétrique (resp. image caméra). Des mesures effectuées sur ce télémètre laser indiquent que le temps de transfert est de l'ordre de 8-10 *ms*. Pour la caméra, nous n'avons pas d'information précise sur le temps de transfert. Cependant, nous supposons que ce temps est assez court à cause du débit élevé du bus PCI, sur lequel la carte d'acquisition est connectée.

Nous notons t_L et t_C les instants de datation respectifs de l'image télémétrique et de l'image vidéo.

La Figure 3.4 montre la séquence temporelle des événements associés à l'acquisition de données. Dans cette Figure, nous avons trois axes temporels qui indiquent les instants de début et de fin de chaque événement, ainsi que leur durée. L'axe du haut fait référence au télémètre laser dans le cas où la requête arrive quand le balayage du secteur visible (*i.e.*, $\phi_{min} \leq \phi \leq \phi_{max}$) n'est pas encore terminé. Ce cas de figure peut engendrer le plus grand décalage temporel entre les deux images. L'axe du milieu fait référence à l'acquisition d'une seule image caméra. Supposons que les deux systèmes reçoivent une requête d'acquisition au même instant. L'acquisition d'images caméra étant plus rapide que celle du télémètre laser, il est fort probable qu'on dispose d'une image caméra bien avant que fin d'acquisition d'une image télémétrique. Ce cas est illustré sur la Figure 3.4. Cependant, il est aussi possible que l'image télémétrique soit disponible avant l'image vidéo, dans le cas où l'acquisition de l'image télémétrique courante est presque terminée. Pour résumer, le décalage temporel entre la fin d'acquisition des deux images est difficile à prévoir *a priori*.

Notre objectif principal ici est d'acquérir des images qui aient le minimum de décalage tem-

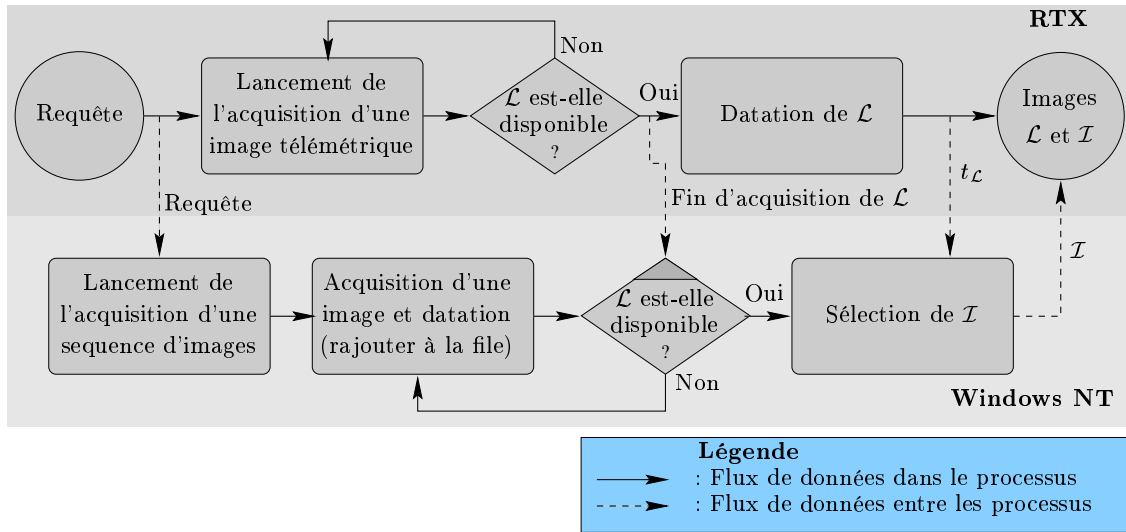


FIG. 3.5 – Schéma de l'acquisition synchronisée de données multicapteur.

porel. La solution adoptée consiste à déclencher plusieurs captures vidéo pendant une acquisition du télémètre laser. Cette solution est illustrée par le troisième axe de la Figure 3.4. Les images vidéo sont datées et gardées dans une file. Quand l'image télémétrique est disponible, l'acquisition d'images vidéo s'arrête. Ainsi, parmi les images vidéo acquises, nous choisissons celle qui présente le moindre décalage temporel par rapport à l'image télémétrique. Pour l'exemple illustré sur la Figure 3.4, l'image vidéo $\mathcal{I}_4$ est la plus cohérente avec l'image télémétrique $\mathcal{L}$.

Le schéma de la Figure 3.5 montre le principe du processus *RTX-WinNT* que nous avons implanté pour l'acquisition synchronisée des données. Etant donné que Matrox, le fabricant de la carte d'acquisition vidéo, ne fournit pas de pilotes pour *RTX*, l'acquisition d'images doit forcément se passer au niveau d'une tâche de priorité haute sous *WinNT*. L'acquisition de données télémétriques est réalisée par une tâche *RTX* qui, périodiquement, interroge le télémètre pour vérifier si le balayage courant est terminé. Si c'est le cas, la tâche *RTX* s'occupe du transfert de l'image vers l'ordinateur central. La Figure 3.5 montre le flux de données, les procédures et la communication entre les tâches *RTX-WinNT*.

Dans cette architecture, la datation des images multicapteur n'a lieu qu'après réception des données par le processus responsable de son acquisition. Les processus sont gérés par un système multitâche monoprocesseur. Ce qui signifie qu'ils ne s'exécutent pas en parallèle, mais de façon séquentielle, en fonction des priorités associées. Ainsi, la tâche d'acquisition de données télémétriques étant gérée par un processus *RTX*, celle-ci a une priorité d'exécution plus importante que l'acquisition de données vidéo. Pour mieux comprendre les effets de l'exécution asynchrone des tâches dans la datation des images, nous présentons l'histogramme de la Figure 3.6. Cet histogramme montre le nombre de paires d'images $(\mathcal{L}, \mathcal{I})$ par décalage temporel $t_{\mathcal{L}} - t_{\mathcal{I}}$, discrétisé en unités de 2 ms . Ainsi, pour un ensemble de 851 paires d'images multicapteur, ce décalage est limité à l'intervalle $[-40\text{ ms}, 40\text{ ms}]$. Nous remarquons que, pendant le transfert de l'image télémétrique, la datation

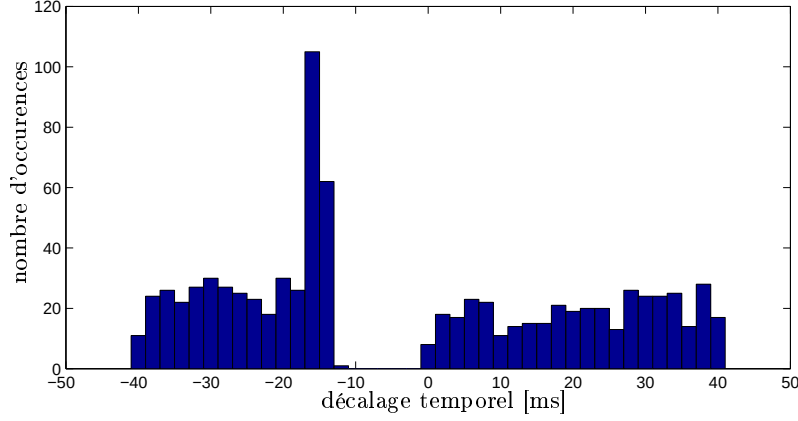


FIG. 3.6 – Histogramme du décalage temporel entre les instants de datation des images télémétriques et des images vidéo.

des images vidéo ne peut pas avoir lieu. Ceci se traduit dans l'histogramme par un intervalle vide de -10 ms à 0 ms .

Remarque 3.1: *L'Histogramme de la Figure 3.6 indique qu'un nombre important de paires d'images $(\mathcal{L}, \mathcal{I})$ présentent un décalage temporel de datation situé dans l'intervalle $[-16\text{ ms}, -14\text{ ms}]$. Cependant, l'histogramme montre qu'il existe une probabilité presque uniforme de l'occurrence d'écarts temporels dans les intervalles $[-40\text{ ms}, -18\text{ ms}]$ et $[0\text{ ms}, 40\text{ ms}]$. Nous pensons donc que le grand nombre de décalages dans l'intervalle $[-16\text{ ms}, -14\text{ ms}]$ est dû aux cas où la fin d'acquisition de l'image vidéo coïncide avec la phase de transfert de l'image télémétrique vers l'ordinateur du robot. Ainsi, pour ce cas, l'erreur de datation de l'image vidéo peut être comprise entre 0 et 10 ms. $\diamond$*

L'histogramme de la Figure 3.6 montre que des écarts temporels peuvent être vérifiés entre $\mathcal{L}$ et $\mathcal{I}$, ce qui peut rendre ces images incohérentes avec l'environnement local. Bien entendu, une telle incohérence ne se manifeste que si le véhicule navigue trop vite. Nous savons que les composantes en translation (v_t) et en rotation (v_θ) de la vitesse v du robot Omni peuvent facilement atteindre $v_t = 1\text{ m/s}$ et $v_\theta = 50^\circ/\text{s}$. Pour ces cas extrêmes, un simple calcul nous permet d'imaginer que le robot pourrait avoir effectué un parcours de 4 cm et un changement de cap de 2° entre les instants de datation des images. Ces valeurs semblent infimes, mais elles peuvent affecter la mise en correspondance d'indices géométriques lors de la construction de cartes locales. Ainsi, nous proposons, dans la section 3.5.3, une solution qui minimise le problème d'incohérence temporelle.

3.5 Prétraitement des données

Une fois la paire d'images $(\mathcal{L}, \mathcal{I})$ acquise et datée, elle doit être pré-traitée avant d'être segmentée. Les objectifs principaux du pré-traitement sont (i) la correction de phénomènes physiques intrinsèques au télémètre laser et (ii) la compensation des effets du mouvement dans la cohérence des images. Pour justifier ces pré-traitements des images télémétriques, nous avons effectué des évaluations qui sont présentées dans cette section.

3.5.1 Caractérisation des mesures du télémètre laser

Les mesures de distance r du télémètre laser ont une incertitude due à divers phénomènes physiques (cf. section 3.2). Ceux-ci peuvent engendrer des non-linéarités de mesure [26]. En conséquence, les fabricants implantent généralement dans l'électronique du capteur des solutions pour corriger au moins une partie de ces effets. Toutefois, nous avons procédé à une caractérisation du bruit à partir des mesures réelles du capteur.

Dans la section 3.2, nous avons supposé que les erreurs de mesure suivaient une distribution gaussienne de moyenne nulle et de variance σ_r^2 , ce qui correspond à un modèle stochastique de la forme

$$r = r_v + \varepsilon_r, \quad (3.3)$$

où r_v représente la vraie distance, et ε_r est une variable aléatoire suivant $\mathcal{N}(0, \sigma_r^2)$. Néanmoins, ce modèle est incohérent à cause de la présence d'un biais de mesure, qui se traduit par une erreur moyenne non-nulle et fonction de la mesure r . Ainsi, pour modéliser ce bruit, considérons le modèle étendu suivant :

$$r = r_v + \varepsilon_b(r) + \varepsilon_r. \quad (3.4)$$

Dans ce nouveau modèle, $\varepsilon_b(r)$ est un terme déterministe correspondant au biais de mesure. La procédure de caractérisation d'erreurs du télémètre laser consiste à identifier $\varepsilon_d(r)$ et la variance σ_r^2 de ε_r . Pour cela, nous avons mené l'expérience suivante : on déplace le robot en ligne droite perpendiculairement à un mur et on réalise, à des intervalles de temps constants, des mesures de la distance r du capteur au mur. Sachant que les mesures du télémètre sont fournies en représentation polaire, la mesure r choisie est celle dont l'angle d'incidence est le plus proche de 90° . Cela signifie que la procédure de caractérisation ne prend pas en compte l'influence de l'incidence. La distance parcourue mesurée par la gyrodrométrie (cf. section 6.2.3), et confirmée par une mesure nouvelle, est utilisée pour calculer la référence r_v de chaque échantillon.

Etant donné $N = 91$ mesures télémétriques $r(n)$ et distances de référence $r_v(n)$, avec $n = 1, \dots, N$, les estimées sont déterminées à partir de

$$\varepsilon_b(r(n)) = r(n) - r_v(n) - \varepsilon_r(n). \quad (3.5)$$

Au regard d'une procédure d'identification, $\varepsilon_b(r(n))$ est estimé à partir des régresseurs $r(n) - r_v(n)$, et les résidus d'estimation suivent la même distribution que $\varepsilon_r(n)$. Ainsi, dans un premier temps, nous supposons que $\varepsilon_d(r_m(n))$ peut être modélisé par un polynôme du sixième degré, *i.e.*,

$$\hat{\varepsilon}_d(r(n)) = \sum_{k=0}^6 \hat{a}_k \cdot r^k(n). \quad (3.6)$$

Ce modèle est estimé à l'aide de la méthode des moindres-carrés comme étant celui qui minimise la somme des carrés de $r(n) - r_v(n) - \hat{\varepsilon}_b(r(n))$. Dans un deuxième temps, la variance de ε_r est estimée de façon empirique à partir des résidus d'identification par l'estimateur non biaisé

$$\hat{\sigma}_r^2 = \frac{1}{N-1} \sum_{n=1}^N (r(n) - r_v(n) - \hat{\varepsilon}_b(r(n)))^2. \quad (3.7)$$

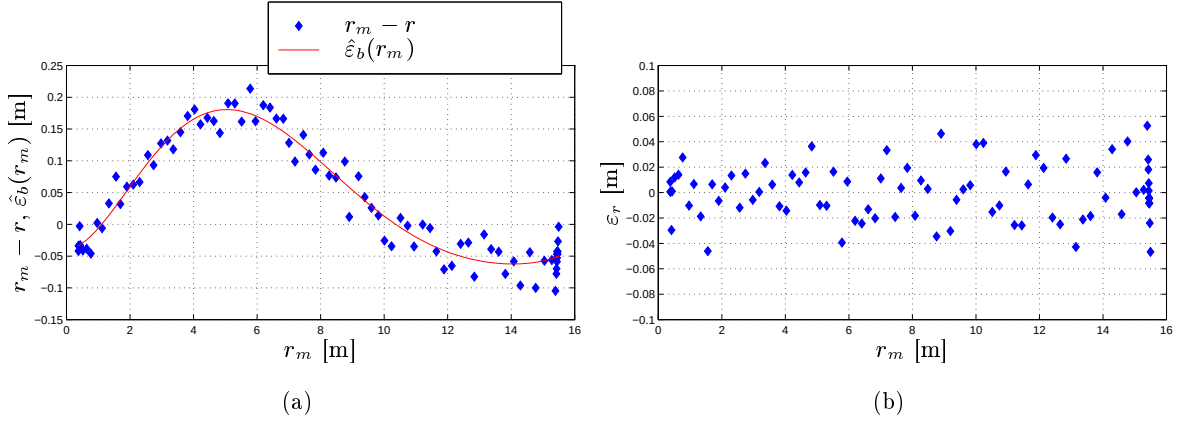


FIG. 3.7 – Caractérisation de l'erreur de mesure du télémètre laser : (a) $r_m - r$ et l'estimée $\hat{\epsilon}_b(r_m)$, et (b) les résidus $r_m - r - \hat{\epsilon}_b(r_m)$.

Les résultats de cette expérience sont présentés dans les graphiques de la Figure 3.7. Sur la Figure 3.7(a), les losanges correspondent à $r(n) - r_v(n)$, ce qui donne l'allure du biais $\epsilon_b(r(n))$. Nous pouvons vérifier que le biais de mesure est assez petit pour des mesures inférieures à 2 m et au delà de 9 m. Néanmoins, le biais de mesure peut être important à l'intérieur de cet intervalle, comme nous pouvons le constater pour un obstacle situé à 5 m du capteur. Dans ce cas, le biais est de l'ordre de 17 cm. Le modèle estimé $\hat{\epsilon}_b(r(n))$ est représenté par la courbe de la Figure 3.7(a). Les paramètres estimés sont

$$\begin{aligned}\hat{a}_6 &= 3,69 \cdot 10^{-7}, \hat{a}_5 = -2,96 \cdot 10^{-5}, \hat{a}_4 = 7,92 \cdot 10^{-4}, \hat{a}_3 = -8,71 \cdot 10^{-3}, \\ \hat{a}_2 &= 3,24 \cdot 10^{-2}, \hat{a}_1 = 1,89 \cdot 10^{-2}, \hat{a}_0 = -4,84 \cdot 10^{-2}.\end{aligned}$$

Les résidus $r(n) - r_v(n) - \hat{\epsilon}_b(r(n))$, utilisés pour estimer la variance de ϵ_r , sont montrés sur la Figure 3.7(b). Nous pouvons vérifier que la dispersion est à peu près uniforme en amplitude pour l'intervalle des distances mesurées dans cette expérience. Cela suggère que ϵ_r a une distribution indépendante de r_m . La variance estimée de cette composante stochastique est $\hat{\sigma}_r^2 = (2,11 \cdot 10^{-2})^2 m^2$, ce qui se traduit par un écart-type de 2,11 cm pour la mesure de distance. Cette valeur est très proche de celle fournie par le fabricant : 3,0 cm (voir tableau 3.1). Néanmoins, le fabricant ne mentionne pas l'existence d'un biais de mesure (cf. Tableau 3.1).

3.5.2 Etude de l'influence du mouvement sur les images télémétriques

Dans un travail précurseur sur la cartographie et la localisation de robots mobiles de J. Leonard et Durrant-Whyte [140], l'acquisition de données sonar se faisait avec le véhicule à l'arrêt. Cela était dû au besoin d'avoir des observations cohérentes avec l'environnement local, et à la puissance de calcul limitée de la machine embarquée. De nos jours, la puissance de calcul n'est plus (ou presque!) une barrière pour ce type d'application. Cependant, la vitesse d'acquisition du télémètre

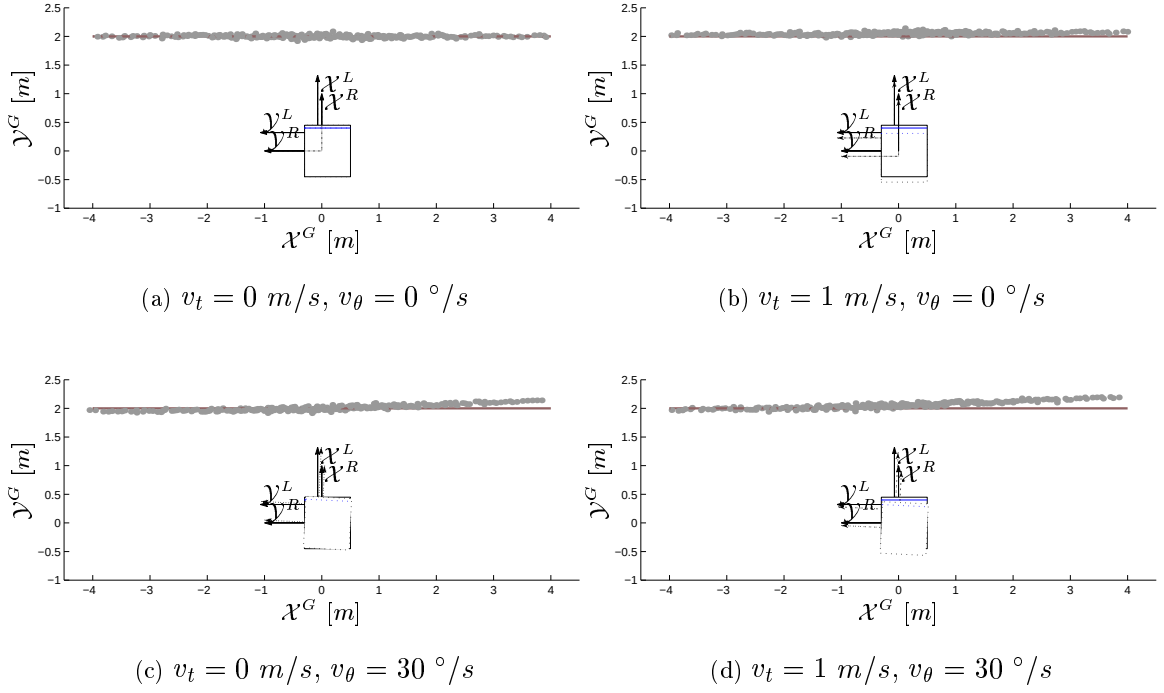


FIG. 3.8 – Simulation d'images télémétriques avec le robot (a) en arrêt et (b,c,d) en mouvement.

laser est encore très lente, au point que les images sont notablement déformées par le mouvement du robot pendant le balayage du capteur.

Nous avons remarqué que la plupart des travaux utilisant des télémètres laser ne font pas allusion à l'influence du mouvement dans les images télémétriques. Ceci peut être dû au fait que les vitesses employées pendant les évaluations expérimentales ne sont pas très importantes, et certainement inférieures à celles atteignables par le véhicule.

Dans cette section nous présentons une analyse par simulation de l'influence du mouvement dans la déformation des images télémétriques. Une étude antérieure présentée dans [14] simulait un capteur à balayage 3-D embarqué sur une voiture se déplaçant à une vitesse importante (de l'ordre de 25 m/s).

Nous nous limitons, dans cette étude, à des vitesses réalisables par des robots mobiles d'intérieur, et susceptibles de provoquer des déformations dans les images télémétriques. L'étude utilise des données générées par un simulateur du télémètre laser qui prend en compte le mouvement réalisé par le robot pendant l'acquisition d'images. Pour simuler le télémètre, nous avons utilisé les paramètres du capteur embarqué sur le robot Omni. Les murs de l'environnement sont représentés par des segments de droite dans le plan de balayage. Aux mesures de distance r_n , on rajoute une erreur aléatoire suivant $\mathcal{N}(0, \sigma_r^2)$. Nous supposons que le robot évolue dans un environnement constitué de murs verticaux. Pour chaque acquisition, le simulateur requiert comme arguments $\mathbf{z}_1$ et $\mathbf{z}_N$, les positions du robot au début et à la fin du balayage. Ces arguments sont déterminés en fonction du type de mouvement simulé.

Sans perte de généralité, nous considérons que la position $\mathbf{z}_N$ du robot correspondant à la fin du balayage est $\mathbf{z}_N = (0, 0, \pi/2)$. La position initiale $\mathbf{z}_1$ lors du balayage dans la direction ϕ_1 est déterminée en supposant que le robot ait réalisé un mouvement uniforme de vitesse de translation v_t et vitesse de rotation v_θ .

L'environnement de cette simulation est constitué seulement d'un mur représenté dans le plan de balayage par le segment qui lie les points $(-4, 2)$ et $(4, 2)$. En fonction de v_t et v_θ , nous avons étudié les cas de figure suivants :

- Robot à l'arrêt ($v_t = 0$ m/s et $v_\theta = 0$ °/s, Figure 3.8(a)). L'image générée dans ce cas nous servira pour comparaison avec les images acquises avec le véhicule en mouvement ;
- Robot en translation ($v_t = 1$ m/s et $v_\theta = 0$ °/s, Figure 3.8(b)). Nous remarquons qu'il y a eu très peu de déformation de l'image télémétrique par rapport à celle obtenue avec le véhicule à l'arrêt. Cette déformation s'exprime principalement par un changement d'inclinaison de la direction principale du nuage de points de balayage. Bien entendu, si la translation était parallèle au mur, la déformation du nuage de points serait infime, comparée au bruit de mesure $\mathcal{N}(0, \sigma_r^2)$;
- Robot en rotation ($v_t = 0$ m/s et $v_\theta = 30$ °/s, Figure 3.8(c)). La vitesse de rotation utilisée dans ce cas n'est pas la vitesse maximale que le système peut atteindre. En effet, le robot Omni peut avoir des vitesses de rotation allant jusqu'à $v_\theta = 50$ °/s. Nous avons choisi $v_\theta = 30$ °/s pour cette simulation pour montrer que même avec des vitesses de rotation parfaitement utilisables en navigation balistique, les déformations dans les images télémétriques sont importantes. Cela se vérifie dans l'image de la Figure 3.8(c). La déformation provoquée dans l'image télémétrique se traduit par une courbure dans le nuage de points. Nous remarquons que les repères robot et capteur n'ayant pas la même origine, un mouvement de rotation dans le premier repère se traduit par un mouvement composé d'une translation et d'une rotation dans le second repère ;
- Robot en translation et rotation ($v_t = 1$ m/s et $v_\theta = 30$ °/s, Figure 3.8(d)). On retrouve ici le cumul des déformations obtenues précédemment.

De cette étude, nous concluons que :

- I. La composante en translation de la vitesse du robot a une forte influence sur l'orientation dans le repère capteur des nuages de points correspondant à des surfaces planes. Cependant, nous avons vérifié que l'impact majeur sur cette orientation est obtenue pour de grandes vitesses dans la direction orthogonale au mur ;
- II. La composante en rotation de la vitesse du robot a une forte influence sur la courbure des nuages de points. Nous avons vérifié que des déformations importantes sont présentes dans les images obtenues avec des vitesses de rotation classiques pour des robots mobiles.

Il est clair que ces conséquences sont encore plus importantes pour des télémètres laser plus lents que celui utilisé ici.

3.5.3 Procédures de prétraitement

Les sections précédentes nous ont permis d'évaluer le degré de dégradation des images télémétriques résultant de l'erreur de mesure et du mouvement du véhicule. Dans cette section, nous présentons des procédures de compensation appliquées aux images télémétriques. En ce que concerne l'incohérence temporelle de la paire d'images $(\mathcal{L}, \mathcal{I})$, nous présentons aussi une méthodologie de correction qui est dérivée de la procédure de compensation du mouvement dans les images télémétriques. Cette méthodologie est intitulée *pseudo-synchronisation des images du système multicapteur*.

Correction du biais de mesure du télémètre laser Dans la caractérisation des mesures du télémètre laser (section 3.5.1), nous avons vérifié que ce capteur présente un biais de mesure ε_b sur la distance mesurée r . Ce biais a été caractérisé, et une estimée $\hat{\varepsilon}_b(r)$ peut être calculée en ligne par un polynôme du sixième degré. Si un tel biais n'est pas compensé, la cohérence morphologique de l'image télémétrique avec l'environnement local est mise en cause. Pour cela, nous proposons une correction des mesures télémétriques en utilisant la formule suivante :

$$r \text{ \{mesure corrigée\}} = r \text{ \{mesure du capteur\}} - \hat{\varepsilon}_b(r) \{\text{biais estimé}\}. \quad (3.8)$$

Nous remarquons que la caractérisation du biais de mesure a été obtenue pour $0,375m \leq r \leq 15,457m$. Les mesures en dehors de cet intervalle n'ont pas été caractérisées. Cependant, étant donné la forme du biais de mesure (cf. Figure 3.7(a)), on peut admettre que celui-ci diminue beaucoup en dehors de cet intervalle. Nous considérons qu'il est du même ordre de grandeur que l'écart-type des mesures. Nous proposons ainsi l'extrapolation suivante : $\hat{\varepsilon}_b(r) = \hat{\varepsilon}_b(0,375)$, pour $r < 0,375m$, et $\hat{\varepsilon}_b(r) = \hat{\varepsilon}_b(15,457)$, pour $r > 15,457m$.

Compensation de la déformation des images télémétriques Dans la section 3.5.2 nous avons vérifié que des vitesses importantes peuvent engendrer des déformations dans les images. Elles peuvent être source d'erreurs importantes pour un système de localisation par cartes géométriques. Une solution pour résoudre ce problème serait de limiter la vitesse du robot, au moins pendant les cycles de localisation absolue, ou encore de ne pas considérer les images télémétriques quand d'importantes rotations sont détectées. Cependant, la première solution impose des contraintes dans le planificateur de trajectoires. La seconde peut accroître l'incertitude de l'estimée de position mise à jour par odométrie pendant les grandes rotations. Nous proposons d'utiliser des estimées de $\mathbf{z}_1$ et $\mathbf{z}_N$ (i.e., positions absolues du robot au début et à la fin du balayage) pour corriger l'image télémétrique courante. La correction est basée sur l'hypothèse d'un mouvement uniforme du robot pendant le balayage.

Cette procédure utilise la représentation cartésienne de l'image télémétrique :

$$\mathcal{L} = \{(x_n, y_n) | n = 1, \dots, N; \}, \quad (3.9)$$

où $x_n = r_n \cdot \cos(\phi_n)$ et $y_n = r_n \cdot \sin(\phi_n)$ sont les coordonnées cartésiennes du n -ième point de balayage (r_n, ϕ_n) . Les estimées de $\mathbf{z}_1$ et $\mathbf{z}_N$, notées respectivement $\hat{\mathbf{z}}_1$ et $\hat{\mathbf{z}}_N$, sont fournies

par l'estime. Dans notre architecture, les estimées de la position $\hat{\mathbf{z}}$ du robot sont stockées de façon temporaire dans des queues circulaires [29][30]. Ces estimées sont fournies par l'estime à une période constante de 5 ms, et sont datées avant d'être stockées dans les queues suivant leur ordre d'acquisition. Une fois l'image télémétrique disponible, $\hat{\mathbf{z}}_1$ et $\hat{\mathbf{z}}_N$ sont récupérées comme estimées de position générées aux instants $t_{\mathcal{L}} - T \cdot (\phi_{\max} - \phi_{\min})/2\pi$ et $t_{\mathcal{L}}$, respectivement.

La compensation est faite dans l'espace du capteur. Ainsi, les estimées $\hat{\mathbf{z}}_{L,1}$ et $\hat{\mathbf{z}}_{L,N}$ de sa position sont obtenues par une transformation de repères à partir de $\hat{\mathbf{z}}_1$ et $\hat{\mathbf{z}}_N$. Comme le repère temporel $t_{\mathcal{L}}$ est associé au dernier point de balayage, nous avons décidé d'attacher à $\hat{\mathbf{z}}_{L,N}$ le repère temporel final de l'image télémétrique. Ainsi, en prenant comme référence le N -ième point, le repère capteur aura effectué un déplacement de $\hat{\mathbf{z}}_{L,N} - \hat{\mathbf{z}}_{L,n}$ dès la n -ième acquisition. Celle-ci a eu lieu à la position $\hat{\mathbf{z}}_{L,n} = (\hat{x}_{L,n}, \hat{y}_{L,n}, \hat{\theta}_{L,n})^T$ de la trajectoire du capteur qui est inconnue. Dans l'image télémétrique, (x_n, y_n) représente les coordonnées de la n -ième acquisition lorsque le capteur était à la position absolue $\hat{\mathbf{z}}_{L,n}$. Pour estimer les coordonnées $(\tilde{x}_n, \tilde{y}_n)$ de la n -ième acquisition dans le repère associé à la position finale $\hat{\mathbf{z}}_{L,N}$, nous appliquons la transformation suivante :

$$\begin{pmatrix} \tilde{x}_n \\ \tilde{y}_n \end{pmatrix} = \mathbf{R}(\hat{\theta}_{L,N} - \hat{\theta}_{L,n}) \left\{ \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \begin{pmatrix} \hat{x}_{L,N} - \hat{x}_{L,n} \\ \hat{y}_{L,N} - \hat{y}_{L,n} \end{pmatrix} \right\}. \quad (3.10)$$

Une procédure similaire utilisant le formalisme des coordonnées homogènes est proposée par Arras *et al.* [8]. Contrairement à notre approche, Arras associe le repère temporel de l'image télémétrique au premier point du balayage. Les excellents résultats publiés illustrent bien l'intérêt de la correction effectuée.

L'équation (3.10) montre qu'il n'est pas nécessaire connaître la position absolue du capteur pour que la compensation soit appliquée. En effet, $\hat{x}_{L,N} - \hat{x}_{L,n}$, $\hat{y}_{L,N} - \hat{y}_{L,n}$ et $\hat{\theta}_{L,N} - \hat{\theta}_{L,n}$ représentent la position relative du capteur à la n -ième acquisition. Ainsi, il nous faut seulement connaître $\Delta\hat{\mathbf{z}}_{L,n} = \hat{\mathbf{z}}_{L,N} - \hat{\mathbf{z}}_{L,n}$. Ainsi, dans l'hypothèse d'un mouvement uniforme du robot au cours d'un balayage, nous calculons le déplacement relatif nécessaire pour compenser la n -ième mesure par :

$$\Delta\hat{\mathbf{z}}_{L,n} = \frac{N-n}{N-1} \cdot (\hat{\mathbf{z}}_{L,N} - \hat{\mathbf{z}}_{L,1}). \quad (3.11)$$

Pseudo-synchronisation des images du système multicapteur La procédure d'acquisition présentée dans la section 3.4 permet de limiter le décalage temporel entre les deux images $(\mathcal{L}, \mathcal{I})$ à l'intervalle $-40ms \leq t_{\mathcal{L}} - t_{\mathcal{I}} \leq 40ms$. Cependant, un mouvement important du véhicule, surtout en rotation, peut rendre la paire d'images incompatible avec l'environnement local. Nous proposons ici une procédure de pseudo-synchronisation des images. Cette procédure consiste à calculer dans le repère télémètre correspondant à l'instant $t_{\mathcal{I}}$ les mesures acquises à l'instant $t_{\mathcal{L}}$ par un simple changement de repère. Ainsi, $\hat{\mathbf{z}}_{L,t_L} = (\hat{x}_{L,t_L}, \hat{y}_{L,t_L}, \hat{\theta}_{L,t_L})$ et $\hat{\mathbf{z}}_{L,t_I} = (\hat{x}_{L,t_I}, \hat{y}_{L,t_I}, \hat{\theta}_{L,t_I})$ étant respectivement les estimées des positions absolues du télémètre aux instants $t_{\mathcal{L}}$ et $t_{\mathcal{I}}$, en supposant un mouvement uniforme du robot durant cette période, nous avons

$$\begin{pmatrix} \tilde{x}_n \\ \tilde{y}_n \end{pmatrix} = \mathbf{R}(\hat{\theta}_{L,t_I} - \hat{\theta}_{L,t_L}) \left\{ \begin{pmatrix} \tilde{x}_n \\ \tilde{y}_n \end{pmatrix} - \begin{pmatrix} \hat{x}_{L,t_I} - \hat{x}_{L,t_L} \\ \hat{y}_{L,t_I} - \hat{y}_{L,t_L} \end{pmatrix} \right\}, \quad (3.12)$$

pour chaque n -ième point de balayage $(\tilde{x}_n, \tilde{y}_n)$ obtenu après compensation du mouvement (eq. 3.10). Après l'application de cette procédure, l'image télémétrique est affectée de la date t_L . Comme pour la procédure de compensation du mouvement, il nous suffit de connaître le déplacement relatif $\hat{\mathbf{z}}_{L,t_I} - \hat{\mathbf{z}}_{L,t_L}$, qui peut être fourni par odométrie.

3.6 Evaluation expérimentale

Pour mieux évaluer la pertinence des prétraitements présentés en section 3.5.3, nous avons réalisé trois expériences de validation avec le robot Omni.

3.6.1 Expérience sur la correction du biais

Cette expérience a consisté à acquérir des images télémétriques au cours du déplacement du robot Omni suivant une trajectoire rectiligne à très basse vitesse. L'environnement de cette expérience est bien structuré, avec une prédominance de longs murs. Les images brutes sont donc superposées (Figure 3.9(a)). L'acquisition à très basse vitesse permet de limiter la déformation des images. Ainsi, ce que nous voyons ce sont des images sans correction de biais et avec un minimum de déformation. Nous pouvons donc remarquer que les nuages de points correspondant aux murs sont très épais. Cet épaississement est encore plus important pour le mur parallèle à l'axe Y qui se trouve le plus à droite sur la Figure 3.9(a). Ceci s'explique par la trajectoire exécutée par le robot, qui fait que ce mur est observé à une distance allant progressivement d'environ 0,5 m jusqu'à 15 m (*i.e.*, le robot recule). Dans cet intervalle, le biais de mesure a varié entre -5 cm et 20 cm. La Figure 3.9(b) présente l'ensemble d'images superposées avec compensation de biais. En regardant le mur le plus à droite, nous voyons que le nuage de points est plus compact. Nous remarquons aussi une observation plus nette des colonnes du bâtiment qui sont contre le mur parallèle à l'axe X , sur le haut. La compensation de biais minimise la dispersion des mesures, et permet une meilleure observation de l'emplacement des obstacles de l'environnement local.

3.6.2 Expérience sur la compensation du mouvement

La deuxième expérience a comme objectif la validation de la correction des déformations des images dues au mouvement du robot. Comme pour l'expérience précédente, le robot Omni navigue dans un environnement réel, mais cette fois ci sans restriction de vitesse. Au total, nous avons obtenu 60 images dans cette expérience, avec v_t allant jusqu'à 50 cm/s, et v_θ allant jusqu'à 30 °/s en module. La Figure 3.10(a) présente les mesures brutes. Etant donné que la correction du biais doit précéder la compensation du mouvement, nous présentons l'évolution des images durant ces procédures. Ainsi, la Figure 3.10(b) montre l'ensemble d'images après correction du biais. Cependant, il n'est pas évident que la correction du biais ait apporté des améliorations entre les Figures 3.10(a) et 3.10(b). En effet, nous pensons qu'ici la déformation due au mouvement du robot est prédominante par rapport au biais de mesure pour ces images. Cette hypothèse est confirmée par le résultat obtenu après compensation du mouvement (Figure 3.10(c)).

Ces bons résultats confirment que l'hypothèse du mouvement uniforme pendant le balayage du télémètre est valable. Cependant, un dérapage du robot pendant la prise d'images télémétriques peut mettre en cause la correction de mouvement, et dans ce cas les images ne seront pas cohérentes avec l'environnement local.

3.6.3 Expérience sur la pseudo-synchronisation des images

Dans cette expérience, nous montrons un cas de figure qui illustre bien le fait que l'incohérence temporelle peut engendrer des erreurs dans l'observation de l'environnement. Pour construire les cartes locales, nous devons extraire des indices géométriques de la paire d'images $(\mathcal{L}, \mathcal{I})$. Parmi ces indices, nous avons des droites verticales extraites des images vidéo. Un exemple est montré sur la Figure 3.11(a). Pour cette expérience, nous nous contentons de dire que ces indices sont sensés correspondre à des arêtes verticales de l'environnement dans le champ de vue de la caméra. La caméra ne fournissant qu'une vue projective de l'environnement, le traitement des images vidéo ne permet de connaître que la direction d'observation des indices visuels dans le repère du système multicapteur.

La Figure 3.11(b) montre le robot Omni et les directions d'observation de deux arêtes verticales de l'environnement local, représentées ici par des cercles. L'image télémétrique dans le repère associé à l'instant $t_{\mathcal{L}}$ est aussi affichée. Les cercles qui correspondent aux arêtes sont exprimés dans le repère de l'image télémétrique. Les directions d'observation ne passent pas par le centre de ces cercles, qui sont sensés être les projections des arêtes dans l'image télémétrique. Ceci est dû au fait que nous avons eu simultanément un grand écart temporel $t_{\mathcal{L}} - t_{\mathcal{I}} = -25 \text{ ms}$ et une vitesse angulaire du robot importante $v_{\theta} = 30,31^{\circ}/s$. Les écarts entre les directions d'observation et les centres des cercles augmenteront avec la distance de ces derniers par rapport aux capteurs.

Après pseudo-synchronisation, les directions d'observation des arêtes passent très près des centres des cercles (voir Figure 3.11(c)). Ainsi, nous pouvons constater que la pseudo-synchronisation est une procédure qui permet de mieux recalibrer les indices vidéo dans le repère du télémètre.

3.7 Conclusion

Dans ce chapitre nous avons présenté l'acquisition des données extéroceptives du système multicapteur du robot Omni. L'acquisition comporte la gestion des capteurs, le transfert des données et leur datation. Nous avons réalisé des études et proposé des solutions à des phénomènes qui affectent directement les images télémétriques (*e.g.*, biais de mesure et mouvement du robot durant le balayage du capteur). Nous avons aussi proposé une procédure de pseudo-synchronisation des images acquises par le système multicapteur. Notre objectif dans ce chapitre était de montrer que ces procédures sont nécessaires pour assurer la cohérence des données avec l'environnement local.

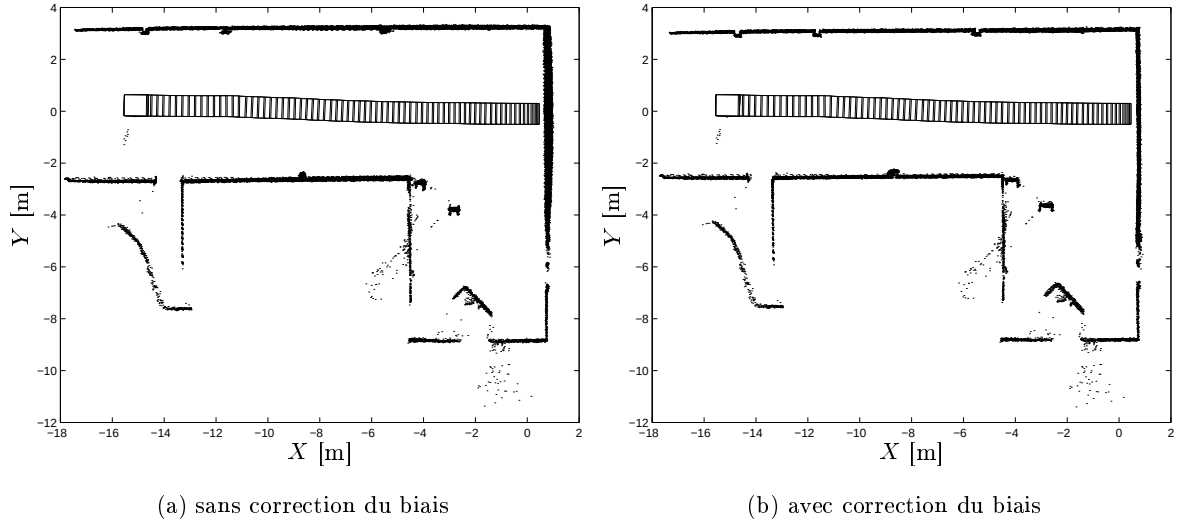


FIG. 3.9 – Evaluation visuelle de la correction du biais de mesure du télémètre laser.

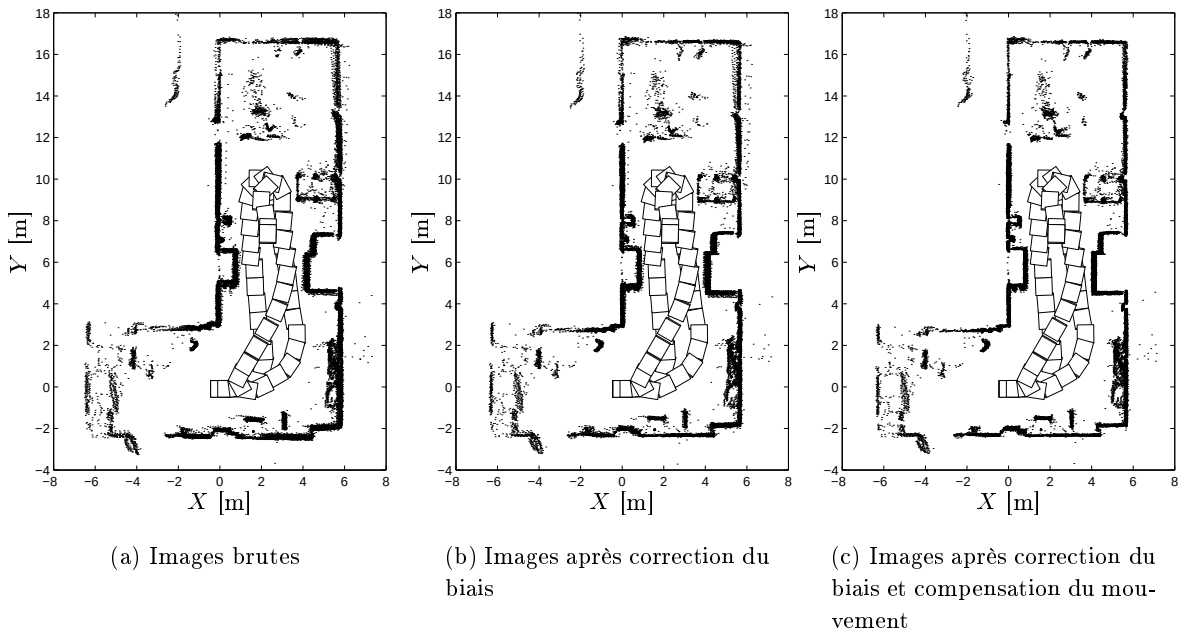
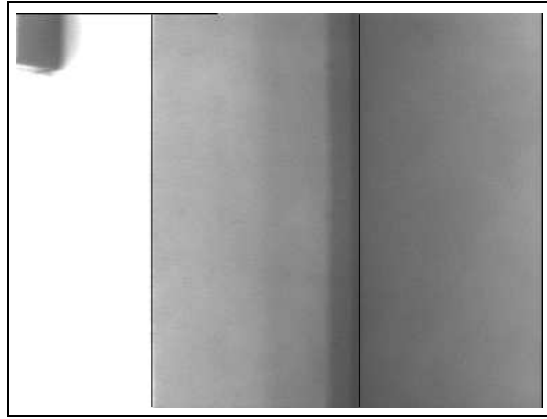
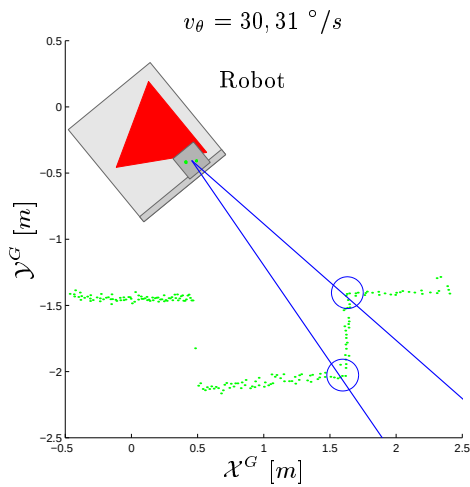


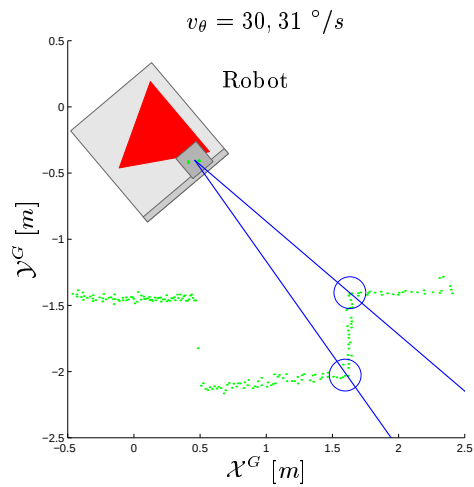
FIG. 3.10 – Résultats expérimentaux obtenus par application des procédures de prétraitement des images télémétriques.



(a) Image vidéo et les coins détectés (droites verticales)



(b) Directions d'observation des coins avant la pseudo-synchronisation des images



(c) Directions d'observation des coins après la pseudo-synchronisation des images

FIG. 3.11 – Résultats expérimentaux obtenus par application de la procédure de synchronisation des images $\mathcal{L}$ et $\mathcal{I}$ ($t_{\mathcal{L}} - t_{\mathcal{I}} = -25 \text{ ms}$).

Segmentation des images

4.1 Introduction

Dans le chapitre précédent, nous avons présenté le processus d'acquisition d'une paire d'images $(\mathcal{L}, \mathcal{I})$ du système multicapteur. Comme nous le verrons dans le chapitre 5, ces images sont utilisées pour la construction d'une carte locale de l'environnement autour du robot. Cette carte est constituée de primitives géométriques représentées dans l'espace 2-D. Notre objectif dans ce chapitre est de présenter les processus de segmentation des images $\mathcal{L}$ et $\mathcal{I}$. Ces processus réalisent l'extraction des primitives suivantes sur les images :

- *Dans l'image vidéo* : des droites verticales qui correspondent à la projection dans le plan image de structures polyédriques de l'environnement (*e.g.*, les arêtes des murs) ;
- *Dans l'image télémétrique* : des segments de droite et des points de cassure. Les segments de droite sont des indices supportés par un nombre minimum de points de l'image télémétrique, et qui correspondent en général à l'intersection du plan de balayage avec les surfaces planes de l'environnement. Les points de cassure sont des points de l'image télémétrique pour lesquels une forte discontinuité de la séquence de mesures a été détectée.

Dans ce chapitre, nous présentons les algorithmes que nous avons développés pour segmenter les deux types d'images. Nous avons mis au point et expérimenté différentes techniques d'extraction de segments dans des images télémétriques. Leur étude comparative est illustrée par des résultats expérimentaux.

4.2 Segmentation d'images vidéo

L'extraction de segments de droites dans des images vidéo est un sujet de recherche pour lequel différentes approches ont été proposées. Ces approches sont basées généralement sur la mise en œuvre d'un filtre de lissage/dérivation permettant de rehausser les contours de l'image. Ce filtrage utilise des filtres linéaires dérivatifs [48, 70], matérialisés par des masques de gradient [45]. Une fois l'image filtrée, différentes méthodes peuvent être appliquées pour sélectionner uniquement les points de fort gradient, et obtenir ainsi une image binaire des contours significatifs [99]. La phase suivante consiste alors à segmenter cette image pour extraire les primitives recherchées, c'est à dire, les droites verticales. Parmi les outils pouvant être utilisés, on peut citer la transformée de Hough [73][190], *line clustering* [225], l'optimisation combinatoire [155] ou l'analyse morphologique [136, 129].

Pour notre application, nous avons développé une approche spécialisée d'extraction de droites verticales. Cette approche est divisée en deux étapes. La première étape consiste à employer des techniques classiques de détection de contours verticaux. Dans la deuxième étape, les droites sont détectées par une sorte de transformée de Hough spécialisée. En suite, les paramètres des droites sont calculés de sorte que leur consistance statistique soit garantie.

Parmi les approches classiques utilisées en robotique mobile, citons les travaux de Arras *et al.* [8] et Castellanos *et al.* [50]. Arras *et al.* [8] utilisent le filtre de Sobel pour calculer la composante horizontale du gradient. Ensuite, une image binaire des contours est obtenue par sélection des extrema locaux suivi d'un seuillage par hystérésis. L'extraction des droites verticales à partir de l'image binaire se fait de la même façon que dans notre approche, par seuillage fixe d'un accumulateur (transformée de Hough). Le temps de calcul moyen correspondant au traitement des images n'est pas mentionné dans [8]. Nous remarquons que cette méthode n'utilise aucune information sur l'orientation du vecteur gradient. Nous pensons que cette information est importante si l'on veut conserver uniquement les points appartenant à un contour vertical. Imaginons l'image d'un mur fortement texturé. Dans ce cas, il existe des gradients suivant toutes les directions dans l'image. Par conséquent, pour certains points les composantes horizontales peuvent être importantes, même si les contours ne sont pas verticaux. Nous pouvons avoir alors de fausses détections si le seuil utilisé pour l'accumulateur est trop bas. Le choix de ce seuil étant délicat, nous proposons de prendre en compte l'orientation des gradients et de rajouter une étape intermédiaire de suppression des points dont le vecteur gradient n'est pas horizontal. Ainsi, les points de contours conservés dans l'image binaire correspondent à des pixels de fort gradient horizontal.

Castellanos *et alii* [50] ont implanté une technique proposée par Burns *et al.* [45]. Cette approche emploie un simple masque 2×2 pour estimer le gradient de l'image. L'image gradient est divisée dans des régions dont les pixels présentent gradient d'orientation similaire, en fonction de huit intervalles possibles entre 0° et 360° . Pour chaque région, une surface est approximée pour déterminer de façon optimale l'emplacement et la direction de chaque contour. Enfin, les attributs associés à chaque contour sont calculés. Castellanos *et al.* ont spécialisé cette approche pour la détection de contours verticaux. Ainsi, ils ont considéré seulement des régions dont l'orientation du gradient se situe dans les intervalles $[(-22, 5^\circ), (22, 5^\circ)]$ et $[(-157, 5^\circ), (202, 5^\circ)]$. D'autres travaux

en robotique mobile ont aussi utilisé des droites verticales issues d'images vidéo pour des applications diverses [161, 218, 10]. Cependant, les auteurs n'ont pas donné les détails d'implémentation.

Dans les sections qui suivent, nous présentons notre méthode en utilisant comme exemple l'image de la Figure 4.1(a). Cette image a été acquise dans un site réel et elle est composée de zones de différents niveaux d'illumination. Nous l'avons donc choisie pour présenter les performances de notre méthode dont le résultat est montré sur la Figure 4.1(b).

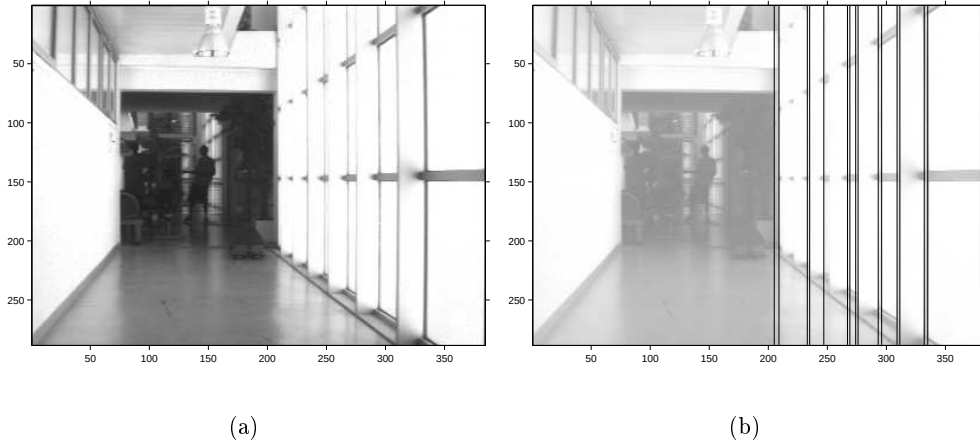


FIG. 4.1 – Exemple d'extraction de droites dans des images vidéo : (a) une image réelle et (b) les droites verticales extraites.

4.2.1 Détection de contours verticaux

Notre objectif est d'obtenir une méthode de détection de contours verticaux présentant un bon compromis efficacité/temps de calcul, comparativement à d'autres approches plus sophistiquées. Les de la détection des contours de l'image vidéo de la Figure 4.1(a) sont illustrées dans la Figure 4.2. Ces étapes comportent :

- I. *L'application du filtre de Canny.* Le filtre de Canny [48] est un filtre optimal de lissage/dérivation très utilisé pour la détection de contours. Les contours se traduisent par des pixels dans l'image qui ont une valeur élevée du gradient d'intensité [99]. Ce filtre s'applique à chaque pixel de coordonnées (u, v) de l'image. Etant séparable, il calcule les composantes $\nabla_u \mathcal{I}(u, v)$ et $\nabla_v \mathcal{I}(u, v)$ du vecteur gradient d'intensité associé au pixel (u, v) par l'application de masques monodimensionnels suivant les directions des axes $\mathcal{U}$ et $\mathcal{V}$ du repère image. Le module du gradient est donné par :

$$\|\nabla \mathcal{I}(u, v)\| = \sqrt{[\nabla_u \mathcal{I}(u, v)]^2 + [\nabla_v \mathcal{I}(u, v)]^2}. \quad (4.1)$$

Pour une implémentation plus rapide, nous l'avons approximé par

$$\|\nabla \mathcal{I}(u, v)\| \approx \|\nabla_u \mathcal{I}(u, v)\| + \|\nabla_v \mathcal{I}(u, v)\|. \quad (4.2)$$

L'image gradient $\|\nabla \mathcal{I}(u, v)\|$ obtenue par application du filtre de Canny est présentée sur la Figure 4.2(b). Pour une meilleure netteté de l'affichage, cette image a été normalisée. Les valeurs absolues des composantes normalisées $\nabla_u \mathcal{I}(u, v)$ et $\nabla_v \mathcal{I}(u, v)$ sont présentées sur les Figures 4.2(c) et 4.2(d) ;

- II. *La sélection de extrema locaux du gradient.* Cette procédure vise à supprimer de l'image gradient les pixels pour lesquels le module du gradient $\|\nabla \mathcal{I}\|$ n'est pas un maximum local. Elle permet de réduire l'épaisseur des lignes délimitant les contours, facilitant ainsi l'identification de leur emplacement dans l'image. Pour cela, on applique $\|\nabla \mathcal{I}(u, v)\| = 0$ à chaque pixel tel que $\|\nabla \mathcal{I}(u, v)\| < \|\widehat{\nabla \mathcal{I}}(u, v)\|$, où $\widehat{\nabla \mathcal{I}}(u, v)$ est une estimée du gradient calculée par interpolation à partir des gradients des pixels voisins. Une description détaillée de la procédure d'interpolation est présentée dans [175].

Le résultat de cette opération appliquée à l'image gradient de la Figure 4.2(b) est présenté sur la Figure 4.2(e) ;

- III. *La sélection des pixels appartenant à un contour vertical.* Cette étape est basée sur l'utilisation des composantes $\nabla_u \mathcal{I}(u, v)$ et $\nabla_v \mathcal{I}(u, v)$ pour déterminer l'orientation du vecteur gradient $\nabla_u \mathcal{I}(u, v)$ par rapport à l'horizontale. La suppression utilise le test suivant : si $\|\nabla_u \mathcal{I}(u, v) / \nabla_v \mathcal{I}(u, v)\| > \tan(\lambda)$, alors $\|\nabla \mathcal{I}(u, v)\| = 0$, où λ est un seuil qui correspond à la valeur absolue maximale de l'angle du vecteur gradient par rapport à l'horizontale.

La Figure 4.2(f) montre le résultat de l'application de cette procédure à l'image gradient de la Figure 4.2(e). Nous utilisons $\lambda = 10^\circ$;

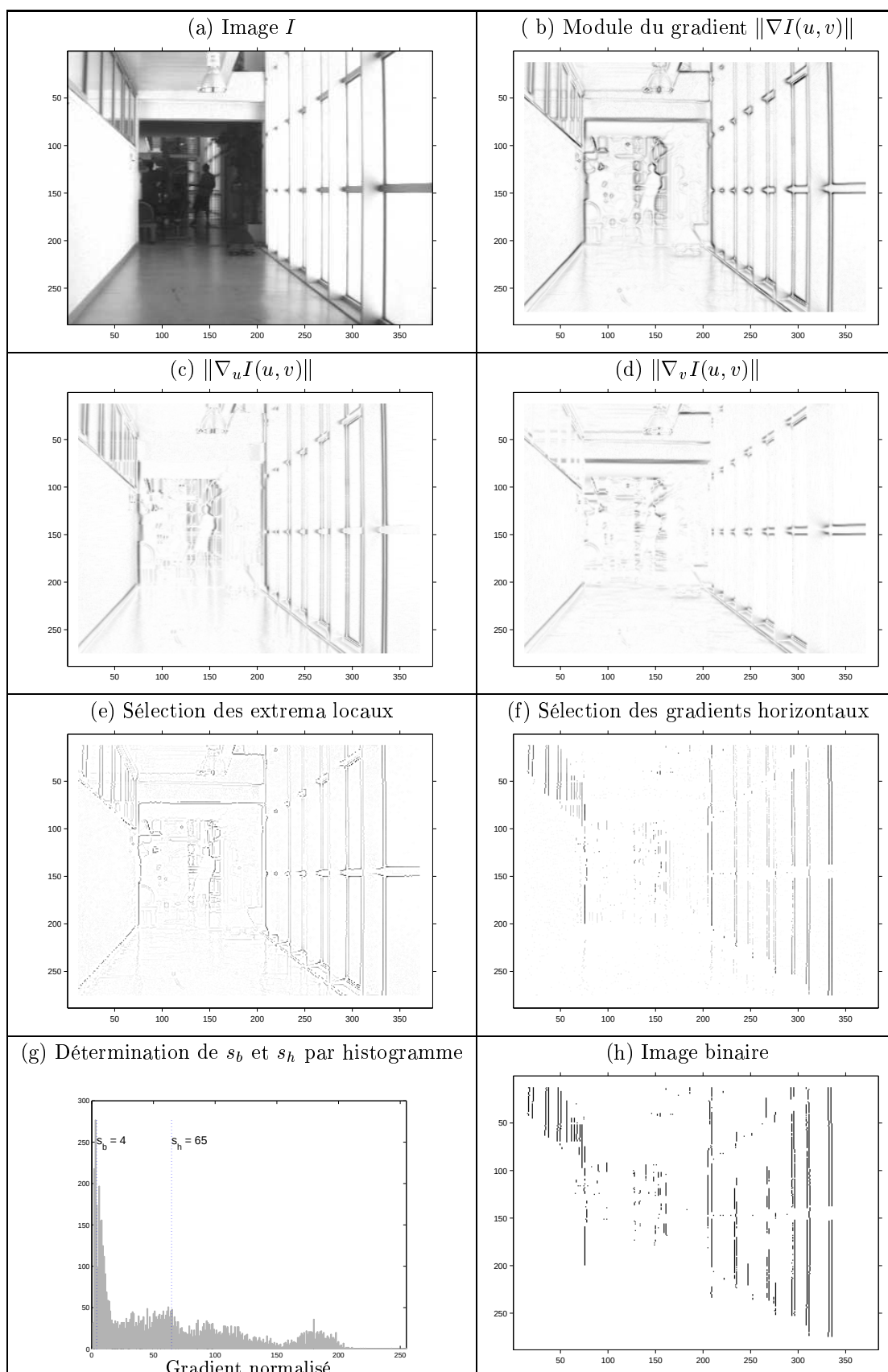
- IV. *Le seuillage par hystérésis* [99] fournit ensuite une image binaire $\mathcal{I}_b$, où la valeur de chaque pixel indique la présence ou non d'un contour vertical. L'approche la plus utilisée est celle proposée par Canny [48]. Elle fait appel à un seuil bas s_b et un seuil haut s_h déterminés par analyse de l'histogramme de l'image gradient 4.2(f) mise à l'échelle maximale. L'histogramme de la Figure 4.2(g) prend en compte seulement les pixels ayant un gradient différent de zéro. Le seuil haut s_h correspond à 60 % des pixels actifs et s_b correspond à 10 %. Cet algorithme, décrit dans [175], effectue une recherche itérative de la trace du contour.

Lorsque le niveau de luminosité n'est pas assez élevé, la mise à l'échelle de l'image gradient se traduit par une dégradation de l'histogramme et un échec de la détection (Figure 4.2(h)).

Des techniques de détection plus robustes existent [112], mais elles souffrent en général d'une complexité de calcul élevée. Pour le cas que nous venons de présenter, le temps de calcul total est de l'ordre de 240 ms. Nous n'avons pas appliqué de technique d'optimisation de code [135] à ces algorithmes.

4.2.2 Détection des droites verticales

Sous l'hypothèse que les pixels actifs de l'image binaire fournie par le traitement précédent (cf. section 4.2.1) correspondent seulement à des contours verticaux, la détection des droites verticales se fait à l'aide d'une transformée de Hough spécialisée. La transformée de Hough est une méthode d'accumulation très souvent utilisée pour l'extraction de droites dans des images binaires [190]. Nous l'avons adaptée pour la recherche de droites verticales. Ainsi, sur la Figure 4.3(a), les cellules



de l'accumulateur contiennent le nombre de pixels actifs pour chaque colonne de l'image binaire 4.2(h). Nous considérons que les colonnes correspondant à des droites verticales sont celles qui contiennent au moins 30 % de pixels actifs. Le choix d'un seuil fixe pour cette phase est justifié par la normalisation déjà effectuée lors de la binarisation de l'image gradient.

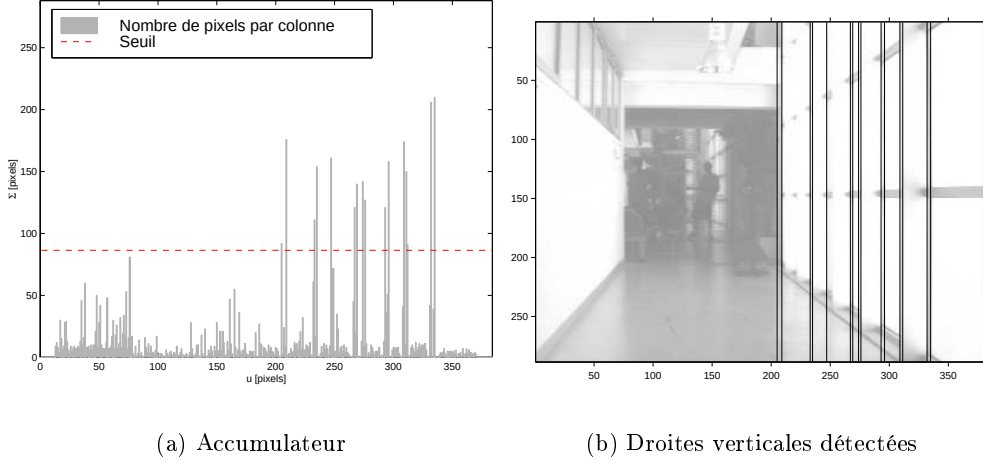


FIG. 4.3 – Extraction de droites verticales utilisant accumulateur

Une fois l'accumulateur construit, nous passons à l'identification des index de la colonne de début u_d et de la colonne de fin u_f pour chaque droite verticale. En effet, une droite verticale peut correspondre à plusieurs colonnes de l'accumulateur. Ainsi, u_d et u_f nous permettent de calculer les paramètres, dans le plan image, de la droite verticale représentée par

$$U = \{u, \mathbf{\Lambda}_u\}, \quad (4.3)$$

où u est la coordonnée suivant l'axe $\mathcal{U}$ de la colonne verticale supportant la droite, et $\mathbf{\Lambda}_u$ est sa covariance. Nous supposons que l'incertitude d'observation d'une droite verticale suit $\mathcal{N}(0, \mathbf{\Lambda}_u)$. Etant donné que $u \in \mathbb{R}$, ce paramètre implique une détection sub-pixelique des droites verticales. Pour réaliser cela, il nous a semblé, à première vue, que les formules suivantes étaient appropriées :

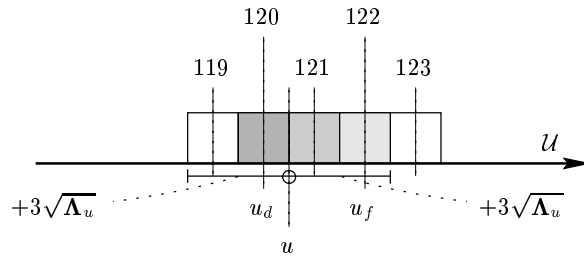
$$u = \frac{\sum_{x=u_d}^{u_f} x \cdot A(x)}{\sum_{x=u_d}^{u_f} A(x)}, \quad \mathbf{\Lambda}_u = \frac{\sum_{x=u_d}^{u_f} (u-x)^2 \cdot A(x)}{\sum_{x=u_d}^{u_f} A(x)}, \quad (4.4)$$

où $A(x)$ est la valeur de l'accumulateur pour la colonne x . Les équations (4.4) sont très souvent utilisées pour le calcul empirique de la moyenne et de la covariance de grandeurs pondérées. Dans notre cas, la pondération est liée au nombre de pixels actifs dans chaque colonne supportant la droite. Cependant, cette formule présente un risque d'inconsistance statistique. Cela peut être facilement vérifié dans le cas d'une droite dont le support est constitué d'une seule colonne, *i.e.*, $u_d = u_f$. Dans ce cas, nous avons $\mathbf{\Lambda}_u = 0$, qui implique qu'il n'y a pas d'incertitude sur u . Et pourtant, nous savons que la projection d'un indice vertical se trouve à l'intérieur de l'intervalle métrique occupé par la colonne en question. Donc, une incertitude au moins égale à la largeur de cet intervalle doit être définie de façon à garantir la consistance de $\mathbf{\Lambda}_u$. Nous proposons d'utiliser une estimée basée

sur un raisonnement 3σ (cf. annexe B.2) dans l'intervalle de mesure $[(u_d - 0, 5), (u_f + 0, 5)]$:

$$\Lambda_u = \left[\max(u_f - u + \frac{1}{2}, u - u_d + \frac{1}{2}) \cdot \frac{1}{3} \right]^2. \quad (4.5)$$

Dans l'équation (4.5), $\max(u_f - u + 1/2, u - u_d + 1/2)$ correspond à la distance maximale entre u et les frontières de l'intervalle 3σ . La constante $\frac{1}{3}$ correspond à l'écart-type 3σ pour un intervalle de largeur un pixel. Pour mieux illustrer ce calcul, nous faisons référence à la figure ci-dessous :



Cette figure correspond au cas $u_d = 120, u_f = 122$ et $u = 120, 5$. Ainsi, le calcul (4.5) donne $\Lambda_u = \frac{4}{9}$, qui est consistante sachant que l'intervalle 3σ correspondant à

$$\left[\left(u - 3\sqrt{\Lambda_u} \right), \left(u + 3\sqrt{\Lambda_u} \right) \right] = [(118, 5), (122, 5)]$$

qui incorpore toujours l'intervalle d'observation $[(u_d - 0, 5), (u_f + 0, 5)] = [(120, 5), (122, 5)]$. Ainsi, nous garantissons une estimation consistante de l'incertitude.

Le résultat final de l'extraction de droites verticales d'une image vidéo se présente sous la forme d'une liste

$$\Omega^U = \{U_k \mid k = 1, \dots, N_{\Omega^U}\}. \quad (4.6)$$

4.3 Segmentation d'images télémétriques

4.3.1 Indices d'intérêt dans les images télémétriques

Le système d'extraction de droites décrit dans cette section comporte deux phases : la détection de points de cassure et l'extraction de segments de droite. Ces indices nous permettent d'obtenir des informations sur la géométrie de l'environnement local, comme décrit ci-dessous :

- Les points de cassure sont en effet des points de l'image télémétrique associés à une discontinuité dans les mesures. Ces discontinuités peuvent avoir deux causes distinctes. La première est l'absence d'obstacles à portée du télémètre dans la l -ième direction ϕ_l . Dans ce cas, il n'y a pas de retour de signal et r_l est mis à zéro pour indiquer une mesure invalide. La deuxième cause est la présence d'une discontinuité sur la surface de balayée qui se traduit par une variation important entre r_{l-1} et r_l ;
- Les segments de droite sont supportés par un ensemble de points consécutifs de l'image laser, et résultent du balayage de surfaces planes de l'environnement.

Avant de présenter les procédures de segmentation des images télémétriques, nous introduisons une nouvelle représentation pour celles-ci. Rappelons que ces images sont représentées initialement en coordonnées polaires (voir section 3.2) :

$$\mathcal{L} = \{(r_l, \phi_l) | l = 1, \dots, N_{\mathcal{L}}\}, \quad (4.7)$$

où le pas angulaire du balayage est $\phi_l - \phi_{l-1} = \Delta\phi$. Pour les procédures qui suivent, nous utilisons la représentation cartésienne étendue :

$$\mathcal{C} = \{(\mathbf{p}_n, \kappa_n^r, \kappa_n^+, \kappa_n^-) | n = 1, \dots, N_{\mathcal{C}}; \mathbf{p}_n = (x_n, y_n)\}. \quad (4.8)$$

Dans cette représentation, $x_n = r_{l^*} \cdot \cos(\phi_{l^*})$ et $y_n = r_{l^*} \cdot \sin(\phi_{l^*})$ sont les coordonnées des points valides de l'image originale $\mathcal{L}^1$. La distance mesurée et l'angle de balayage sont obtenus à partir de ces coordonnées par

$$r_n = \sqrt{x_n^2 + y_n^2} \quad \text{et} \quad \phi_n = \arctan(y_n/x_n). \quad (4.9)$$

Les étiquettes κ_n^r , κ_n^+ et κ_n^- sont des variables logiques qui indiquent une discontinuité associée au n -ième point de balayage valide de l'image. κ^r identifie les points qui délimitent un intervalle dans lequel il y a eu *rupture* dans l'acquisition n (i.e., il n'y avait pas d'obstacles dans les directions correspondant à cet intervalle). Par exemple, si dans une image $\mathcal{L}$ les seuls points de $l = 21$ jusqu'à $l = 25$ ne sont pas valides, alors nous avons $N_{\mathcal{L}} - 5$ points valides indexés par $l^* = 1, \dots, 20, 26, \dots, N_{\mathcal{L}}$. Ces points sont utilisés pour générer les $N_{\mathcal{C}} = N_{\mathcal{L}} - 5$ points de l'image cartésienne $\mathcal{C}$ ayant comme index $n = 1, \dots, 20, 21, \dots, N_{\mathcal{C}}$. Par ailleurs, κ_{20}^r et κ_{21}^r sont mis à l'état VRAI.

Les points de cassure sont indiqués par les étiquettes κ_n^+ et κ_n^- . Ces étiquettes correspondent respectivement à des *points de cassures forts* et à des *points de cassure faibles*, définies ci-dessous :

Définition 4.1: Points de cassure faibles : *ce sont des points de l'image télémétrique pour lesquels la séquence d'acquisition présente une discontinuité.* $\diamond$

Définition 4.2: Points de cassure forts : *ce sont des points de cassure qui déterminent les limites physiques des surfaces balayées par le capteur.* $\diamond$

Par conséquent, κ_n^+ est une étiquette qui est activée lors que le n -ième point est un point de cassure fort, et κ_n^- a la même signification pour des points de cassure faibles.

Pour mieux illustrer les indices d'intérêt pour la segmentation d'images télémétriques, nous présentons une simulation sur la Figure 4.4(a). L'environnement simulé est constitué seulement de murs verticaux dont la vue de haut nous permet de mieux visualiser les indices d'intérêt. Dans cette simulation, le robot est à l'arrêt pendant la prise d'une image télémétrique. L'image et les indices obtenus par segmentation sont montrés sur la Figure 4.4(b). Les numéros affichés dans cette figure

¹Nous utilisons cette notation pour indiquer que ni tout point de balayage d'index l de l'équ. (4.7) est valide. Les points valides ont index l^* .

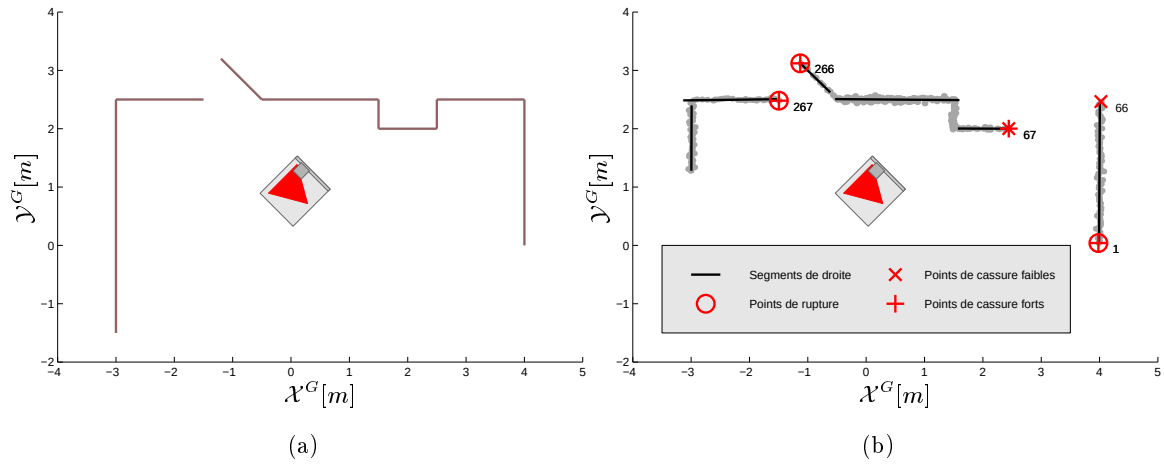


FIG. 4.4 – Illustration des indices d'intérêt pour la segmentation d'image télémétriques.

correspondent aux index n des points de balayage pour lesquels au moins une des étiquettes κ_n^r , κ_n^+ ou κ_n^- est à l'état **VRAI**. Nous soulignons qu'un point de l'image peut avoir plusieurs étiquettes actives en même temps, comme c'est le cas des points 1, 67, 266 et 267. Le point 1 est un point de rupture parce que les points de balayage qui le précédaient dans l'image originale $\mathcal{L}$ ont $r_l = 0$ à cause de l'absence d'obstacles dans les directions correspondantes. En conséquence, comme nous le verrons dans la prochaine section, le point 1 est aussi un point de cassure fort. C'est aussi le cas des points 266 et 267. Entre ces points, comme nous pouvons le constater, il a eu une rupture de la séquence d'acquisition par l'absence d'obstacles. La différence entre points de cassure forts et points de cassure faibles est plus évidente lorsque nous présentons le cas des points 66 et 67. Ces deux points sont marqués comme des points de cassure faibles parce qu'une discontinuité a été détectée entre eux. Comme nous pouvons le voir, cette discontinuité est due à une occlusion. Cependant, seul le point de cassure 67 est marqué comme point de cassure fort. Ceci s'explique par le fait que, pour être plus proche du capteur, seul ce point indique avec plus de fiabilité qu'il entoure une des extrémités de la surface sur laquelle le balayage s'est effectué. Cela n'est pas le cas du point 66, étant donné que la surface à laquelle appartient le point 67 occulte celle du point 66. Ainsi, nous incorporons dans l'image segmentée des informations qui concernent la présence d'occlusions dans le balayage. Les segments de droite extraits de la segmentation sont aussi illustrés dans la Figure 4.4(b).

4.3.2 Détection de points de cassure

La section précédente nous a permis de mieux comprendre la signification des discontinuités dans une image télémétrique. Ces discontinuités qualifiées de rupture, cassure forte et cassure faible, sont indiquées dans les points de l'image par les étiquettes κ_n^r , κ_n^+ et κ_n^- , respectivement. Nous avons vu que les points de rupture sont identifiables par une simple vérification des mesures fournies par le télémètre.

Dans cette section nous nous intéressons seulement à la détection des points de cassure. Leur détection est importante, non seulement pour aider à la construction de cartes locales (chapitre 5), mais aussi pour l'extraction de segments de droite (section 4.3.3). Sans l'aide des points de cassure, l'extracteur de segments de droite risque de fusionner deux segments voisins correspondant à des éléments coplanaires mais distincts de l'environnement. Dans le cas de deux points consécutifs $\mathbf{p}_{n-1}$ et $\mathbf{p}_n$ séparés par une grande distance, les étiquettes κ^+ et κ^- associées à ces points doivent indiquer la discontinuité. Bien entendu, cette notion de *distance entre points consécutifs* dépend localement de la région sur laquelle le balayage s'effectue.

Dans les sections suivantes, nous présentons deux algorithmes pour la détection de points de cassure. Ces algorithmes simples et intuitifs sont initialement utilisés pour la détection de points de cassure faibles. La section 4.3.2.3 présente la procédure d'identification des points de cassure forts à partir des points de cassure faibles et des points de rupture.

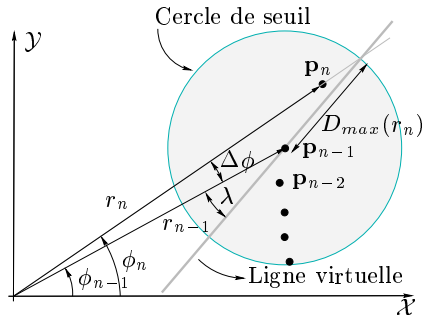
4.3.2.1 Détection de points de cassure faibles par seuillage adaptatif

Nous pouvons simplement imaginer qu'un détecteur de discontinuités pourrait être fondé sur un test tel que :

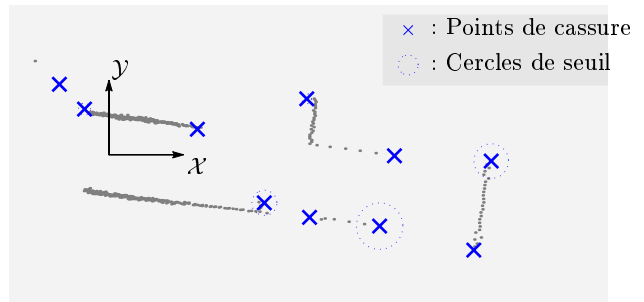
$$\text{si } \|\mathbf{p}_n - \mathbf{p}_{n-1}\| > D_{\max}, \text{ alors } \kappa_n^- := \text{VRAI} \text{ et } \kappa_{n-1}^- := \text{VRAI}, \quad (4.10)$$

où $\|\mathbf{p}_n - \mathbf{p}_{n-1}\|$ représente la distance euclidienne entre les points $\mathbf{p}_n$ et $\mathbf{p}_{n-1}$. Une discussion sur un tel détecteur et ses variantes a été présentée dans [193]. La difficulté principale avec ce détecteur reste la détermination du seuil $D_{\max}$.

Nous proposons dans cette section une approche intuitive pour déterminer $D_{\max}$, comme illustré par la Figure 4.5(a). Dans cette méthodologie, nous utilisons une ligne virtuelle qui passe par le point $\mathbf{p}_{n-1}$, faisant un angle λ avec la direction de balayage ϕ , pour représenter le cas extrême d'un segment droite à la limite de la faculté de détection fiable. Une telle ligne sert à extrapoler le point de balayage $\mathbf{p}_n^h$ sous l'hypothèse que la ligne virtuelle représente vraiment la surface sur laquelle le point $\mathbf{p}_{n-1}$ a balayé ($\mathbf{p}_n^h$ est n'importe quel point sur le cercle de seuil de la Figure 4.5(a)).



(a) Illustration du détecteur



(b) Exemple avec une image réelle

FIG. 4.5 – Détection de points de cassure par seuillage adaptatif.

Algorithme 1: Détecteur de points de cassure faibles par seuillage adaptatif.

```

1:    $\kappa_1^- \leftarrow \text{FAUX}$ 
2:   for  $n = 2$  to  $N_C$ 
3:      $D_{\max} \leftarrow r_{n-1} \cdot \frac{\sin(\Delta\phi)}{\sin(\lambda - \Delta\phi)} + 3\sigma_r$ 
4:     if  $\|\mathbf{p}_n - \mathbf{p}_{n-1}\| > D_{\max}$ 
5:       (Des points de cassure faibles ont été trouvés)
6:        $\kappa_n^- \leftarrow \text{VRAI}$ 
7:        $\kappa_{n-1}^- \leftarrow \text{VRAI}$ 
8:     else
9:        $\kappa_n^- \leftarrow \text{FAUX}$ 

```

Ainsi, comme la ligne virtuelle représente un cas extrême, nous pouvons déterminer la plus grande distance acceptable entre les points $\mathbf{p}_n$ and $\mathbf{p}_{n-1}$. Dans cette optique, la mesure hypothétique r_n^h correspondant à $\mathbf{p}_n^h$ est reliée à r_{n-1} par

$$r_{n-1} \cdot \sin(\lambda) = r_n^h \cdot \sin(\lambda - \Delta\phi). \quad (4.11)$$

En développant l'équation (4.11), nous trouvons que la distance euclidienne entre $\mathbf{p}_{n-1}$ et le point hypothétique $\mathbf{p}_n^h = (r_n^h \cdot \cos(\phi_n), r_n^h \cdot \sin(\phi_n))$ est

$$\|\mathbf{p}_n^h - \mathbf{p}_{n-1}\| = r_{n-1} \cdot \frac{\sin(\Delta\phi)}{\sin(\lambda - \Delta\phi)}. \quad (4.12)$$

A ce stade, nous pourrions utiliser $\|\mathbf{p}_n^h - \mathbf{p}_{n-1}\|$ comme seuil pour le test des points de cassure faibles (équation (4.10)). Néanmoins, un tel seuil ne prend pas en compte le bruit de mesure associé à r_n . Le test peut échouer en présence d'obstacles à proximité du capteur à cause d'une petite valeur de r_{n-1} . Donc, nous proposons d'utiliser comme seuil

$$D_{\max} = \|\mathbf{p}_n^h - \mathbf{p}_{n-1}\| + 3\sigma_r, \quad (4.13)$$

qui permet de prendre en compte la composante aléatoire de r_n . Intuitivement, le paramètre λ correspond au cas extrême de l'angle d'incidence du faisceau laser avec une droite dont la détection est encore fiable. Le seuil utilisé peut être représenté par un cercle centré en $\mathbf{p}_{n-1}$ de rayon $D_{\max}$ (voir Figure 4.5(a)). Si le point de balayage suivant $\mathbf{p}_n$ est en dehors de ce cercle, $\mathbf{p}_n$ et $\mathbf{p}_{n-1}$ sont marqués comme points de cassure faibles.

Nous avons sur la Figure 4.5(b) une image télémétrique réelle sur laquelle ce détecteur a été appliqué. Les paramètres utilisés sont $\sigma_r = 0,03 \text{ m}$, qui est l'écart-type des mesures de distance du télémètre, et $\lambda = 10^\circ$. Ces résultats semblent satisfaisants. Cependant, une évaluation expérimentale sera présentée dans la section 4.5. L'implémentation de ce détecteur est décrite par l'Algorithme 1.

4.3.2.2 Détection de points de cassure faibles par filtrage de Kalman

Un détecteur stochastique de points de cassure a été présenté par Castellanos *et al.* dans [51]. Un filtre de Kalman est utilisé en conjonction avec un test statistique pour vérifier si deux points consécutifs de l'image télémétrique appartiennent toujours à la même surface de la scène. Ce problème, formulé comme celui du suivi de cibles [15], est fondé sur un modèle dynamique ayant comme observations les mesures r_n . Si le suiveur échoue, alors cela signifie qu'une nouvelle région a été balayée. Nous proposons une version légèrement modifiée de ce détecteur pour les points de cassure faibles. Le modèle discret stochastique suivant décrit le comportement dynamique des mesures de distance r_n :

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{w}_n, \quad (4.14)$$

$$z_n = \mathbf{C}\mathbf{x}_n + \mathbf{v}_n. \quad (4.15)$$

$\mathbf{x}_n = (r_n \quad \frac{dr_n}{d\phi})^T$ est le vecteur d'état, z_n représente la variable de mesure scalaire. $\mathbf{w}_n$ et $\mathbf{v}_n$ sont des variables aléatoires gaussiennes de moyenne nulle et de matrice de covariance respectives $\mathbf{Q}_n$ et $\mathbf{R}_n$. Les matrices $\mathbf{A}$ et $\mathbf{C}$ correspondent à :

$$\mathbf{A} = \begin{pmatrix} 1 & \Delta\phi \\ 0 & 1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 \end{pmatrix}. \quad (4.16)$$

L'équation (4.14) est la plus simple expression du modèle dynamique qui puisse être utilisée sans connaissance préalable de la trajectoire formée par la séquence de points de balayage. Des suiveurs de cibles de ce genre sont analysés dans [15]. Ils supposent que la cible a une vitesse ou une accélération constante. Dans le cas de (4.14), de telles hypothèses ne sont pas acceptables, même pour des environnements structurés. Cela traduit la présence de composantes du mouvement non modélisables dans la trajectoire de la séquence de points. Dans ces cas, la matrice $\mathbf{Q}_n$ a une grande importance pour le maintien de la consistance du modèle, *i.e.*, ses termes doivent être suffisamment grands pour incorporer l'incertitude due aux composantes non modélisables du mouvement de la cible. Si cela n'est pas le cas, beaucoup de faux points de cassure sont détectés. Inversement, si $\mathbf{Q}_n$ est très surestimé, l'algorithme ne détecte pas de points de cassure, même s'ils existent. Une implémentation de ce détecteur est illustrée par l'Algorithme 2. A chaque pas de cet algorithme, un test du χ^2 est utilisé pour détecter si la n -ième mesure r_n est compatible avec l'état prédit $\hat{\mathbf{x}}_{n|n-1}$. Si le test échoue, les étiquettes de points de cassure faibles κ_{n-1}^- et κ_n^- sont alors mises à l'état VRAI, et le filtre est réinitialisé.

Comme nous venons de l'expliquer, maintenir la consistance du filtre de Kalman est fondamentale pour la détection de points de cassure. Le théorème E.1 (annexe E.1) nous donne les conditions nécessaires pour que la consistance du filtre linéaire soit garantie. En accord avec ce théorème, le réglage de ce filtre repose sur un choix approprié des matrices $\mathbf{P}_0$, $\mathbf{Q}_n$ et $\mathbf{R}_n$. Dans notre application, $\mathbf{R}_n$ est déjà modélisée par la caractérisation du capteur (section 3.5.1). Par contre, étant donné le comportement inconnu de r_n pour des environnements quelconques, nous ne pouvons pas garantir que la valeur de $\mathbf{Q}_n$ soit consistante. Cependant, nous avons vérifié que plus l'incertitude initiale $\mathbf{P}_0$ est importante, plus longtemps $\hat{\mathbf{P}}_n$ reste consistant. Partant de ce

Algorithme 2: Détection de points de cassure faibles utilisant le filtre de Kalman.

```

1:    $n \leftarrow 1$  (Initialiser l'index du point courant de l'image)
2:    $n_i \leftarrow 1$  (Initialiser l'index du dernier point de cassure détecté)
3:    $\kappa_n^- \leftarrow \text{FAUX}$  pour  $n = 1, \dots, N_C$  (Initialiser les étiquettes)
4:   while  $n \leq N_C$ 
5:     if  $n_i = n$ 
6:        $\hat{\mathbf{x}}_n \leftarrow \mathbf{x}_0$  (Initialisation du filtre)
7:        $\hat{\mathbf{P}}_n \leftarrow \mathbf{P}_0$ 
8:     else
9:        $\hat{\mathbf{x}}_{n|n-1} \leftarrow \hat{\mathbf{x}}_{n-1}$  (Prédiction)
10:       $\hat{\mathbf{P}}_{n|n-1} \leftarrow \mathbf{A} \cdot \hat{\mathbf{P}}_n \cdot \mathbf{A}^T + \mathbf{Q}_n$ 
11:       $v_n \leftarrow r_n - \mathbf{C} \cdot \hat{\mathbf{x}}_{n|n-1}$  (Mesure)
12:       $S_n \leftarrow \mathbf{C} \cdot \hat{\mathbf{P}}_{n|n-1} \cdot \mathbf{C}^T + \mathbf{R}_n$ 
13:      if  $v_n^2/S_n \geq 3,84$  (Test  $\chi_1^2$  de compatibilité)
14:        (Des points de cassure faibles ont été trouvés)
15:         $\kappa_n^- \leftarrow \text{VRAI}$  and  $\kappa_{n-1}^- \leftarrow \text{VRAI}$ 
16:         $n_i \leftarrow n$ 
17:         $n \leftarrow n - 1$ 
18:      else
19:        (Le modèle est consistant. Corriger l'état prédit)
20:         $\mathbf{K}_n \leftarrow \hat{\mathbf{P}}_{n|n-1} \cdot \mathbf{C}^T \cdot S_n^{-1}$ 
21:         $\hat{\mathbf{x}}_n \leftarrow \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n \cdot v_n$ 
22:         $\hat{\mathbf{P}}_n \leftarrow \hat{\mathbf{P}}_{n|n-1} - \mathbf{K}_n \cdot \mathbf{C} \cdot \hat{\mathbf{P}}_{n|n-1}$ 
23:       $n \leftarrow n + 1$ 

```

principe, nous proposons d'appliquer le concept de ligne virtuelle pour initialiser le filtre chaque fois qu'un point de cassure est détecté. Ainsi, nous supposons que la ligne virtuelle passe par le n -ième point de balayage, et qu'elle fait un angle λ avec la direction ϕ_n . Les coordonnées polaires (r_n, ϕ_n) du n -ième point satisfont :

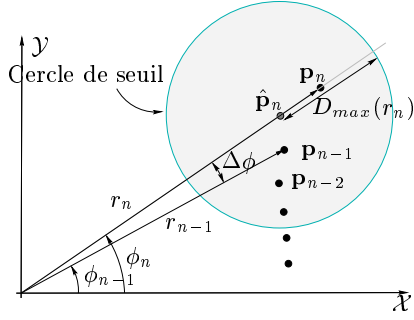
$$r_n \cdot \cos(\alpha - \phi_n) = \text{constante},$$

où α est l'angle de direction du vecteur normal à la ligne virtuelle. Comme $\alpha - \phi_n = \lambda - \pi/2$, nous avons :

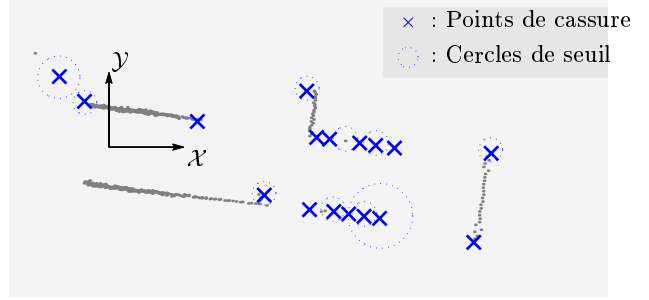
$$\frac{dr_n}{d\phi} = \cot(\lambda) \cdot r_n. \quad (4.17)$$

L'équation (4.17) peut être interprétée comme une borne supérieure pour $dr_n/d\phi$. La borne inférieure est alors : $-\cot(\lambda) \cdot r_n$. Des telles bornes impliquent que la mesure r_{n+1} associée au point de balayage suivant soit dans l'intervalle $[r_n - \cot(\lambda) \cdot r_n \cdot \Delta\phi, r_n + \cot(\lambda) \cdot r_n \cdot \Delta\phi]$. Sans connaissance *a priori* du comportement de la séquence de points de balayage qui débute au n -ième point, nous initialisons l'état du filtre avec

$$\mathbf{x}_0 = \begin{pmatrix} r_n & 0 \end{pmatrix}^T.$$



(a) Illustration du détecteur



(b) Exemple avec une image réelle

FIG. 4.6 – Détection de points de cassure utilisant l'approche basée sur le filtre de Kalman.

A partir des bornes introduites par la ligne virtuelle, nous proposons une initialisation empirique de la matrice de covariance :

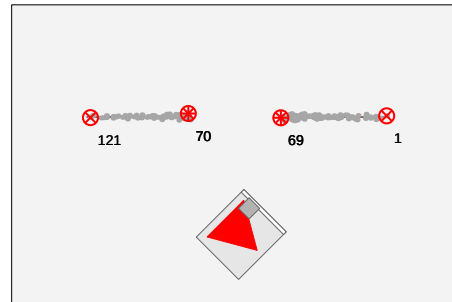
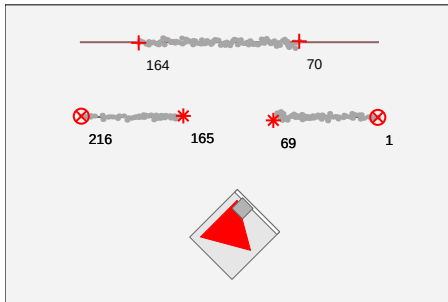
$$\mathbf{P}_0 = \frac{1}{3} \begin{pmatrix} |\cot(\lambda) \cdot r_n \cdot \Delta\phi| & 0 \\ 0 & |\cot(\lambda) \cdot r_n| \end{pmatrix} \quad (4.18)$$

Dans notre implémentation finale, nous avons utilisé $\mathbf{Q}_n = (0, 1)^2 \cdot \mathbf{I}_2$.

Ce détecteur utilise un seuil qui peut être modélisé par un cercle centré en $\hat{p}_n = (\hat{r}_n, \phi_n)$ et de rayon $D_{\max} = \sqrt{3.84 \cdot S_n}$ (voir Fig. 4.6(a)). Dans la Figure 4.6(b), nous avons une image télémétrique réelle sur laquelle ce détecteur de points de cassure a été appliqué. Les paramètres utilisés dans cet exemple étaient $\mathbf{R}_n = \sigma_r^2 = (0, 03 \text{ m})^2$, et $\lambda = 10^\circ$. Les points de cassure obtenus semblent satisfaisants. Par rapport au détecteur adaptatif (voir Figure 4.5(b)), celui-ci fournit un peu plus de points de cassure. Une comparaison plus complète des deux approches est présentée dans la section 4.5.2.1.

4.3.2.3 Détection des points de cassure forts

Les points de cassure forts sont sensés représenter les limites physiques des surfaces balayés par le télémètre. Le détecteur est construit à partir d'un raisonnement simple. Les simulations ci-dessous illustrent bien ce raisonnement :



Nous avons simulé à gauche un robot situé dans un environnement composé de deux murs avec un passage entre eux. En face du passage, il y a un autre mur partiellement balayé par le télémètre. A droite nous avons la même configuration sans mur derrière le passage. Sur images télémétriques simulées, les points de rupture, les points de cassure forts et les points de cassure faibles sont indiqués respectivement par les symboles (o), (+) et ($\times$). Nous pouvons remarquer que certains de ces points appartiennent à plusieurs classes de discontinuité. Les points 1 et 216 (resp. 1, 69, 70 et 121) de l'image gauche (resp. droite) sont des points de rupture parce qu'il n'y a pas d'obstacle dans leur voisinage. Ils deviennent aussi des points de cassure forts une fois que nous pouvons affirmer qu'ils indiquent des extrémités de la surface balayée. Cependant, étant donné l'occlusion du mur en face du passage (dans la simulation de gauche), nous ne pouvons pas affirmer que les points 70 et 164 indiquent les extrémités de cet obstacle. Et pourtant, ces mêmes points et leurs voisins, les points 69 et 165, sont des points de cassure faibles, comme indiqué par le premier détecteur (section 4.3.2.1). Parmi ces points, seulement ceux d'index 69 et 165 correspondent bien à des extrémités sûres de surfaces de balayage. Partant de ces deux exemples, nous proposons une règle simple pour déterminer si le n -ième point correspond à un point de cassure fort. Ce n -ième point de balayage est un point de cassure fort si au moins une des deux conditions suivantes est vérifiée :

- I. S'il est un point de rupture, *i.e.* $\kappa_n^r = \text{VRAI}$;
- II. S'il est un point de cassure faible ayant la plus courte distance de mesure r parmi ses voisins à $n - 1$ ou à $n + 1$, si ceux-ci sont des points de cassure faibles. Cette condition exploite la propriété des points de cassure faibles qui ont toujours un voisin, qu'il soit le précédent ou le suivant.

Bien entendu, la simplicité d'une telle règle, fondée sur des cas très simples, nous laisse méfiants quant à son efficacité lors de configurations plus complexes. En effet, il y a des contre-exemples comme celui que nous montrons dans la section 4.5.2.1 de l'évaluation expérimentale.

4.3.3 Extraction de segments de droite

Dans ce système, nous employons la démarche division/fusion lors de l'application d'algorithmes classiques d'extraction de droites. L'objectif est de réduire le temps de calcul et d'éviter de fusionner des nuages de points voisins mais distincts. Ainsi, au lieu d'appliquer un algorithme d'extraction de segments dans une image complète, nous la divisons en régions continues sur lesquelles la segmentation est appliquée. Une approche similaire a été utilisée par Castellanos *et al.* [51]. Les régions continues sont définies ainsi :

Définition 4.3: Une région continue de l'image télémétrique est formée par l'ensemble de points télémétriques $\mathbf{p}_{n_i}, \mathbf{p}_{n_i+1}, \dots, \mathbf{p}_{n_e}$, ordonnés suivant leur acquisition ($n_e \geq n_i$), pour lesquels seulement $\mathbf{p}_{n_i}$ et $\mathbf{p}_{n_e}$ sont des points de cassure forts ou des points de cassure faibles. $\diamond$

Nous construisons de façon incrémentale une liste de segments de droites

$$\Omega^S = \{S_k \mid k = 1, \dots, N_{\Omega^S}\}.$$

Dans cette liste, S_k correspond à la représentation stochastique d'un indice, composée des attributs suivants :

$$S_k = \{\mathbf{l}_k, \mathbf{\Lambda}_{\mathbf{l}_k}, \mathbf{p}_{a_k}, \kappa_{a_k}^+, \mathbf{p}_{b_k}, \kappa_{b_k}^+\}, \quad (4.19)$$

où $\mathbf{l}_k = (\rho_k, \alpha_k)^T$ sont les paramètres polaires de la k -ième droite tels que $\rho_k \geq 0$ et $0 \leq \alpha_k \leq 360^\circ$ et $\mathbf{\Lambda}_{\mathbf{l}_k}$ est la matrice de covariance associée à $\mathbf{l}_k$. Ces paramètres correspondent à la représentation stochastique de la droite infinie associée au segment. $\mathbf{p}_{a_k}$ et $\mathbf{p}_{b_k}$ sont les coordonnées des extrémités du segment de droite, et $\kappa_{a_k}^+$ et $\kappa_{b_k}^+$ sont des étiquettes qui indiquent si ces extrémités ont été obtenues à partir de points de cassure forts. Pour une description plus détaillée, voir annexe C.1.3.

L'extraction incrémentale de segments de droite dans une image formée de régions continues est illustrée par l'Algorithme 3. Un noyau d'extraction de droites est appliqué à chaque région. Les points de balayage qui correspondent aux extrémités d'une région sont identifiés par leurs étiquettes κ^+ ou κ^- . Pour chaque région trouvée, qui doit être composée d'au moins $N_{\min}$ points de balayage, une liste intermédiaire de segments Ω_*^S est construite par l'application d'un algorithme appelé noyau Φ . Le noyau est en effet un algorithme d'extraction de droites, qui pourrait être utilisé pour extraire des segments de l'image complète indépendamment de l'existence de points de cassure. Cependant, notre objectif ici est de réduire le temps de calcul et d'éviter de fusionner des nuages de points alignés voisins mais qui se trouvent dans des régions différentes. Avec la technique présentée ici, des algorithmes classiques d'extraction de droites peuvent être utilisés sans qu'il soit nécessaire d'adapter leur structure pour que les points de cassure soient pris en compte. Quelques noyaux pour l'extraction de droites sont présentés dans la section 4.3.4. Une fois les segments déterminés, une procédure de sélection est appliquée. Cette procédure fait le tri des segments, en ne gardant que ceux qui satisfont les critères suivants :

- Critère sur la longueur : tous les segments doivent avoir une longueur minimale $l_{\min}$;
- Critère sur le nombre de points supports : le nombre de points du support doit être supérieur à $N_{\min}$;
- Critère de continuité du support : la séquence de points du support ne doit pas contenir de point de cassure faible. Ce critère est assuré par l'application du premier détecteur (section 4.3.2.1) à la séquence de points considérée.

4.3.4 Noyaux pour l'extraction de segments

En général, les noyaux sont des algorithmes qui, de façon itérative, partitionnent un ensemble de points en groupes caractérisés par un grand moment d'inertie. Ces groupes sont nommés nuages de points linéaires (de l'anglais, *linear clusters*). Dans le domaine de la reconnaissance de formes, il existe des nombreux algorithmes destinés à extraire ce genre d'indice, notamment les algorithmes basés sur les C-moyennes floues. En robotique mobile, ces algorithmes n'ont pas trouvé d'adeptes pour l'extraction d'indices dans des images télémétriques.

Dans cette section, nous présentons les algorithmes noyaux pour l'extraction de droites. Nous avons généralisé leur présentation pour permettre une unification de la notation (cf. section 4.3.4.1). La différence principale entre les noyaux repose donc dans la façon dont les partitions sont déterminées. Nous avons choisi deux approches classiques en robotique mobile [191][210][51][75], présentées

Algorithme 3: Extraction de droites à partir de régions continues.

```

1:    $\Omega^S \leftarrow \emptyset$  (Liste initialisée vide)
2:    $n_e \leftarrow 1$  (Initialiser la limite finale)
3:   while  $n_e \leq N_C$ 
4:      $n_i \leftarrow n_e$  (L'index du premier point de la région)
5:      $n_e \leftarrow n_i + 1$  (Recherche l'index du dernier point à partir de  $n_i + 1$ )
6:     while  $\kappa_{n_e}^+ = \text{FAUX}$  and  $\kappa_{n_e}^- = \text{FAUX}$  (Test de région continue)
7:        $n_e \leftarrow n_e + 1$ 
8:       if  $n_e = N_C$  (Faire attention à la fin de l'image)
9:         break
10:    if  $(n_e - n_i + 1) > N_{\min}$ 
11:       $\Omega_*^S \leftarrow \Phi(I^T, n_i, n_e)$  (Liste de segments de la région courante)
12:       $\Omega^S \leftarrow \Omega^S \cup \Omega_*^S$  (Ajouter les segments extraits dans la liste principale)
13:    SÉLECTION( $\Omega^S$ ) (Appliquer les critères de sélection).

```

dans les sections 4.3.4.2 et 4.3.4.3. La section 4.3.4.4 décrit notre approche originale basée sur les C-moyennes floues pour l'extraction de droites. Cette approche hérite d'une difficulté classique des C-moyennes floues qui est le besoin de connaître *a priori* le nombre d'indices. Nos études nous ont permis de développer une variante des C-moyennes floues qui évite cette difficulté [32]. Ce nouvel algorithme intitulé *SMF* (de l'anglais *Split-and-Merge Fuzzy*) présenté dans la section 4.3.4.5.

4.3.4.1 Calcul unifié des paramètres

Dans cette section, nous présentons une formulation unifiée pour déterminer les paramètres des segments de droite S_k (équation (4.19)).

Paramètres de la droite infinie Un noyau d'extraction de droites calcule les paramètres ρ_k et α_k de la k -ième droite de façon à minimiser la fonction de coût pondérée suivante :

$$J_k(\rho_k, \alpha_k, Q, U) = \sum_{n|q_n=k} u_n^m d^2(x_n, y_n, \rho_k, \alpha_k), \quad (4.20)$$

où m est une constante positive et

$$d(x_n, y_n, \rho_k, \alpha_k) = \rho_k - x_n \cos(\alpha_k) - y_n \sin(\alpha_k) \quad (4.21)$$

est la distance algébrique du n -ième point $\mathbf{p}_n = (x_n, y_n)^T$ de l'image $\mathcal{C}$ à la k -ième droite de paramètres $\mathbf{l}_k = (\rho_k, \alpha_k)^T$. Les partitions déterminées par les noyaux sont modélisées par les structures $Q = \{q_n, n = 1, \dots, N_C\}$ et $U = \{u_n, n = 1, \dots, N_C\}$. Dans Q , q_n contient l'index k de la droite pour laquelle le n -ième point de l'image est un des points de support. Dans U , u_n est le poids associé à la correspondance entre le n -ième point et la q_n -ième droite. Pour des approches de partitionnement itératif, comme c'est le cas des C-moyennes floues, un point de

l'image peut être associée à plusieurs droites, mais avec des poids différents. Ainsi, nous avons C droites, $U = \{u_{nc}, n = 1, \dots, N_C \text{ et } c = 1, \dots, C\}$, où u_{nc} est le poids de l'appartenance du n -ième point à la c -ième droite. Dans ce cas, q_n est déterminé à la fin des itérations comme l'index de la droite pour laquelle le n -ième point a le plus grand poids de correspondance.

Nous pouvons calculer les paramètres $\mathbf{l}_k$ qui minimisent la fonction de coût unifiée (4.20) ainsi que la matrice de covariance associée $\mathbf{\Lambda}_{\mathbf{l}_k}$. Partant de (4.20), nous pouvons montrer que cette fonction est convexe et que son minimum satisfait $\partial J_k / \partial \rho_k = \partial J_k / \partial \alpha_k = 0$. Ainsi, nous avons

$$\rho_k = \tilde{x}_k \cos(\alpha_k) + \tilde{y}_k \sin(\alpha_k), \quad (4.22)$$

$$\alpha_k = \frac{1}{2} \arctan \left(\frac{-2\tilde{S}_{xy_k}}{\tilde{S}_{yy_k} - \tilde{S}_{xx_k}} \right), \quad (4.23)$$

où

$$\tilde{x}_k = \frac{\sum_{n|q_n=k} u_n^m x_n}{\sum_{n|q_n=k} u_n^m}, \quad \tilde{y}_k = \frac{\sum_{n|q_n=k} u_n^m y_n}{\sum_{n|q_n=k} u_n^m}, \quad (4.24)$$

$$\tilde{S}_{xx_k} = \sum_{n|q_n=k} u_n^m (x_n - \tilde{x}_k)^2, \quad \tilde{S}_{yy_k} = \sum_{n|q_n=k} u_n^m (y_n - \tilde{y}_k)^2, \quad (4.25)$$

$$\tilde{S}_{xy_k} = \sum_{n|q_n=k} u_n^m (x_n - \tilde{x}_k) \cdot (y_n - \tilde{y}_k). \quad (4.26)$$

et $m \geq 1$ est un facteur positif. Dans ces équations, $n | q_n = k$ indique que les sommes sont effectuées seulement pour les points indexés par n pour lesquels $q_n = k$, *i.e.*, le point n fait partie du support de la k -ième droite. Nous remarquons que $\tilde{x}_k$ et $\tilde{y}_k$ sont les coordonnées pondérées du centre de gravité du nuage de points qui supporte la droite. En particulier, lorsque $u_n^m = 1$, les équations (4.22) et (4.23) sont la solution du critère moindres-carrés classique.

Incertitude La matrice de covariance $\mathbf{\Lambda}_{\mathbf{l}_k}$ est calculée en utilisant la méthode de Haralick [93] (voir section B.4). Cette méthode permet de propager l'incertitude des régresseurs (*i.e.*, les entrées (x_n, y_n)) vers les estimées qui minimisent une fonction de coût de la forme (4.20). Elle permet aussi de prendre en compte les résidus d'estimation qui sont modélisés comme les variables aléatoires échantillonnées d'une distribution gaussienne ayant une moyenne nulle et une covariance $\mathbf{\Lambda}_{\mathbf{d}}$ estimée par

$$\mathbf{\Lambda}_{\mathbf{d}} = \frac{\sum_{n|q_n=k} u_n^m d^2(x_n, y_n, \rho_k, \alpha_k)}{\sum_{n|q_n=k} u_n^m} = \frac{J_k(\rho_k, \alpha_k, Q, U)}{\sum_{n|q_n=k} u_n^m}. \quad (4.27)$$

La matrice de covariance $\mathbf{\Lambda}_{\mathbf{p}_n}$ associée au n -ième point de balayage est calculée à partir de la variance σ_r^2 du capteur dans la direction du balayage ϕ_n comme

$$\mathbf{\Lambda}_{\mathbf{p}_n} = \sigma_r^2 \cdot \begin{pmatrix} \cos(\phi_n) \\ \sin(\phi_n) \end{pmatrix} \cdot \begin{pmatrix} \cos(\phi_n) & \sin(\phi_n) \end{pmatrix}, \quad (4.28)$$

Ainsi, nous considérons seulement les erreurs associées à la mesure de distance. En accord avec [93], nous avons

$$\mathbf{\Lambda}_{\mathbf{l}_k} = \left(\frac{\partial \mathbf{g}_k}{\partial \mathbf{l}_k} \right)^{-1} \cdot (\mathbf{C}_{\mathbf{d}} + \mathbf{C}_{\mathbf{p}}) \cdot \left(\frac{\partial \mathbf{g}_k}{\partial \mathbf{l}_k} \right)^{-1}, \quad (4.29)$$

avec $\mathbf{g}_k = \partial J_k / \partial \mathbf{l}_k$ et

$$\mathbf{C}_d = \sum_{n|q_n=k} \frac{\partial \mathbf{g}_k}{\partial d(x_n, y_n, \rho_k, \alpha_k)} \cdot \Lambda_d \cdot \frac{\partial \mathbf{g}_k^T}{\partial d(x_n, y_n, \rho_k, \alpha_k)}, \quad (4.30)$$

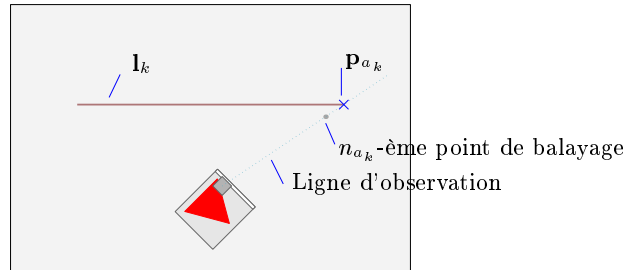
$$\mathbf{C}_p = \sum_{n|q_n=k} \frac{\partial \mathbf{g}_k}{\partial \mathbf{p}_n} \cdot \Lambda_p \cdot \frac{\partial \mathbf{g}_k^T}{\partial \mathbf{p}_n}. \quad (4.31)$$

Extrémités L'acquisition séquentielle des données télémétriques peut être exploitée pour faciliter la détermination des extrémités des segments. Ainsi, nous proposons dans un premier temps de déterminer les index n_{a_k} et n_{b_k} des points de balayage correspondant aux extrémités du k -ième segment par les heuristiques suivantes :

$$n_{a_k} = \min_n (n \mid q_n = k), \quad (4.32)$$

$$n_{b_k} = \max_n (n \mid q_n = k). \quad (4.33)$$

Une fois n_{a_k} (resp. n_{b_k}) déterminé, le point d'extrémité $\mathbf{p}_{a_k}$ (resp. $\mathbf{p}_{b_k}$) est calculé comme étant l'intersection entre les droites infinies données par $\mathbf{l}_k$, qui est la droite extraîte, et la ligne d'observation du n_{a_k} -ième (resp. n_{b_k} -ième) point du balayage. Cette procédure est illustrée ci-dessous :



Les étiquettes $\kappa_{a_k}^+$ et $\kappa_{b_k}^+$ héritent respectivement des valeurs des étiquettes des points de cassure forts $\kappa_{n_{a_k}}^+$ et $\kappa_{n_{b_k}}^+$, respectivement.

4.3.4.2 Line tracking (LT)

Le noyau “*line tracking*” (*LT*) est une méthode simple de mise à jour de la droite fondée sur un traitement séquentiel des points de balayage. Il a été utilisé dans [191] et [210]. Ainsi, partant des deux premiers points $\mathbf{p}_{n_i}$ et $\mathbf{p}_{n_i+1}$ de la région, cet algorithme initialise la k -ième droite comme étant celle qui passe par ces points. Si la distance T_{n_i+2} du prochain point $\mathbf{p}_{n_i+2}$ à la k -ième droite est inférieure à un certain seuil $T_{\max}$, alors $\mathbf{p}_{n_i+2}$ est accepté comme point support de cette droite dont les paramètres sont mis à jour en utilisant les points supports déjà identifiés. Cela se fait au sens des moindres carrés classiques en utilisant la formulation unifiée. Cette procédure se répète tant que $T_n < T_{\max}$. La k -ième droite est terminée lorsque nous avons $T_{n_a} > T_{\max}$ pour un



FIG. 4.7 – Illustration des noyaux classiques d'extraction de droites.

certain point $\mathbf{p}_{n_a}$. Alors, une nouvelle droite est initialisée avec les points supports $\mathbf{p}_{n_a}$ et $\mathbf{p}_{n_a+1}$. L'algorithme se termine lorsque l'autre extrémité $\mathbf{p}_{n_e}$ de la région est atteinte.

Cette approche est très rapide, mais comme elle ne prend pas en compte les résidus de la droite déjà ajustée, l'algorithme a tendance à incorporer des nuages de points curvilignes comme points supports de la droite.

4.3.4.3 Iterative end-point fit (IEPF)

Un algorithme très connu pour l'extraction de droites est l'“*iterative end point fit*” (IEPF) [73]. Lors de la première itération, cette approche divise l'ensemble de points $P = \{\mathbf{p}_{n_i}, \mathbf{p}_{n_i+1}, \dots, \mathbf{p}_{n_e}\}$ d'une région en deux sous-ensembles $P' = \{\mathbf{p}_{n_i}, \dots, \mathbf{p}_{n_a}\}$ et $P'' = \{\mathbf{p}_{n_a}, \dots, \mathbf{p}_{n_e}\}$ si un critère de validation n'est pas satisfait. $\mathbf{p}_{n_a}$ est le point pour lequel la distance T_{n_a} à la droite qui passe par les extrémités $\mathbf{p}_{n_i}$ et $\mathbf{p}_{n_e}$ de P est maximale. Le critère de validation est $T_{n_a} \leq T_{\max}$. Cet algorithme peut être implémenté de manière récursive. Etant donné sa simplicité, l'algorithme IEPF et ses variantes sont très souvent utilisés pour l'extraction de droites dans des images télémétriques en robotique mobile [51] [75].

4.3.4.4 C-moyennes prototypées

Les algorithmes du type C-moyennes cherchent à partitionner de façon itérative un ensemble P de N points en C classes. L'appartenance de chaque point aux différentes classes est définie par $U = \{u_{nc}, n = 1, \dots, N \text{ et } c = 1, \dots, C\}$, où u_{nc} indique l'appartenance du n -ième point à la c -ième classe. Dans le cas des C-moyennes classiques, u_{nc} est une valeur binaire indiquant si l'appartenance est vérifiée ou pas, *i.e.*, $u_{nc} = 1$ ou 0 . Pour les C-moyennes floues, $u_{nc} \in \mathbb{R}$ est une valeur qui correspond au degré d'appartenance du n -ième point à la c -ième classe. Chaque classe est représentée par un prototype β_c , d'où le nom de C-moyennes prototypées [123]. La détermination des prototypes et des degrés d'appartenance se fait par la minimisation en alternance d'une fonction de coût J . La plupart de ces algorithmes utilise un critère de la forme

$$J(\beta, U) = \sum_{c=1}^C \sum_{n=1}^N u_{nc}^m d^2(x_n, y_n, \beta_c). \quad (4.34)$$

Dans l'équation (4.34), $\beta = \{\beta_c | c = 1, \dots, C\}$ représente les paramètres des C prototypes. $d(x_n, y_n, \beta_c)$ est une fonction de distance entre le point (x_n, y_n) et le prototype β_c , et m est une constante. Pour l'algorithme des C-moyennes floues, les contraintes suivantes sont imposées :

$$u_{nc} \in [0, 1], \quad 0 < \sum_{n=1}^N u_{nc} < N, \quad \forall c, \quad (4.35)$$

$$\sum_{c=1}^C u_{nc} = 1, \quad \forall n. \quad (4.36)$$

Pour l'extraction de droites, il existe différentes représentations des prototypes sous la forme de matrices de covariance [19] [80]. Pour l'approche présentée ici, nous avons développé une représentation plus compacte. Cette représentation est donnée par $\beta_c = (\rho_c, \alpha_c)$, paramètres polaires de la c -ième droite. Etant donné la représentation polaire d'une droite $\rho = x \cos(\alpha) + y \sin(\alpha)$ [224] (voir annexe C.1.2), notre choix pour la fonction de distance d est

$$d^2(x_n, y_n, \beta_c) = f^2(x_n, y_n, \beta_c) + g^2(x_n, y_n, \beta_c), \quad (4.37)$$

dans laquelle $f(x_n, y_n, \beta_c) = (\rho_c - x_n \cos(\alpha_c) - y_n \sin(\alpha_c))$, et $g(x_n, y_n, \beta_c)$ est une fonction de pénalité pour des points qui sont très éloignés du centre du nuage de points qui supporte le c -ième prototype.

Dans le partitionnement itératif, la minimisation de J (équation (4.34)) considère aussi les contraintes (4.35) et (4.36). Ainsi, en utilisant la méthode des multiplicateurs de Lagrange, la minimisation sous contraintes de J est équivalente à la minimisation de

$$V(\beta, U) = J(\beta, U) + \sum_{n=1}^N \lambda_n \left(1 - \sum_{c=1}^C u_{nc} \right), \quad (4.38)$$

dans laquelle les λ_j sont les multiplicateurs de Lagrange. V est minimisée en utilisant la technique d'optimisation en alternance. Ainsi, dans la phase de partitionnement, on considère β comme constante, et les composantes de U sont déterminées comme celles qui satisfont $\partial V(\beta, U) / \partial u_{nc} = 0$ et $\partial V(\beta, U) / \partial \lambda_n = 0$. Alors, chaque u_{nc} est donné par

$$u_{nc} = \frac{1}{\sum_{k=1}^C \left[\frac{d^2(x_n, y_n, \beta_c)}{d^2(x_n, y_n, \beta_k)} \right]^{\frac{1}{m-1}}}. \quad (4.39)$$

Ensuite, pour déterminer les paramètres β_c , une nouvelle itération est réalisée en considérant les u_{nc} comme constants et en calculant la solution de $\partial J(\beta, U) / \partial \beta_c = \mathbf{0}$. A partir de (4.34) et (4.37), nous pouvons montrer que ρ_c et α_c sont donnés par les équations (4.22) et (4.23). Pour arriver à ce résultat, nous avons considéré comme constante la fonction de pénalité g . Cette fonction est donnée par

$$g^2(x_n, y_n, \beta_c) = (x_n - \tilde{x}_c)^2 + (y_n - \tilde{y}_c)^2, \quad (4.40)$$

où $\tilde{x}_c$ et $\tilde{y}_c$ sont les coordonnées du centre de gravité pondéré du nuage de points associé au c -ième prototype, et calculées dans l'itération précédente de l'algorithme par (4.24).

Algorithme 4: C-moyennes floues prototypées.

Entrée: L'ensemble de points $P = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$, la constante m , le nombre de prototypes C et leurs valeurs initiales β .

Sortie: Les prototypes β trouvés et les mesures d'appartenance U .

```

1:    $i \leftarrow 1$  (Initialiser le nombre d'itérations)
2:   for  $c = 1$  to  $C$  (Mise à jour des mesures d'appartenance)
3:     for  $n = 1$  to  $N$ 
4:        $u_{nc} = \frac{1}{\sum_{k=1}^C \left[ \frac{d^2(x_n, y_n, \beta_c)}{d^2(x_n, y_n, \beta_k)} \right]^{\frac{1}{m-1}}}$ 
5:     for  $c = 1$  to  $C$  (Mise à jour des prototypes)
6:        $\tilde{x}_c = \frac{\sum_{n=1}^N u_{cn}^m x_n}{\sum_{n=1}^N u_{cn}^m}$     $\tilde{y}_c = \frac{\sum_{n=1}^N u_{cn}^m y_n}{\sum_{n=1}^N u_{cn}^m}$ 
7:        $\tilde{S}_{xx_c} = \sum_{n=1}^N u_{cn}^m (x_n - \tilde{x}_c)^2$     $\tilde{S}_{yy_c} = \sum_{n=1}^N u_{cn}^m (y_n - \tilde{y}_c)^2$ 
8:        $\tilde{S}_{xy_c} = \sum_{n=1}^N u_{cn}^m (x_n - \tilde{x}_c) \cdot (y_n - \tilde{y}_c)$ 
9:       (Calcul des paramètres des prototypes)
10:       $\rho_c = \tilde{x}_c \cos(\alpha_i) + \tilde{y}_c \sin(\alpha_i)$ 
11:       $\alpha_c = \frac{1}{2} \arctan \left( \frac{-2\tilde{S}_{xy_c}}{\tilde{S}_{yy_c} - \tilde{S}_{xx_c}} \right)$ 
12:       $i \leftarrow i + 1$ 
13:      if  $i \leq i_{max}$ 
14:        goto 2

```

Nous résumons ce calcul des C-moyennes floues prototypées dans l'Algorithme 4. La condition d'arrêt utilisée par cette implémentation est donnée par le nombre maximum d'itérations i_{max} .

Les inconvénients de cet algorithme, présenté ici dans sa forme classique, sont :

- I. *La sensibilité à l'initialisation des prototypes β .* En général, un mauvais départ de l'algorithme peut dégénérer les prototypes. Cette question est abordée par Peña *et al.* dans un travail d'analyse comparative de quatre méthodologies d'initialisation [167] ;
- II. *La non robustesse à des points aberrants.* Ceci peut arriver à cause du fait que le partition est soumise aux contraintes (4.35)-(4.36). La version possibiliste des C-moyennes [16] semble être plus robuste car elle n'impose pas de fortes contraintes aux partitions. Des techniques de statistiques robustes ont été aussi employées pour résoudre ce problème [112][65] ;
- III. *La connaissance a priori de C , le nombre de prototypes.* Bien entendu, dans un certain nombre d'applications cette information n'est pas connue au préalable. Cependant, il existe des approches adaptées à ce problème [65].

4.3.4.5 *Split-and-merge fuzzy (SMF)*

Dans [32], nous avons proposé l'algorithme *Split-and-Merge Fuzzy (SMF)*. Cet algorithme applique l'approche des C-moyennes floues prototypées dans un processus de *division / fusion*.

L'avantage principale du *SMF* est que nous n'avons plus besoin de connaître *a priori* le nombre de prototypes.

Le processus de division de cet algorithme est inspiré de l'algorithme *IEPF* (section 4.3.4.3), où à chaque itération un ensemble de points $P = \{\mathbf{p}_{n_i}, \mathbf{p}_{n_i+1}, \dots, \mathbf{p}_{n_e}\}$ est divisé en deux sous-ensembles P' et P'' si un critère n'est pas satisfait. Dans le *SMF*, la division est effectuée par l'algorithme des C-moyennes floues prototypées avec $C = 2$. Les prototypes sont initialisés de la manière suivante : le prototype β_1 est donné par la droite qui passe par les deux premiers points de P , et β_2 correspond à la droite qui passe par les deux derniers points de P . Ainsi, P est divisé en P' et P'' , qui sont déterminés à partir des mesures d'appartenance U . Le critère utilisé est le suivant : $\mathbf{p}_n$ appartient à P' si $u_{1n} \geq u_{2n}$, ou à P'' si $u_{1n} < u_{2n}$. Nous pouvons bien imaginer que les ensembles P' et P'' sont des fragments de P qui ne respectent pas l'ordonnement des points, comme c'est le cas du *IEPF*. Cette même procédure de division est appliquée à P' (resp. P'') si la dispersion du nuage de points supports caractérisé par l'écart-type

$$\sigma = \sqrt{\frac{\sum_{n|\mathbf{p}_n \in P'(\text{resp. } P'')} (\rho_c - x_n \cos(\alpha_c) - y_n \sin(\alpha_c))^2}{\sum_{n|\mathbf{p}_n \in P'(\text{resp. } P'')}}}, \quad (4.41)$$

ne satisfait pas au critère $\sigma \leq \sigma_{\max}$. Il faut souligner que la trace des correspondances est toujours conservée dans Q .

En général, à l'issue de la phase de division, nous obtenons un nombre important de segments. Le principal avantage de cette fragmentation est que les plus petits segments de droite sont détectés, ce qui permet d'extraire des segments même dans des conditions difficiles d'observation comme par exemple dans des images d'environnements encombrés.

La phase de regroupement a comme objectif de fusionner les nuages de points voisins, en respectant toujours les critères de dispersion et de continuité. Ainsi, pour chaque prototype β_c , deux autres prototypes β_a et β_b sont choisis comme des candidats à la fusion avec β_c . Ces candidats sont en effet les prototypes les plus proches de β_c en considérant les centres de gravité des points supports. Nous définissons la fusion entre deux prototypes β' et β'' par $\beta = \beta' \oplus \beta''$, ce qui correspond simplement à calculer les paramètres de β à partir de l'ensemble de points supports des prototypes β' et β'' . En conséquence, le prototype fusionné β_f est celui parmi $\beta_c \oplus \beta_a$ et $\beta_c \oplus \beta_b$ tel que :

- I. Aucun point de cassure n'est détecté, ce qui requiert une nouvelle recherche pour des points de cassure faibles ;
- II. La dispersion σ est minimale et elle satisfait $\sigma \leq \sigma_{\max}$.

La phase de regroupement se termine lorsqu'il n'y a plus de fusion possible.

4.4 Simulations

Nous avons réalisé des simulations pour évaluer la qualité de l'extraction de droites dans des images télémétriques en utilisant l'approche de segmentation de la section 4.3. Notre objectif est

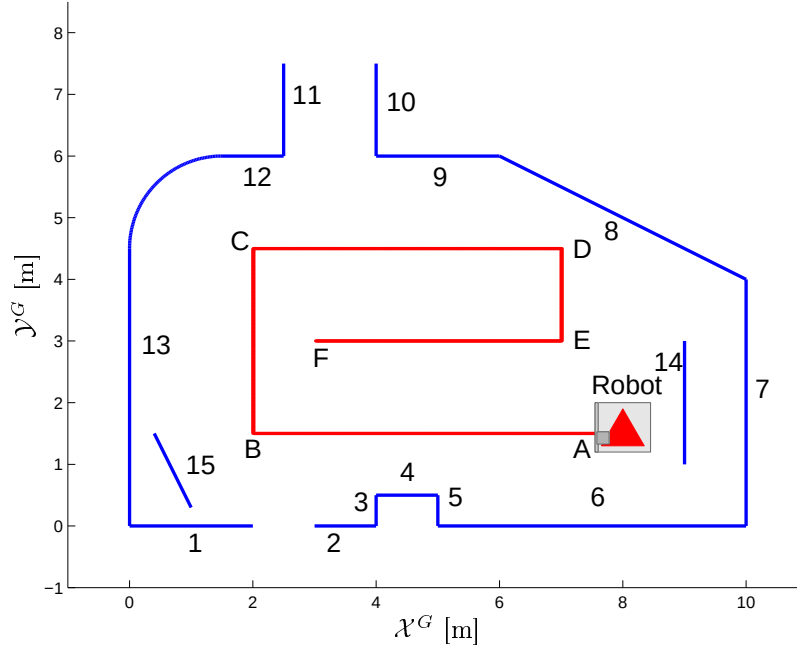


FIG. 4.8 – Evaluation des noyaux d'extraction de droites : environnement de simulation.

de valider l'utilisation des différents noyaux avec le système proposé, mais aussi d'évaluer leur robustesse vis-à-vis du bruit de mesure du capteur. L'évaluation est basée sur la simulation d'un environnement structuré, tel que celui présenté sur la Figure 4.8. Un robot mobile équipé d'un télémètre laser réalise un trajet qui est aussi illustré sur cette figure. Nous avons ainsi effectué deux simulations : une dans laquelle nous supposons un télémètre avec un bruit de mesure d'écart-type $\sigma_r = 0,03 \text{ m}$, et une autre avec $\sigma_r = 0,05 \text{ m}$. Les points de cassure ont été détectés en utilisant le détecteur adaptatif. Les mesures aberrantes dues à l'ouverture du faisceau laser n'ont pas été simulées.

Pour ces simulations, le robot a réalisé le trajet $ABCDEF$, et le capteur a balayé les surfaces représentées par les segments numérotés de 1 à 15 (voir Fig. 4.8). Ces surfaces sont modélisées par les paramètres polaires $\mathbf{e}_l = (\rho_l^e, \alpha_l^e)$, $l = 1, \dots, 15$, des droites infinies associées à ces segments. Nous pouvons voir aussi l'existence d'une surface curviligne du côté gauche en haut de la figure. Durant ce trajet, 1000 images télémétriques ont été simulées dans des positions intermédiaires sans que le mouvement du capteur soit pris en compte. Pour chaque image, l'extraction de droites a été effectuée par l'Algorithme 3 en utilisant chacun des noyaux présentés dans ce manuscrit : LT , $IEPF$ et SMF . Les paramètres employés ont été $T_{\max} = 10 \text{ cm}$ pour LT et $IEPF$, et $\sigma_{\max} = \sigma_r$ pour SMF .

Nous définissons quatre mesures de qualité pour mieux évaluer les résultats obtenus. Ces mesures sont des statistiques associées aux droites extraites par le balayage de chacune des surfaces de l'environnement. Ainsi, soit $\Omega_l \in \Omega$ une sous liste composée de N_l droites extraites ayant des points supports générés par le balayage de $\mathbf{e}_l$. Un segment extrait ne peut pas appartenir à plus d'une liste Ω_l . Cependant, un segment extrait $\mathbf{l}_k$ peut avoir des points supports qui ont été générés

TAB. 4.1 – Valeurs de $\bar{N}_l$, $\bar{\eta}_l$, $\bar{\varepsilon}_\rho$, et $\bar{\varepsilon}_\alpha$ pour la simulation avec $\sigma_r = 0,03 \text{ m}$.

e_l index	$\bar{N}_l$			$\bar{\eta}_l$			$\bar{\varepsilon}_\rho$ en cm			$\bar{\varepsilon}_\alpha$ en degrés		
	LT	IEPF	SMF	LT	IEPF	SMF	LT	IEPF	SMF	LT	IEPF	SMF
1	1,00	1,02	1,00	0,57	0,57	0,58	1,45	1,67	1,49	0,61	0,68	0,62
2	1,01	1,06	1,02	0,61	0,55	0,58	8,13	8,33	8,26	1,36	1,41	1,38
3	1,00	1,01	1,00	0,86	0,98	0,97	10,14	2,28	3,74	8,59	3,13	4,10
4	1,02	1,06	1,04	0,99	0,98	0,98	9,88	8,03	8,14	1,27	1,03	1,04
5	1,00	1,01	1,01	0,91	1,00	0,97	4,85	1,95	1,34	9,02	3,46	3,49
6	1,07	1,65	1,05	0,49	0,41	0,46	2,64	6,51	2,56	0,24	0,60	0,23
7	1,06	1,15	1,07	1,00	0,99	1,00	6,71	6,68	6,29	1,26	1,28	1,19
8	1,04	1,76	1,16	0,98	0,94	0,98	4,54	5,92	3,63	0,66	0,93	0,57
9	1,04	1,22	1,09	0,99	0,95	0,98	4,49	5,31	4,60	0,52	0,61	0,53
10	1,00	1,00	1,00	1,00	0,99	1,00	4,29	5,55	5,77	0,36	0,47	0,49
11	1,00	1,00	1,00	0,97	0,99	1,00	13,41	7,37	8,04	1,11	0,62	0,68
12	1,03	1,13	1,19	0,98	0,98	0,93	14,06	7,18	7,94	3,73	2,00	2,24
13	1,30	1,98	1,38	0,24	0,37	0,33	1,52	6,94	2,60	0,35	1,31	0,58
14	1,02	1,04	1,02	1,00	1,00	1,00	2,19	2,51	2,33	0,56	0,66	0,60
15	1,02	1,08	1,06	1,00	1,00	1,00	0,97	1,31	1,23	0,87	1,16	1,11

par le balayage de différents segments de l'environnement simulé. Pour résoudre ce conflit, $\mathbf{l}_k$ est associé à la sous liste correspondant au segment de droite $\mathbf{e}_l$ qui ressemble le plus à $\mathbf{l}_k$ dans le sens des paramètres polaires ρ et α . Nous pouvons aussi avoir une droite de l'environnement $\mathbf{e}_l$ qui peut être associée à plus d'un segment extrait. Dans ce cas, il est aussi intéressant de pouvoir mesurer la validité de ces segments. Ainsi, un segment $\mathbf{l}_k$ extrait à partir des points de balayage d'un segment d'environnement $\mathbf{e}_l$ est reconnu comme valide si $\varepsilon_\rho \leq 0,5 \text{ m}$ et $\varepsilon_\alpha \leq 20^\circ$, où $\varepsilon_\rho = \|\rho_l^e - \rho_k\|$ et $\varepsilon_\alpha = \|\alpha_l^e - \alpha_k\|$ sont les écarts absolus des paramètres. Nous définissons la mesure de qualité $\eta_l = M_l/N_l$ avec $0 \leq \eta_l \leq 1$, où M_l est le nombre de segments valides de la sous-liste Ω_l . η_l est proche de 1 lorsque les segments extraits ressemblent à $\mathbf{e}_l$. Les mesures de qualité sont calculées pour chaque droite de l'environnement. En résumé, ces mesures sont donc :

- N_l , qui est le nombre de droites extraites du segment $\mathbf{e}_l$. Idéalement, nous attendons que seulement un segment soit généré à partir du balayage du segment $\mathbf{e}_l$ de l'environnement, *i.e.*, $N_l = 1$. Si $N_l > 1$, c'est parce que l'algorithme a fragmenté $\mathbf{e}_l$ en plusieurs morceaux ;
- η_l , qui doit être le plus proche possible de 1 ;
- ε_ρ et ε_α , qui tendent vers zéro lorsque le segment extrait a une forte ressemblance avec le segment $\mathbf{e}_l$.

Les résultats obtenus sont présentés comme les moyennes des mesures de qualité calculées pour les 1000 images simulées. Le Tableau 4.1 présente les moyennes $\bar{N}_l$, $\bar{\eta}_l$, $\bar{\varepsilon}_\rho$ et $\bar{\varepsilon}_\alpha$ respectives de N_l , η_l , ε_ρ et ε_α , calculées à partir des segments extraits, pour la simulation d'un capteur avec $\sigma_r = 0,03 \text{ m}$. Les valeurs en gras indiquent les meilleures performances pour chaque mesure de qualité parmi les noyaux *LT*, *IEPF* et *SMF*. Nous concluons avec cette simulation que les trois noyaux ont présenté des résultats similaires pour $\bar{N}_l$ et $\bar{\eta}_l$. Cela signifie qu'avec un capteur laser plus précis, les segments de l'environnement présenteraient presque le même degré de détectabilité indépendamment du noyau utilisé. Cependant, nous rappelons que les mesures aberrantes n'ont pas été simulées, et cela est un phénomène courant avec un capteur réel. Pour les écarts absolus

TAB. 4.2 – Valeurs de $\bar{N}_l$, $\bar{\eta}_l$, $\bar{\varepsilon}_\rho$, et $\bar{\varepsilon}_\alpha$ pour la simulation avec $\sigma_r = 0,05 m$.

e_l index	$\bar{N}_l$			$\bar{\eta}_l$			$\bar{\varepsilon}_\rho$ en cm			$\bar{\varepsilon}_\alpha$ en degrés		
	LT	IEPF	SMF	LT	IEPF	SMF	LT	IEPF	SMF	LT	IEPF	SMF
1	1,11	1,14	1,01	0,61	0,62	0,62	4,82	6,32	2,62	1,96	2,62	1,08
2	1,39	1,39	1,08	0,43	0,39	0,59	19,78	20,04	15,93	3,39	3,43	2,70
3	1,01	1,02	1,03	0,93	0,97	0,87	7,10	3,25	9,02	7,15	5,30	8,35
4	1,36	1,38	1,11	0,76	0,66	0,92	17,44	17,46	13,28	2,24	2,25	1,71
5	1,02	1,02	1,01	0,88	0,94	0,92	3,94	4,04	4,13	7,30	5,02	6,80
6	2,74	3,18	1,17	0,32	0,26	0,44	12,86	18,71	5,61	1,15	1,62	0,47
7	1,23	1,25	1,09	0,90	0,84	0,98	15,81	19,07	9,90	3,09	3,90	1,92
8	2,59	3,30	1,36	0,76	0,66	0,97	12,23	17,78	5,91	1,91	2,88	0,89
9	1,77	2,04	1,16	0,74	0,60	0,94	14,22	17,95	8,25	1,68	2,11	0,96
10	1,01	1,06	1,01	0,98	0,95	0,95	8,91	10,56	13,37	0,76	0,90	1,14
11	1,00	1,01	1,02	0,89	0,94	0,94	14,90	11,86	13,55	1,26	1,01	1,14
12	1,33	1,36	1,20	0,88	0,87	0,91	14,93	12,66	12,01	4,46	4,41	3,28
13	2,53	2,84	1,33	0,35	0,35	0,30	14,52	20,74	3,73	3,05	4,12	0,79
14	1,26	1,36	1,02	0,97	0,95	0,99	8,33	10,17	3,33	2,46	3,29	0,87
15	1,43	1,41	1,09	0,91	0,89	0,99	4,27	5,08	1,87	4,45	5,09	1,83

moyens $\bar{\varepsilon}_\rho$ et $\bar{\varepsilon}_\alpha$, nous concluons qu'aucun algorithme ne présente les meilleurs résultats pour tous les cas. Cependant, dans tous les cas où le noyau *SMF* ne donne pas l'écart absolu moyen minimum, son résultat est très proche du meilleur résultat. Cela ne se vérifie pas pour les autres algorithmes. Ainsi, nous concluons que, pour cette simulation, le noyau *SMF* a présenté plus de stabilité que les noyaux *LT* et *IEPF*.

Le Tableau 4.2 présente les moyennes des mesures de performance pour la simulation d'un capteur avec $\sigma_r = 0,05 m$. En analysant $\bar{N}_l$, nous voyons que le noyau *SMF* a généré moins de segments par surface de l'environnement que *LT* et *IEPF*. Mais cela peut être la conséquence d'un mauvais choix du seuil $T_{\max}$ utilisé par les noyaux *LT* et *IEPF*. Pour $\bar{\eta}_l$, le noyau *SMF* a présenté aussi les meilleurs résultats, sauf pour les segments 1, 3, 5, 10 et 13 de l'environnement. Pour ces cas, $\bar{\eta}_l$ a été très proche du meilleur résultat, qu'il soit obtenu par *LT* ou par *IEPF*. En ce qui concerne les écarts moyens absolus $\bar{\varepsilon}_\rho$ and $\bar{\varepsilon}_\alpha$, pour la majeure partie des cas le noyau *SMF* a été celui qui a présenté une dégénération moins importante par rapport aux résultats du Tableau 4.1. Cela est un indice de robustesse caractéristique des approches *fuzzy clustering*, dû à une mise à jour itérative des mesures d'appartenance u_{nc} .

4.5 Résultats expérimentaux

Dans cette section nous présentons des résultats expérimentaux obtenus par l'application des approches présentées dans ce chapitre à des images réelles.

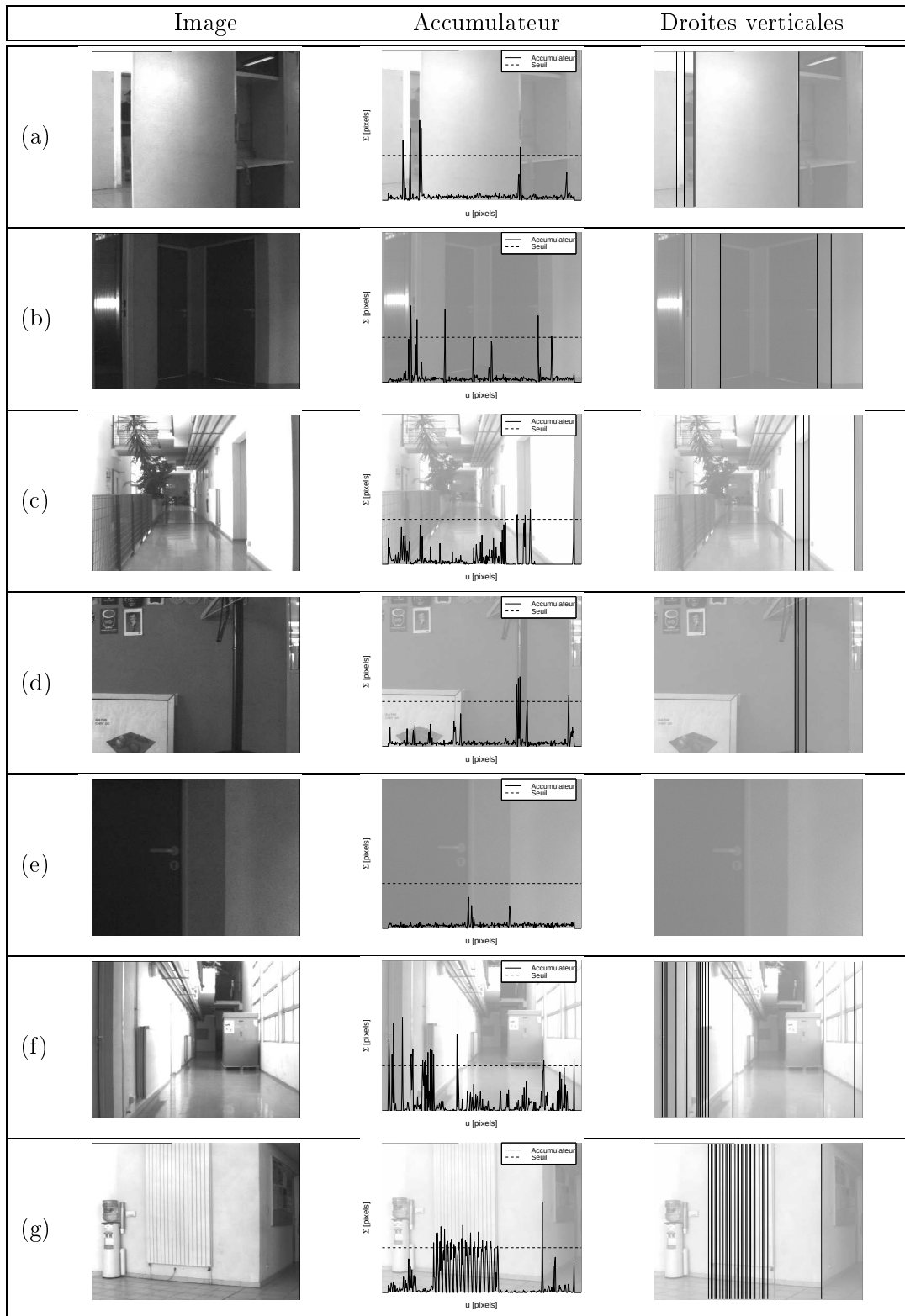


FIG. 4.9 – Exemples de la détection de droites verticales : (a)-(c) bons résultats et (d)-(g) mise en défaut de la méthode.

4.5.1 Segmentation d'images vidéo

La méthode d'extraction de droites verticales dans des images vidéo a été mise à l'épreuve dans des conditions réelles d'expérimentation. Nous présentons sur la Figure 4.9 les résultats obtenus sous la forme d'images vidéo, d'accumulateurs et de droites verticales extraites. Ces deux dernières sont affichées en sur-impression sur l'image réelle. Nous présentons sur cette figure les bons résultats en (a)-(c), mais également des cas illustrant les échecs de la méthode en (d)-(g). Dans ces résultats, le seuil de détection appliqué à l'accumulateur est fixé à 30% du nombre de pixels de la colonne.

Tout d'abord, commentons les bons résultats. Sur (a) nous avons un cas simple où le robot se trouve en face d'un pilier. Nous vérifions que les droites verticales délimitant le pilier ont été bien détectées. L'algorithme a bien fonctionné dans le cas (b), où le niveau d'intensité dans l'image était assez bas. La plupart des contours ont été détectés avec succès. Nous vérifions aussi dans l'accumulateur que d'autres contours encore visibles n'ont pas été détectés parce que le seuil utilisé n'était pas assez bas pour la scène en question. Sur (c) nous avons un cas où, à cause de la présence de différentes structures peu nettes (comme une grille sur le côté gauche de l'image), un niveau considérable de *bruit de fond* apparaît dans l'accumulateur. Cependant, le niveau de ce bruit n'est pas assez élevé pour que de faux contours soient détectés. Nous remarquons que la méthode a donné de bons résultats tant pour le cas (b), que pour le cas (c), qui sont des cas opposés. Nous rappelons que pour cette méthode le seuil de l'accumulateur est fixe, et qu'il a été déterminé de façon empirique, au préalable, en utilisant d'autres images réelles. Nous avons préféré le mettre suffisamment haut pour éviter des fausses détections.

Nous passons maintenant à des cas où la détection est mise en cause. Sur (d) nous avons un reflet parasite sur une surface cylindrique verticale. Ce phénomène naturel se traduit par des contours factices sur l'image. Ce genre d'indice ne représente pas une discontinuité géométrique dans l'environnement réel.

Le cas de l'image (e) montre aussi qu'un niveau assez bas d'intensité dans l'image rend plus difficile la détection de contours verticaux. Enfin, les cas (f) et (g) présentent des images où un nombre excessif de contours réels a été détecté. Ceci doit être pris en compte dans la méthodologie de construction de cartes locales.

Ces résultats montrent que le choix d'un seuil fixe optimal est difficile. Il doit être haut pour éviter les fausses détections, tout en assurant la détection des contours verticaux caractéristiques de l'environnement. Le même problème est rencontré avec la méthode de Arras *et al.* [8].

4.5.2 Segmentation d'images télémétriques

4.5.2.1 Détection de points de cassure

Dans la section 4.3.2, nous avons présenté des algorithmes pour la détection de points de cassure. Ces indices sont utilisées dans l'algorithme d'extraction de segments de droites pour la détermination de régions continues. Dans le chapitre 5, nous abordons la construction de cartes locales en utilisant des indices extraits par segmentation des données multicauteur. Nous verrons

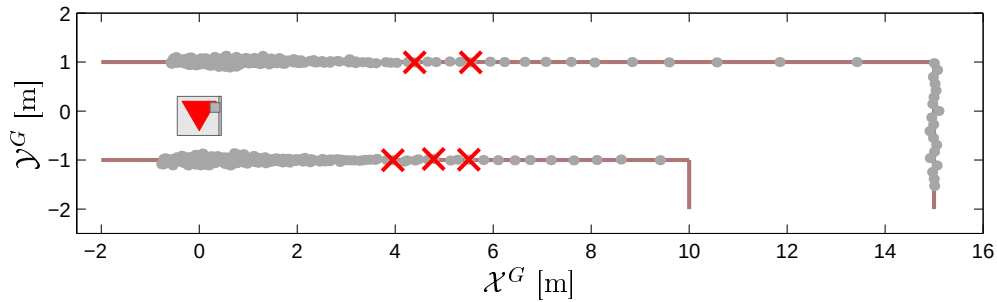


FIG. 4.10 – Contre-exemple où les points de cassure forts ne représentent pas les extrémités des surfaces balayées.

que les extrémités de segments déterminées à partir de points de cassure forts de l'image télémétrique correspondent probablement à des limites physiques des surfaces balayées par le capteur (cf. section 4.3.2.3). Cependant, la règle simple sur laquelle repose l'affectation des points de cassure forts présente des contre-exemples, tel que celui de la Figure 4.10. Elle montre la simulation d'une image télémétrique prise dans un couloir par le robot positionné à l'origine du repère global. Les points de cassure forts sont indiqués par le symbole ($\times$). Il est très clair que ces points ne correspondent pas aux extrémités des surfaces balayées. Cela s'explique par le fait que les détecteurs sont fondés uniquement sur l'angle d'incidence minimale. Ainsi, nous démontrons que ces indices ne peuvent pas à eux seuls définir des limites physiques des surfaces. Pour définir ces limites, il est nécessaire d'utiliser d'autres mécanismes de détection pour lesquels la présence de points de cassure forts est une condition nécessaire, mais pas suffisante. Nous reviendrons sur ce sujet dans le chapitre 5.

Nous nous permettons d'avancer ces informations pour justifier la méthodologie d'évaluation employée dans cette section. Nous proposons donc d'évaluer les détecteurs de points de cassure par vérification visuelle de certains des points de cassure forts extraits dans une séquence d'images. Nous allons nous restreindre seulement aux points de cassure qui ont une plus forte chance d'indiquer des limites physiques des obstacles. Mais, comment pouvons-nous faire pour sélectionner de tels points parmi tous ceux qui sont détectés? Sachant que la résolution spatiale du télémètre diminue en fonction de la distance mesurée, nous supposons pour l'instant que les extrémités des surfaces sont détectées avec une plus grande fiabilité si elles sont suffisamment proches du capteur. Ce simple raisonnement permet une réduction considérable du nombre de points de cassure forts sélectionnés pour cette évaluation.

Notre évaluation a été menée avec des données acquises par le robot Omni lors de l'exploration d'un couloir en circuit fermé dans le laboratoire (cf. section G.7). Les résultats obtenus sont présentés sur la Figure 4.11. Toutes les images télémétriques acquises durant l'expérience sont affichées dans un repère global. La position du véhicule est estimée par gyrométrie. Dans cette expérience, nous nous intéressons seulement aux points de cassure forts, indiqués par le symbole ($\times$), et qui ont été sélectionnés en fonction de leur distance par rapport au robot. Tous ces points étaient à moins de deux mètres du capteur lors de leur acquisition. Nous pouvons vérifier que les résultats

des deux approches ne diffèrent pas tellement, et que la majeure partie des points sélectionnés correspond bien à des extrémités de surfaces. Cependant, l'approche utilisant le filtre de Kalman a tendance à générer un nombre plus important de points de cassure (Figure 4.11(b)). Parmi ces points, nous avons deux faux résultats indiqués par des flèches. Pour que cette approche détecte un nombre moins important de points, il suffit d'augmenter les covariances $\mathbf{P}_0$, $\mathbf{Q}_n$ et $\mathbf{R}_n$. Cependant, cette solution peut se traduire par la non détection de points de cassure importants. Ainsi, il n'est pas évident de savoir de comment augmenter ces matrices.

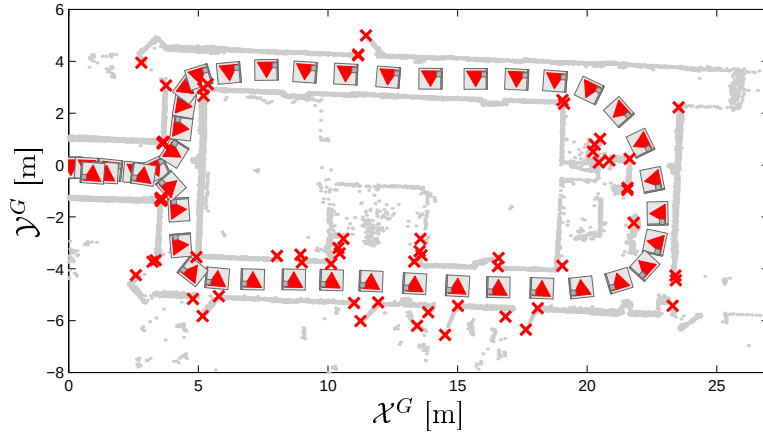
Les expériences réalisées avec d'autres données nous permettent de conclure que le détecteur par seuillage adaptatif présente des résultats plus stables et plus précis que celui qui utilise le filtre de Kalman. A cela nous rajoutons la difficulté de régler les matrices de covariance du filtre. Même si nous avons pris soin de déterminer des valeurs consistantes pour $\mathbf{P}_0$ et $\mathbf{R}_n$, nous ne pouvons pas garantir que $\mathbf{Q}_n$ le soit. On pourrait cependant proposer d'utiliser la forme générique $\mathbf{Q}_n = \gamma \cdot \mathbf{I}_2$, et déterminer le scalaire γ par essai-erreur avec des données réelles. Cela est une autre approche du problème. Mais nous préférons une méthode facile à mettre en œuvre, intuitive et efficace, comme c'est le cas du détecteur adaptatif.

4.5.2.2 Extractions de segments de droite

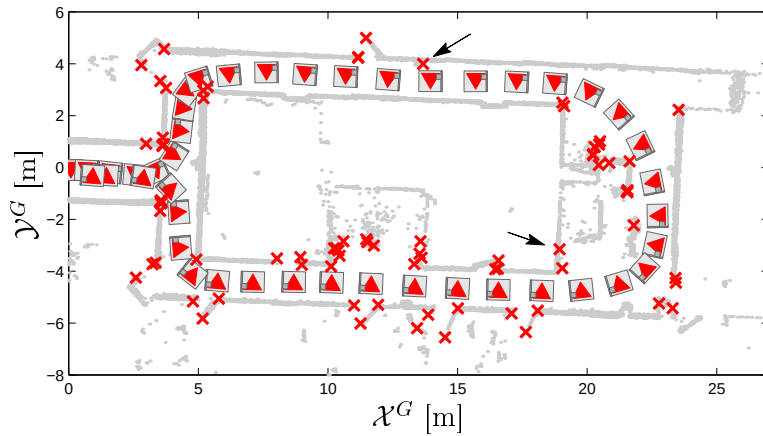
Pour évaluer l'approche d'extraction de segments de droite, présentée dans la section 4.3.3, nous proposons une comparaison expérimentale des noyaux *LT*, *IEPF* et *SMF*. Les images utilisées pour cette comparaison ont été choisies parmi celles de l'expérience de la section précédente (cf. Figure 4.11). En effet, cette évaluation a consisté en une analyse de cas (cf. Figure 4.12), pour lesquels les noyaux ont présenté des mauvais résultats. Ces résultats sont indiqués par des flèches. Les points de cassure forts et faibles sont aussi indiqués respectivement par les symboles (+) et (×).

Pour cette évaluation, nous avons utilisé $l_{\min} = 0,5 \text{ m}$ et $N_{\min} = 10$ comme critères de sélection de segments. Pour les noyaux, les paramètres employés ont été $T_{\max} = 10 \text{ cm}$ pour *LT* et *IEPF*, et $\sigma_{\max} = \sigma_r = 0,03 \text{ m}$ pour *SMF*, qui sont les valeurs utilisées dans la simulation de la section 4.4.

Nous commençons par l'analyse du cas de l'image (a), pour laquelle le noyau *IEPF* a donné un segment qui ne correspond pas vraiment à une surface de l'environnement. En effet, les points supports de ce segment correspondent au balayage d'un extincteur d'incendie qui se trouve installé sur un mur plan. Nous pouvons constater que les noyaux *LT* et *SMF* n'ont pas été affectés par la présence de cet objet. Par contre, le noyau *LT* a présenté des segments qui ne correspondent pas à des surfaces de l'environnement (cas (b)). Dans ce cas, ce sont les noyaux *IEPF* et *SMF* qui ont donné les meilleurs résultats. Nous avons remarqué que dans certains cas le noyau *SMF* a aussi fourni des segments superflus. Cependant, ces cas ont été bien moins nombreux qu'avec les noyaux *LT* et *IEPF*. Sur l'image (c) nous avons en réalité une porte fermée qui se traduit dans l'image par un léger relief dans la séquence de points de balayage (voir la flèche). Pour ce cas, les trois noyaux ont présenté des résultats différents. Le noyau *LT* n'a pas réussi à déterminer une des extrémités de la porte. *IEPF* a généré un segment qui ne correspond ni à la porte, ni aux surfaces qui l'entourent. Avec le noyau *SMF*, la porte est négligée et un seul segment est obtenu. Nous avons ici un cas extrême, où le relief causé dans l'image par la porte passe presque inaperçu. Ce phénomène est très



(a) Détecteur par seuillage adaptatif



(b) Détecteur filtre de Kalman

FIG. 4.11 – Evaluation expérimentale des détecteurs de points de cassure.

fréquent pour des milieux d'intérieur. L'image suivante de cette scène correspond au cas (d). Cette fois ci, la détection du segment correspondant à la porte est effectué avec succès par les noyaux *LT* et *SMF*. Cependant, les extrémités sont détectées avec une meilleure précision par le noyau *SMF*. Le noyau *IEPF* n'a pas réussi la détection. Le cas (e) illustre bien la capacité de chaque noyau à discerner l'intersection de surfaces. Pour l'image en question, le télémètre a balayé une porte ouverte. Si nous regardons les segments extraits, on voit clairement que seulement le noyau *LT* n'a pas réussi à différencier les surfaces.

4.5.2.3 Temps de calcul

Au delà de l'analyse des résultats de segmentation, nous avons aussi mesuré le temps de calcul nécessaire pour différentes configurations. Nous analysons le système complet de segmentation,

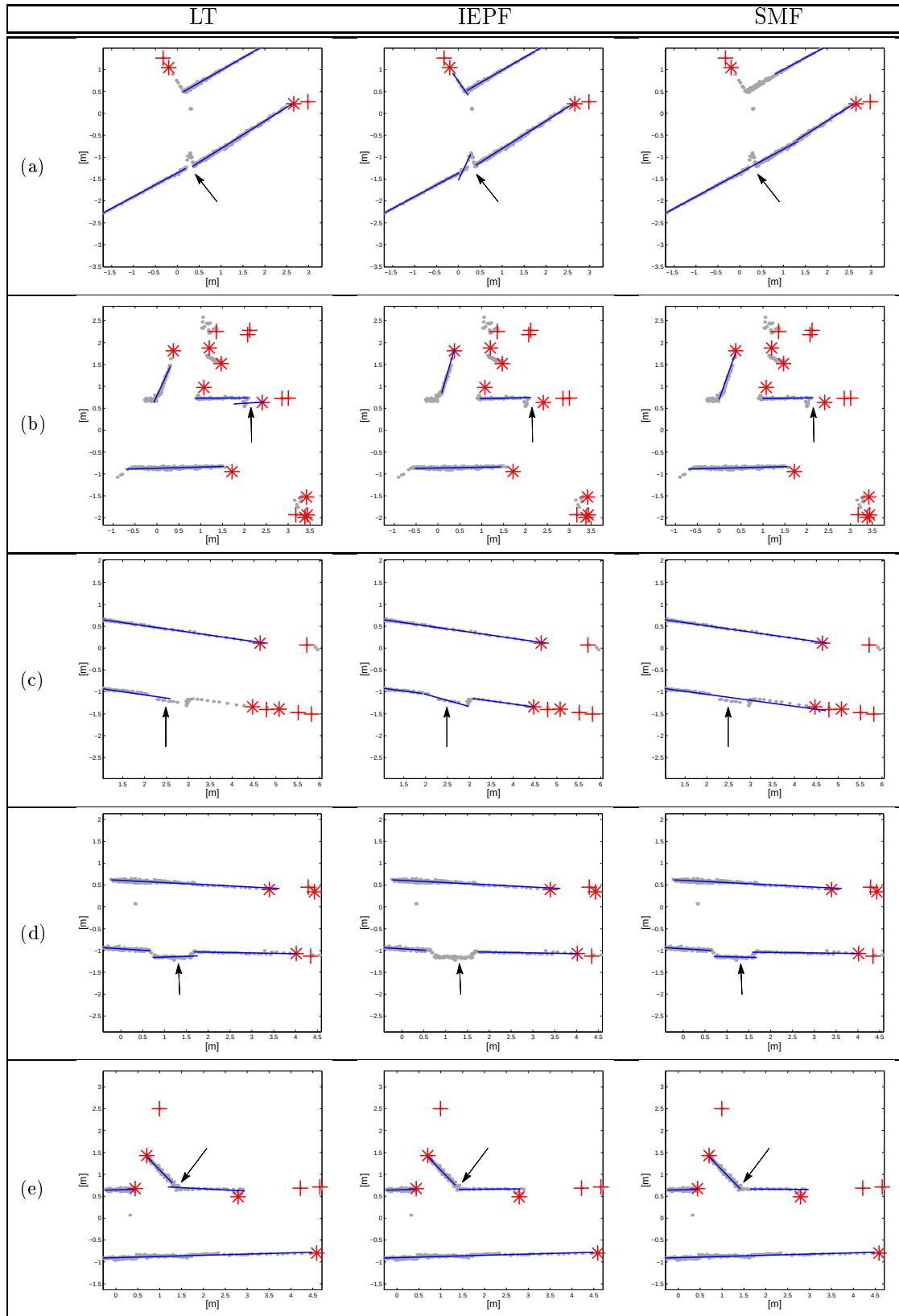


FIG. 4.12 – Evaluation expérimentale de la méthode d'extraction de segments dans les images télémétriques : comparaison des résultats obtenus avec les différents noyaux.

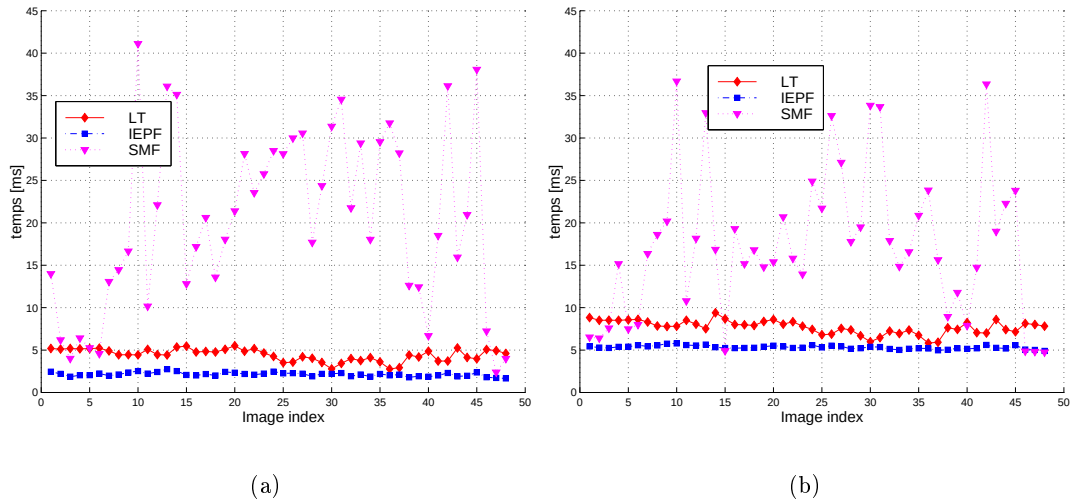


FIG. 4.13 – Temps de calcul des algorithmes de segmentation d’images télémétriques utilisant les différents noyaux pour l’extraction de segments. Ces résultats sont présentés en (a) pour l’approche adaptative de détection de points de cassure, et en (b) pour le détecteur basé sur le filtre de Kalman.

i.e., détection de points de cassure et extraction de segments de droite. Les mesures ont été faites sur la séquence d’images de l’expérience de la Figure 4.11. Pour un total de 48 images, nous présentons les temps d’exécution sur deux graphiques. La Figure 4.13(a) donne le temps total de segmentation de chaque image en utilisant l’approche par seuillage adaptatif pour la détection de points de cassure. Avec le filtre de Kalman, les temps sont donnés sur la Figure 4.13(b). Le temps d’exécution maximal est de 41 *ms*, ce qui est largement suffisant pour l’application en temps réel de ces approches dans notre système. Cependant, les temps de calcul sont plus importants avec l’approche filtre de Kalman. Toutefois, cette augmentation n’est pas d’une grande importance. Nous remarquons la grande variance du temps de calcul en utilisant le noyau *SMF*. Une telle variance s’explique par la nature itérative du partitionnement engendré par les C-moyennes floues prototypées. Dans notre implémentation, le nombre maximum d’itérations de l’algorithme des C-moyennes floues (Algorithme 4) a été fixé à $i_{\max} = 15$. En plus, nous avons modifié le critère d’arrêt de cet algorithme de sorte qu’il puisse terminer son exécution avant la $i_{\max}$ -ième itération pour les cas où le changement de la fonction de coût J (équation (4.34)) est inférieur à un seuil, ce qui permet d’arrêter le partitionnement lorsqu’il n’est plus utile.

4.6 Conclusions

Ce chapitre a présenté les procédures de segmentation d’images du système multicapteur décrit dans le chapitre 3. La segmentation d’images vidéo a comme objectif la détection de contours verticaux qui correspondent à des structures géométriques de l’environnement. Les droites verticales sont alors obtenues par filtrage puis seuillage d’un accumulateur. L’extraction de segments de droite dans les images télémétriques est effectué par une approche *division / fusion*, où des procédures

classiques de segmentation sont appliquées pour des régions continues de l'image. Les régions sont délimitées par des points de cassure, pour lesquels nous avons mis au point deux techniques de détection. En ce qui concerne l'extraction de segments, nous avons présenté une classe d'algorithmes noyaux sous une forme unifiée. Parmi ces noyaux, nous avons mis au point l'algorithme *SMF*, dérivé des C-moyennes prototypées. Des comparaisons en simulation ont indiqué que le noyau *SMF* est plus robuste vis-à-vis du bruit de mesure du capteur télémétrique que d'autres approches classiques très utilisées en robotique mobile. Une évaluation expérimentale nous a permis de vérifier les défauts des méthodes exposées lors de leur application dans des cas réels.

Même si le plus grand soin est pris pour l'implémentation des méthodes développées dans ce chapitre, nous ne pouvons pas faire aveuglement confiance à leurs résultats. Nous pensons qu'il est possible de pallier leur manque de robustesse en améliorant les traitements ultérieurs des données, lors de la phase de construction de cartes locales.

Cartographie locale

5.1 Introduction

Dans le chapitre 4 nous avons présenté des algorithmes destinés à la segmentation d'images télémétriques $\mathcal{L}$ et d'images vidéo $\mathcal{I}$, fournies par le système multicapteur. Lorsque ces procédures sont appliquées à une paire d'images $(\mathcal{L}, \mathcal{I})$, nous obtenons deux listes de primitives géométriques Ω^S et Ω^U . Ces listes de primitives sont exploitées par le module de *cartographie locale* pour la construction d'une représentation structurée de l'environnement local sous forme d'une carte locale $\mathcal{M}^R$ (voir annexe D pour un descriptif). La carte locale $\mathcal{M}^R$ est utilisée tant pour la localisation, que pour la mise à jour de la carte globale de l'environnement. Cependant, il faut éviter que des erreurs détectées lors de la cartographie locale se propagent dans la localisation et dans la cartographie globale. Par conséquent, un grand soin doit être apporté à la sélection des primitives géométriques de Ω^S et Ω^U pour la détermination des structures cartographiques. Cela se traduit par la prise en considération uniquement des indices les plus fiables, étant donné leurs conditions d'observation, et/ou leur incertitude. En ce qui concerne la détermination des paramètres d'incertitude, nous avons choisi de les estimer en considérant les cas les plus extrêmes, dans le but de pouvoir surévaluer l'incertitude et d'éviter ainsi des problèmes d'inconsistance stochastique.

La construction explicite d'une carte locale de l'environnement n'est pas une étape toujours présente dans des systèmes de localisation multicapteur. Par exemple, Arras *et al.* [8] utilisent séparément pour la localisation des indices géométriques extraits d'images fournies par un télémètre laser et par une caméra vidéo. Une approche semblable a été utilisée par Neira *et al.* [170] avec un capteur télémétrique 3D qui fournit des images de distance et d'intensité. Dans ces deux cas,

les indices géométriques extraits sont mis en correspondance directement avec une carte globale construite au préalable pour le recalage d'un robot mobile. L'utilisation découplée de données multicapteur simplifie le processus d'acquisition où une pseudo-synchronisation n'est pas nécessaire. Cependant, la fusion d'indices géométriques pour la cartographie locale permet d'exploiter la complémentarité des différents capteurs pour mieux initialiser les structures cartographiques. Ainsi, Lallement *et al.* [130] ont appliqué une technique de fusion d'indices géométriques fournis par une caméra vidéo et un télémètre laser à la détection de coins dans l'environnement local.

Nous exploitons le système multicapteur du robot Omni de la façon suivante. Une initialisation de la carte locale est d'abord effectuée en utilisant seulement les données télémétriques de la liste Ω^S . Une partie de ces structures cartographiques est passible d'être observée par la caméra vidéo sous la forme de droites verticales dans l'image. Ces nouvelles observations sont utilisées pour la mise à jour des paramètres et pour une réduction de leur incertitude. Nous appliquons des concepts classiques en fusion de données multisensorielles. Cependant, l'originalité de notre approche consiste à exploiter également des indices vidéo qui ne correspondent pas obligatoirement à des variations du relief de la scène représentées dans la carte locale. Ces indices ne sont pas systématiquement rejetés comme le font d'autres approches. En effet, nous cherchons à exploiter ces indices photométriques complémentaires pour enrichir la carte locale.

L'annexe D décrit plus précisément la représentation géométrique d'environnements choisie pour notre étude. Les structures cartographiques sont introduites dans la section D.1, et leur organisation dans la carte locale est présentée dans la section D.4. Dans ce chapitre, la section 5.2 décrit les procédures impliquées dans la construction d'une carte locale. La section 5.3 présente une validation expérimentale de notre approche, ainsi que l'analyse d'un contre-exemple.

5.2 Construction de la carte locale

A l'issue de la segmentation d'images, nous obtenons :

- Une liste $\Omega^U = \{U_k \mid k = 1, \dots, N_{\Omega^U}\}$ d'indices $U_k = \{u_k, \mathbf{\Lambda}_{u_k}\}$ extraits de l'image vidéo. Dans chaque indice U_k , u_k est la coordonnée suivant l'axe $\mathcal{U}$ de la colonne supportant la k -ième droite verticale, et $\mathbf{\Lambda}_{u_k}$ est sa variance estimée ;
- Une liste $\Omega^S = \{S_k \mid k = 1, \dots, N_{\Omega^S}\}$ de segments de droites $S_k = \{\mathbf{l}_k, \mathbf{\Lambda}_{\mathbf{l}_k}, \mathbf{p}_{a_k}, \kappa_{a_k}^+, \mathbf{p}_{b_k}, \kappa_{b_k}^+\}$ extraits de l'image télémétrique. Pour le k -ième indice, $\mathbf{l}_k = (\rho_k, \alpha_k)^T$ représente les paramètres de la droite et $\mathbf{\Lambda}_{\mathbf{l}_k}$ est la matrice de covariance associée à $\mathbf{l}_k$. $\mathbf{p}_{a_k}$ et $\mathbf{p}_{b_k}$ sont les coordonnées des extrémités du segment de droite, et $\kappa_{a_k}^+$ et $\kappa_{b_k}^+$ sont des étiquettes qui indiquent si ces extrémités ont été obtenues à partir de points de cassure forts ;

Nous remarquons que les repères $\mathcal{R}^S$ du système multicapteur et $\mathcal{R}^R$ du robot ne coïncident pas (voir Annexe A). La carte locale étant référencée en $\mathcal{R}^R$, les procédures de segmentation doivent fournir des indices dans ce repère. Or l'image télémétrique est obtenue dans le repère $\mathcal{R}^L$. Pour résoudre ce problème, nous proposons que la segmentation des images télémétriques soit effectuée de la façon suivante : les points de cassure sont détectés dans le repère télémétrique avec l'image $\mathcal{L}$ qui vient d'être acquise, et l'extraction de segments s'effectue sur l'image télémétrique

transformée dans le repère local. Ainsi, tous les indices télémétriques sont calculés dans le repère $\mathcal{R}^R$, et les étiquettes des points de cassure sont déterminées avec l'image dans le repère $\mathcal{R}^S$.

Nous abordons dans cette section la mise en œuvre d'une méthodologie de construction de cartes locales à partir des résultats de segmentation d'une paire d'images $(\mathcal{L}, \mathcal{I})$. Les structures qui composent ces cartes sont des demi-plans et des arêtes de type bord, coin et indice photométrique (cf. section D.1 en annexe). Nous construisons la carte locale en deux étapes : le traitement des indices télémétriques, et la fusion avec des indices visuels.

5.2.1 Traitement des indices télémétriques

Dans cette étape, les structures de type demi-plan, bord et coin sont d'abord déterminées en utilisant seulement les indices télémétriques. Pour mieux illustrer les procédures de cette section, nous prenons comme exemple les indices de la Figure 5.1(a). Sur cette figure, nous avons aussi l'image télémétrique sur laquelle la segmentation a été appliquée.

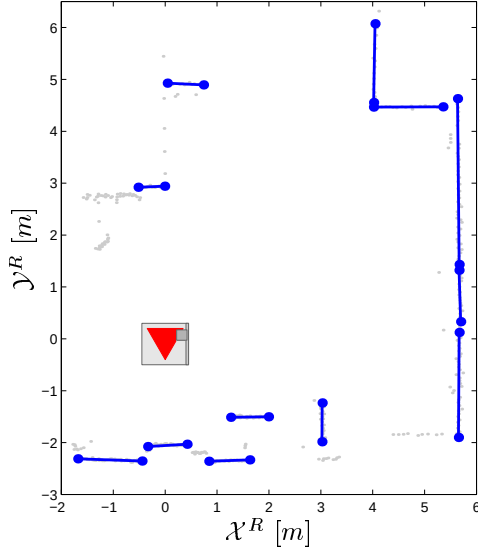
5.2.1.1 Détermination des demi-plans

Les segments de l'image télémétrique sont systématiquement utilisés pour modéliser les demi-plans. Les paramètres de la zone de visibilité sont initialisés en utilisant la règle (annexe D.2). Mais cette règle requiert la connaissance d'un point de contrôle $\mathbf{p}_v$, se trouvant dans la zone de visibilité du demi-plan observé. Etant donné que tous les segments de droite de la liste Ω^S ont été extraits de l'image télémétrique $\mathcal{L}$ acquise dans le repère capteur $\mathcal{R}^S$, l'origine de $\mathcal{R}^S$ se trouve dans la zone de visibilité des demi-plans correspondants. Par conséquent, $\mathbf{p}_v = (x_S^R, y_S^R)^T$ constitue le point de contrôle pour l'initialisation de tous les paramètres des zones de visibilité des demi-plans de la carte locale (cf. Figure 5.2). Appliquée à notre exemple de la Figure 5.1(a), la procédure détermine les demi-plans montrés sur la Figure 5.1(b). Comme nous pouvons le constater, tous les segments initiaux sont convertis en demi-plans sur la carte locale.

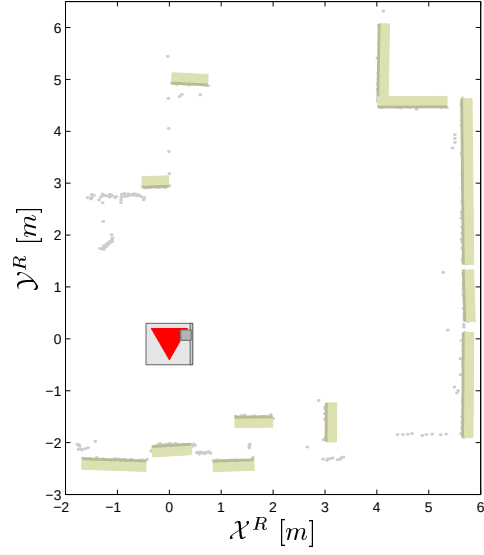
5.2.1.2 Détermination des arêtes de type bord

Nous avons vu dans la section 4.3.2.3 que les points de cassure forts de l'image télémétrique représentent les limites physiques des surfaces balayées par le télémètre. Or, l'intérêt principal des arêtes de type bord est de pouvoir représenter ces limites physiques dans la carte locale. Ainsi, les bords sont déterminés à partir des extrémités $\mathbf{p}_a$ et $\mathbf{p}_b$ des segments associés aux demi-plans, déjà inclus dans la carte $\mathcal{M}^R$, et qui ont été générés à partir de points de cassure forts. Pour cela, il suffit de vérifier les étiquettes κ_a^+ et κ_b^+ correspondantes. Chaque extrémité sélectionnée est alors une candidate potentielle pour être utilisée dans la détermination d'un bord. Soit $\mathbf{p}_a$ une des extrémités sélectionnées d'un demi-plan D (Figure 5.3). Pour qu'un bord soit confirmé, les étapes suivantes sont effectuées :

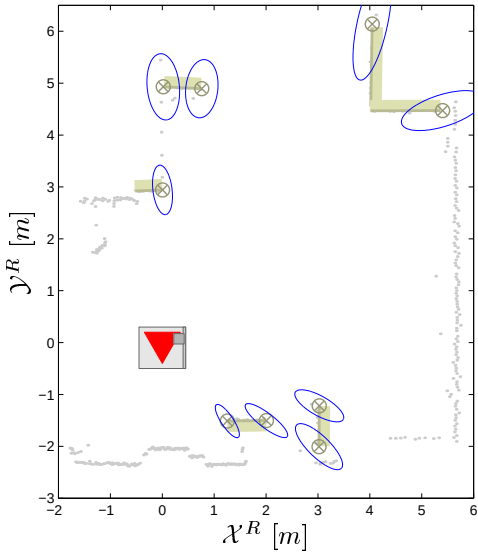
- I. *Estimer les paramètres $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$ du point indiquant la position probable du bord.* Les coordonnées $\mathbf{p}$ sont déterminées selon le principe décrit par la Figure 5.3. En accord avec la



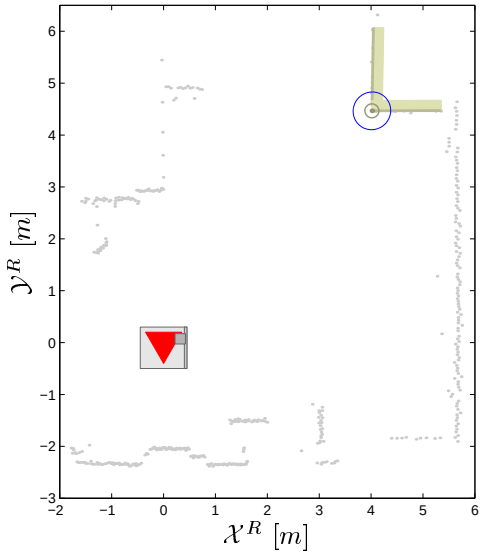
(a) Indices télémétriques



(b) Demi-plans



(c) Arêtes de type bord et demi-plan associé



(d) Arête de type coin et demi-plans associés

FIG. 5.1 – Construction d'une carte locale par traitement d'indices télémétriques. Les ellipsoïdes d'incertitude ont été agrandis pour une meilleure visualisation.

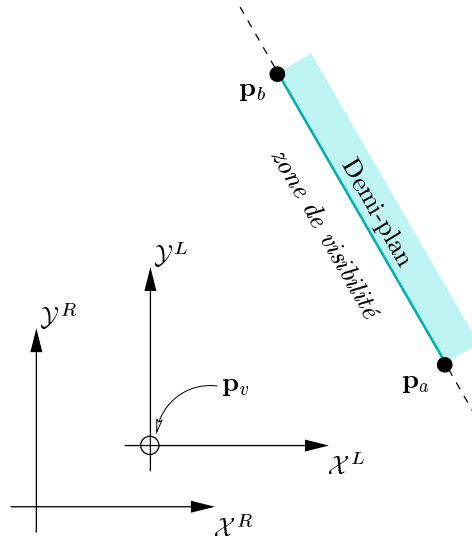


FIG. 5.2 – Détermination des paramètres des demi-plans.

section 4.3.4.1, l'extrémité $\mathbf{p}_a$ est un point appartenant au segment S du demi-plan D . La ligne d'observation de $\mathbf{p}_a$ est représentée par O_a . $\mathbf{p}_a$ étant un point de cassure fort, le point de mesure précédent (ou postérieur, si $\mathbf{p}_a$ était l'autre extrémité du segment), qui suit la direction O'_a , n'appartient pas à la surface du demi-plan D . En conséquence, le bord de la surface balayée doit se trouver dans l'intervalle entre le point d'intersection de O_a et S (i.e., $\mathbf{p}_a$) et le point d'intersection de O'_a et S (représenté par $\mathbf{p}'_a$). Ainsi, nous considérons que le bord $\mathbf{p}$ se situe à mi-chemin entre $\mathbf{p}_a$ et $\mathbf{p}'_a$. Pour estimer l'incertitude de $\mathbf{p}$, nous utilisons la même approche que pour le cas de l'intersection de droites, décrit dans l'annexe B.4. Ainsi, $\mathbf{p}$ étant défini par l'intersection entre sa ligne d'observation O et la droite du demi-plan D , $\Lambda_{\mathbf{p}}$ est calculé en fonction de $\Lambda_{\mathbf{I}}$ et de Λ_O , la matrice de covariance de O . La matrice Λ_O est déterminée en considérant que la ligne d'observation de $\mathbf{p}$ passe par l'origine du repère $\mathcal{R}^L$, ce que se traduit par une incertitude nulle pour le paramètre ρ de la droite. Nous considérons aussi que par le fait que $\mathbf{p}$ puisse se trouver à n'importe quelle position entre $\mathbf{p}_a$ et $\mathbf{p}'_a$, l'écart type associé au paramètre α de la droite d'observation doit être compatible avec la résolution $\Delta\phi$ du télémètre. Ainsi, pour garantir que Λ_O soit compatible avec le raisonnement 3σ sur l'intervalle de $\mathbf{p}$, nous suggérons que Λ_O soit donnée par

$$\Lambda_O = \begin{pmatrix} 0 & 0 \\ 0 & (\Delta\phi/3)^2 \end{pmatrix}.$$

- II. *Valider le bord en fonction de sa fiabilité.* La présence de points de cassure forts dans l'image télémétrique n'est pas une condition suffisante pour la détection des discontinuités (cf. section 4.5.2.1). En effet, nous avons rajouté une heuristique basée sur leur proximité par rapport au capteur pour sélectionner ceux de condition d'observation plus fiable. Ainsi, nous avons mené une évaluation avec des données réelles, dans lesquelles des points de cassure forts observés à moins de deux mètres du capteur ont bien indiqué la présence de bords. Cependant, nous pensons que cette heuristique est assez faible, et nous l'avons utilisée dans la section 4.5.2.1

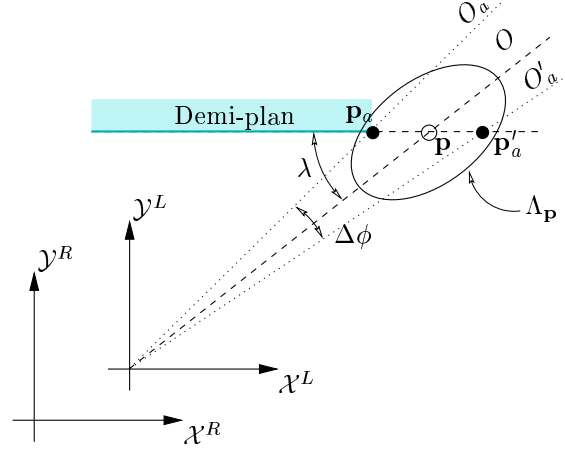


FIG. 5.3 – Détermination des paramètres des arêtes de type bord.

dans un objectif purement didactique. En fait, la fiabilité de détection d'un bord dépend de l'angle d'incidence λ de sa ligne d'observation O avec le demi-plan. Elle suppose aussi une incertitude réduite du paramètre $\mathbf{p}$, ce qui n'est pas le cas pour de faibles valeurs de λ . Ainsi, nous proposons qu'un bord ne soit validé que si $\lambda \geq 30^\circ$ et $\sqrt{e_{\max}(\Lambda_{\mathbf{p}})} \leq 0,1 \text{ m}$, où $e_{\max}(\Lambda_{\mathbf{p}})$ correspond à la plus grande valeur propre de la matrice de covariance $\Lambda_{\mathbf{p}}$.

Un bord confirmé est inclus dans la carte $\mathcal{M}^R$ avec ses paramètres $A = \{P, \kappa^A, n_1^A, n_2^A\}$ où $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$ représente la position et son incertitude dans le repère $\mathcal{R}^R$, $\kappa^A = \mathbf{B}$ et n_1^A reçoit l'index du demi-plan auquel le bord est attaché. Le paramètre n_2^A n'a pas d'importance pour ce type d'indice.

Soit l'exemple de la Figure 5.1(a), pour lequel les arêtes de type bord obtenues sont illustrées sur la Figure 5.1(c). Les ellipsoïdes d'incertitude des bords sont volontairement agrandis pour faciliter leur observation. Nous pouvons vérifier que toutes les extrémités de segments ne sont pas utilisées dans la détermination des bords, seuls les cas le plus sûrs ont été exploités.

5.2.1.3 Détermination des arêtes de type coin

Les structures de type coin sont sensées représenter les angles des murs de l'environnement. Ce type de structure est caractérisé par un point 2-D représentant l'intersection des segments de deux demi-plans (cf. Figure 5.4). Pour leur détermination, nous utilisons en plus des segments extraits, l'image télémétrique initiale sous sa forme cartésienne $\mathcal{C}$ (équation (4.8)). En effet, nous cherchons dans cette image deux points de balayage consécutifs qui ne sont pas des points de cassure, mais qui ont été utilisés pour générer des extrémités de segments. Les segments des demi-plans associés à ces points sont en effet candidats pour le calcul d'un coin. Ainsi, soit deux extrémités représentées par $\mathbf{p}_a$ et $\mathbf{p}_b$ satisfaisant les conditions qui nous venons de décrire (cf. Figure 5.4). A partir des demi-plans associés, nous calculons l'angle λ entre les segments associés, qui est la seule mesure

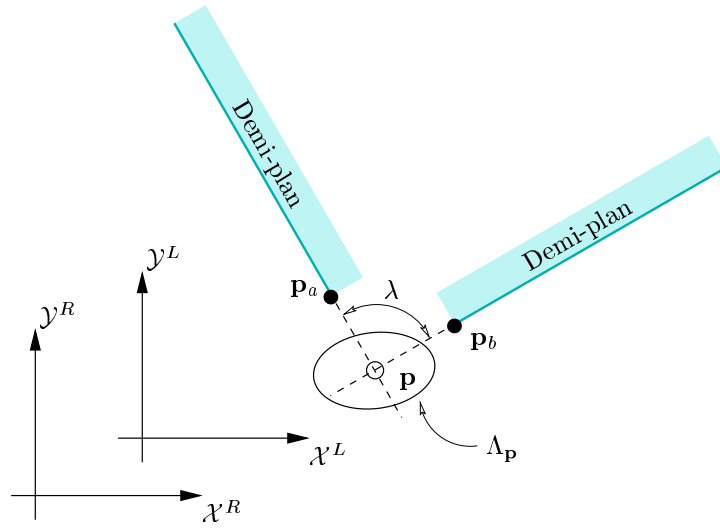


FIG. 5.4 – Détermination des paramètres des arêtes de type coin.

utilisée pour valider un coin. Ainsi, un coin est validé si $60^\circ \leq \lambda \leq 120^\circ$, et le centre $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$ est calculé comme étant le point d'intersection des droites (cf. annexe C.3.2). Pour compléter les paramètres d'une arête $A = \{P, \kappa^A, n_1^A, n_2^A\}$, nous faisons $\kappa^A = \mathbf{C}$, n_1^A et n_2^A reçoivent les index des demi-plans auxquels le coin est attaché.

Pour l'exemple de la Figure 5.1(a), nous n'avons détecté qu'un seul coin, comme nous pouvons voir sur la Figure 5.1(d).

5.2.2 Fusion avec les indices vidéo

Nous utilisons les indices vidéo pour deux raisons distinctes : (i) réduire l'incertitude des arêtes de type bord déjà présentes sur la carte et (ii) déterminer des arêtes photométriques. Pour une explication plus concrète de la fusion d'indices vidéo, nous proposons d'utiliser des données réelles comme celles de la Figure 5.5(a). Cette figure présente une image vidéo et les deux droites verticales U_1 et U_2 qui en sont extraites. Dans le plan laser, ces indices sont observés suivant les lignes d'observation $\hat{O}_1$ et $\hat{O}_2$, issues de l'origine du repère laser $\mathcal{R}^L$, et d'inclinaison angulaire $\hat{\phi}_1$ et $\hat{\phi}_2$ par rapport à l'axe χ^L . Les angles $\hat{\phi}_1$ et $\hat{\phi}_2$ sont en effet des estimées obtenues à partir de l'étalonnage du système multicapteur (annexe F). Cet étalonnage nous a permis d'identifier une fonction reliant l'index u d'une droite verticale de l'image vidéo à son angle d'observation $\hat{\phi}$ dans le repère télémètre. En effet, $\hat{\phi}$ représente l'angle de balayage de ce capteur. La fonction identifiée s'écrit sous la forme :

$$\hat{\phi}(u) = \hat{a} \cdot u + \hat{b} \quad (5.1)$$

dans laquelle $\hat{a}$ et $\hat{b}$ sont numériquement donnés par l'équation (F.6). La variance $\Lambda_{\hat{\phi}}$ de $\hat{\phi}$ est estimée suivant équation (F.9). Sur la Figure 5.5(b) nous avons la carte locale obtenue à partir des procédures décrites précédemment dans la section 5.2.1. A ce stade, les indices visuels n'ont

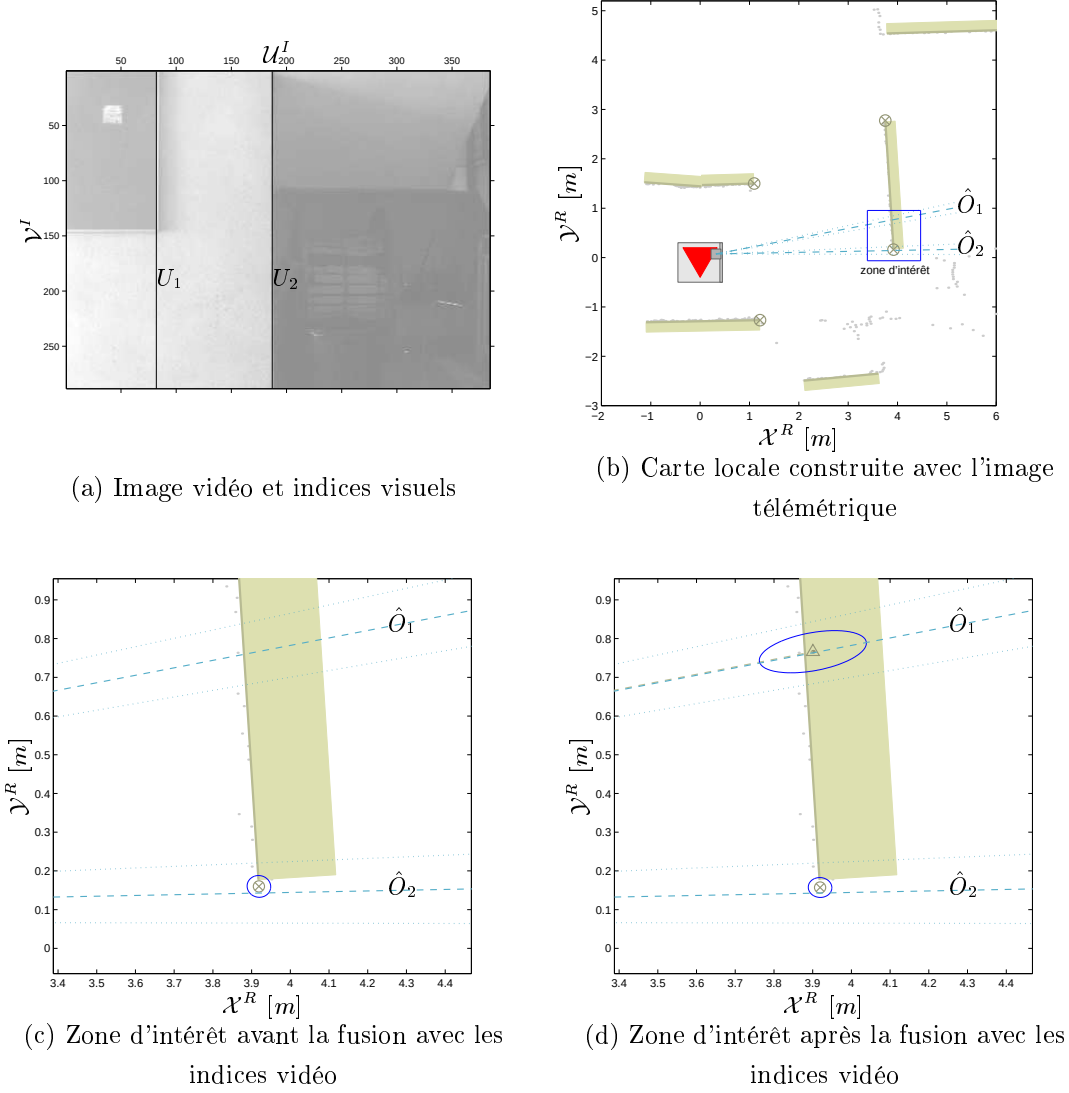


FIG. 5.5 – Fusion d'indices vidéo dans la construction de cartes locales

pas encore été exploités. Les lignes d'observation $\hat{O}_1$ et $\hat{O}_2$ sont représentées par des tirets sur la Figure 5.5(b), et les tracés en pointillés indiquent les limites d'incertitude à 3σ . La zone d'intérêt pour la fusion de données est celle délimitée par un rectangle sur la Figure 5.5(b).

5.2.2.1 Mise à jour des arêtes

La mise à jour des arêtes de la carte locale en utilisant les mesures vidéo requiert une procédure de mise en correspondance. En effet, cette procédure permet de trouver toute mesure vidéo U_j qui correspond à l'observation d'une arête A_i . Ainsi, nous effectuons pour chaque paire (A_i, U_j) , le test statistique suivant :

$$D_{\chi^2}(\{\phi_i, \Lambda_{\phi_i}\}, \{\hat{\phi}_j, \Lambda_{\hat{\phi}_j}\}) \leq Q_1(0, 95), \quad (5.2)$$

où ϕ_i est l'angle de vue de l' i -ième arête (voir section C.3.1), $\hat{\phi}_j$ est l'angle d'observation du j -ième indice vidéo, $D_{\chi^2}(\phi_i, \hat{\phi}_j)$ est la distance de Mahalanobis entre ϕ_i et $\hat{\phi}_j$ et $Q_1(0, 95)$ est donnée par la loi du χ^2_1 (voir section B.5). ϕ_i est calculé à l'origine du repère $\mathcal{R}^L$, dans lequel $\hat{\phi}_j$ est mesuré. Les correspondances étant trouvées, nous imposons que les conditions suivantes soient satisfaites pour que la mise à jour de A_i soit réalisée :

- Si A_i est de type bord, alors sa mise à jour est réalisée seulement si A_i n'a qu'un seul correspondant U_j . Si nous avons des correspondances multiples, elles ne sont pas prises en considération pour la mise à jour de A_i . Les correspondances multiples sont également ignorées dans la phase de détection d'arêtes photométriques (section 5.2.2.2) ;
- Si A_i est de type coin, toutes les mesures vidéo correspondantes sont ignorées (ainsi que les correspondances multiples).

Nous remarquons que seules les arêtes de type bord sont mises à jour par les mesures vidéo. En effet, ces structures ont été initialisées en utilisant un calcul empirique dont la précision dépend fortement de la résolution angulaire du télémètre. Les mesures vidéo peuvent contribuer à réduire leur incertitude. Par contre, les observations vidéo correspondant à des arêtes de type coin sont ignorées. En effet, étant donné que la position de ces structures est assez bien définie par leurs demi-plans supports, l'image vidéo n'apporte pas d'aide supplémentaire dans leur détermination.

Soit $P_i = \{\mathbf{p}_i = (x_i, y_i)^T, \Lambda_{\mathbf{p}_i}\}$ le point représentant le bord A_i qui est en correspondance avec U_j et qui satisfait les conditions ci-dessus. $\hat{\phi}_j$ est une observation de l'angle de vue ϕ_i de P_i . La mise à jour des paramètres de P_i est effectuée en utilisant le formalisme du filtre de Kalman étendu (section E.4) appliqué aux équations suivantes :

$$\begin{aligned} \text{modèle de mesure :} \quad & \mathbf{y} = \mathbf{h}(\mathbf{x}) = \arctan(\mathbf{x}_2 - y_S^R / \mathbf{x}_1 - x_S^R), \\ \text{prédiction :} \quad & \hat{\mathbf{x}}(k|k-1) = \mathbf{p}_i, \quad \Lambda_{\hat{\mathbf{x}}}(k|k-1) = \Lambda_{\mathbf{p}_i}, \\ \text{mesure :} \quad & \mathbf{y}(k) = \hat{\phi}_j, \quad \Lambda_{\mathbf{y}}(k) = \Lambda_{\hat{\phi}_j}. \end{aligned}$$

Revenons maintenant à notre exemple réel dont la zone d'intérêt est agrandie sur la Figure 5.5(c). Nous pouvons vérifier sur cette figure que la ligne d'observation $\hat{O}_2$ est en correspondance avec une arête de type bord. En effet, l'indice U_2 de l'image vidéo correspond à un bord d'un mur qui se trouve en face de la caméra (voir Figure 5.5(a)). L'ellipse d'incertitude 3σ associée à ce bord est affichée. Une fois ce bord mis à jour par la mesure $\hat{\phi}_2$, nous obtenons le résultat illustré sur la Figure 5.5(d). Ce résultat a une incertitude légèrement réduite suivant la direction $\hat{\phi}_2$.

5.2.2.2 Détermination des arêtes de type indice photométrique

Les arêtes photométriques sont obtenues à partir des mesures visuelles qui ne sont pas en correspondance avec des arêtes déjà présentes sur la carte locale. Par exemple, U_1 de l'image vidéo sur la Figure 5.5(a) ne correspond pas à l'observation d'une arête de la carte locale, ce qui fait de cette mesure un candidat pour la détermination d'une arête photométrique. Nous notons que les indices photométriques n'ont pas de rapport avec les discontinuités des surfaces balayées par le télémètre, comme c'est le cas des arêtes de type bord. Cependant, ces indices peuvent être exploités pour enrichir la carte de l'environnement.

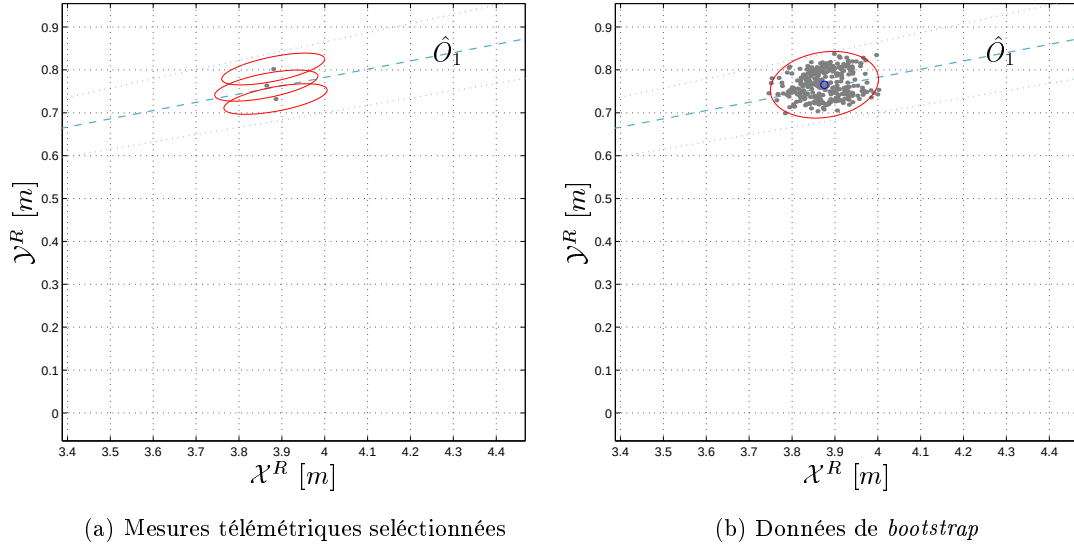


FIG. 5.6 – Initialisation de l'incertitude d'une arête photométrique

L'insertion d'une mesure vidéo U sous forme d'une arête photométrique de centre P dans la carte locale implique la détermination, dans le repère local, de l'intersection de la ligne d'observation de U avec l'élément 3-D qui l'a généré. En effet, un indice vidéo U fournit seulement une estimée $\hat{\phi}$ de l'angle de vue de P . Si nous considérons l'indice U_1 de la Figure 5.5(a), l'objet générateur est en fait l'arête d'un tableau fixé sur un mur. Ce mur est représenté sous la forme d'un demi-plan dans la carte locale. Cependant, nous n'exploitons pas cet indice pour la détermination de l'arête photométrique. En effet, nous utilisons pour initialiser P les mesures télémétriques brutes qui correspondent à l'indice vidéo. Pour cela, on emploie une technique connue en statistique qui s'appelle le *bootstrap* [74][219]. Cette méthode nous a semblé tout à fait adaptée pour l'initialisation tant de la moyenne $\mathbf{p}$ de la distribution de P , que de sa matrice de covariance $\mathbf{\Lambda}_p$. Elle consiste à utiliser l'incertitude des mesures correspondant à P pour générer un ensemble aléatoire de points qui serviront à caractériser sa distribution. Nous supposons que P est le point moyen parmi les points instruments. Ainsi, les mesures télémétriques correspondant au j -ième indice vidéo U_j satisfont

$$D_{\chi^2}(\{\phi_i, \Lambda_{\phi_i}\}, \{\hat{\phi}_j, \Lambda_{\hat{\phi}_j}\}) \leq Q_1(0, 95), \quad (5.3)$$

où ϕ_i est l'angle de balayage pour l' i -ième mesure télémétrique correspondant à U_j , $\hat{\phi}_j$ est l'angle d'observation de U_j , et Λ_{ϕ_i} et $\Lambda_{\hat{\phi}_j}$ sont les variances associées. Cependant, dans la section 3.2, nous avons présenté le télémètre laser comme un capteur sans incertitude sur l'angle de balayage ϕ_i , ce qui semble contradictoire avec l'idée actuelle de lui associer une matrice de covariance Λ_{ϕ_i} . En fait, nous cherchons à associer une mesure vidéo à une mesure télémétrique, sachant que ces dernières sont acquises avec un pas d'échantillonnage angulaire $\Delta\phi$. Cela veut dire qu'un indice vidéo peut se trouver entre deux mesures télémétriques consécutives, séparées par $\Delta\phi = 0,6^\circ$. Ainsi, de manière intuitive, suivant un raisonnement 3σ , on peut définir $\Lambda_{\phi_i} = (\frac{1}{3}\Delta\phi)^2$.

Une fois les mesures télémétriques correspondant à l'observation vidéo U_j déterminées par

(5.3), nous générons les échantillons qui serviront à initialiser P . Soit $\mathbf{p}_i = (x_i, y_i)$ les coordonnées cartésiennes de l' i -ième mesure télémétrique (r_i, ϕ_i) , avec $x_i = r_i \cdot \cos(\phi_i)$ et $y_i = r_i \cdot \sin(\phi_i)$. L'incertitude associée à $\mathbf{p}_i$ peut être estimée de façon consistante en suivant un simple raisonnement 3σ . La matrice de covariance associée a des axes principaux de longueur σ_r^2 suivant la direction ϕ_i , et $(\frac{1}{3}r_i \cdot \tan(\Delta\phi))^2$ suivant $\phi_i + \pi/2$. Ainsi, d'après l'équation (B.6), la matrice de covariance de l'erreur sur $\mathbf{p}_i$ est estimée à partir de ses axes principaux :

$$\begin{aligned} \Lambda_{\mathbf{p}_i} = & \left(\frac{1}{3}r_i \cdot \tan(\Delta\phi) \right)^2 \cdot \begin{pmatrix} \cos(\phi_i + \pi/2) & \sin(\phi_i + \pi/2) \end{pmatrix} \cdot \begin{pmatrix} \cos(\phi_i + \pi/2) \\ \sin(\phi_i + \pi/2) \end{pmatrix} \\ & + \sigma_r^2 \cdot \begin{pmatrix} \cos(\phi_i) & \sin(\phi_i) \end{pmatrix} \cdot \begin{pmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{pmatrix}. \end{aligned} \quad (5.4)$$

Pour mieux illustrer ce que nous venons de décrire, nous présentons sur la Figure 5.6(a) les ellipsoïdes 3σ des distributions gaussiennes des mesures télémétriques correspondant à l'indice vidéo U_1 de l'exemple de la Figure 5.5(a). En partant de ces mesures, notre procédure de *bootstrap* génère de façon aléatoire $N = 100$ échantillons $\check{\mathbf{p}}_{n,i}$ ($n = 1, \dots, N$) pour chaque mesure $\mathbf{p}_i$, suivant une distribution gaussienne de moyenne $\mathbf{p}_i$ et covariance $\Lambda_{\mathbf{p}_i}$. Notre initialisation pour $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$ est alors donnée par

$$\mathbf{p} = \frac{1}{\check{N} - 2} \sum_i \sum_n \check{\mathbf{p}}_{n,i}, \quad (5.5)$$

$$\Lambda_{\mathbf{p}} = \frac{1}{\check{N} - 2} \sum_i \sum_n (\check{\mathbf{p}}_{n,i} - \mathbf{p}) \cdot (\check{\mathbf{p}}_{n,i} - \mathbf{p})^T, \quad (5.6)$$

où $\check{N}$ est le nombre total d'échantillons générés en prenant en compte toutes les mesures télémétriques en question. La Figure 5.6(b) montre les échantillons générés à partir des mesures télémétriques de la Figure 5.6(a), ainsi que l'ellipse d'incertitude 3σ associée à $\mathbf{p}$. Même si l'initialisation obtenue pour cet exemple nous semble tout à fait consistante avec les instruments $\check{\mathbf{p}}_{n,i}$, cela ne veut pas dire que l'indice observé est intéressant pour la cartographie. En effet, il y a des cas où les mesures télémétriques correspondantes sont très éloignées les unes des autres, ce qui se produit souvent dans des environnements très encombrés et mal structurés. Bien entendu, ces indices ne nous intéressent pas parce que nous ne pouvons pas initialiser leur point central P de façon fiable. Une solution plus simple aurait été d'utiliser seulement la mesure télémétrique la plus proche au sens de la distance (5.3). Cependant, nous croyons qu'utiliser l'ensemble des mesures correspondantes nous permet d'initialiser P de façon consistante, et de rejeter les indices vidéo peu fiables (par exemple, ceux issus de régions encombrées). Ainsi, nous proposons de ne garder comme initialisation de P que celle qui a eu au moins deux mesures télémétriques correspondantes, à condition que la plus grande valeur propre $e_{\max}$ de $\Lambda_{\mathbf{p}}$ respecte

$$e_{\max}(\Lambda_{\mathbf{p}}) \leq (0,1 \text{ m})^2. \quad (5.7)$$

Cette initialisation du point central P d'une arête photométrique nous sert seulement de point de départ pour la détermination de ce type de structure cartographique. La matrice de covariance

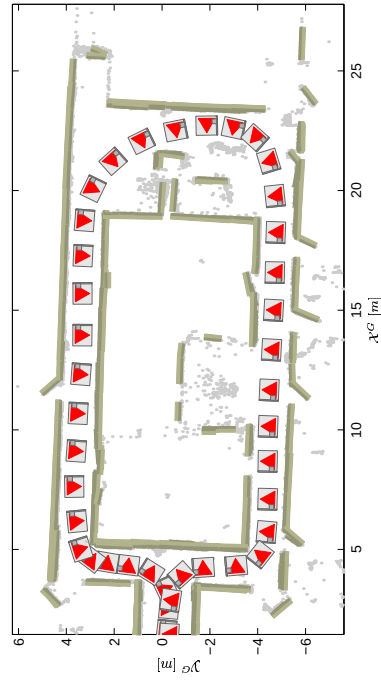
calculée par le *bootstrap* s'étend même au delà des limites 3σ d'observation de l'indice vidéo (cf. Figure 5.6(b)). La consistance de Λ_P est pourtant garantie, au moins suivant la ligne d'observation vidéo. A ce stade, l'indice vidéo n'est pas encore intervenu dans la détermination de l'arête photométrique. En effet, nous avons ici le même cas de figure que celui d'une arête de type bord observée par la caméra vidéo (section 5.2.2.1) : le point central P de l'arête est initialisé en utilisant les mesures télémétriques. Nous procédons ensuite à la mise à jour de ce point P en utilisant l'observation vidéo U_j correspondante. Cela est effectué en utilisant le filtre de Kalman étendu, de la même façon que pour la mise à jour des arêtes de type bord décrite dans la section 5.2.2.1. Les paramètres de la nouvelle arête photométrique sont : $A = \{P, \kappa^A, n_1^A, n_2^A\}$, avec $\kappa^A = P$, où n_1^A et n_2^A ne sont pas utilisés étant donné que ce type de structure n'est pas associé à un demi-plan.

La Figure 5.5(d) montre l'arête de type indice photométrique obtenue à partir de l'observation vidéo U_1 . Nous remarquons que cet indice, observé à une distance d'à peu près quatre mètres, a une incertitude relativement faible. Ce sera une "balise" intéressante pour le processus de localisation.

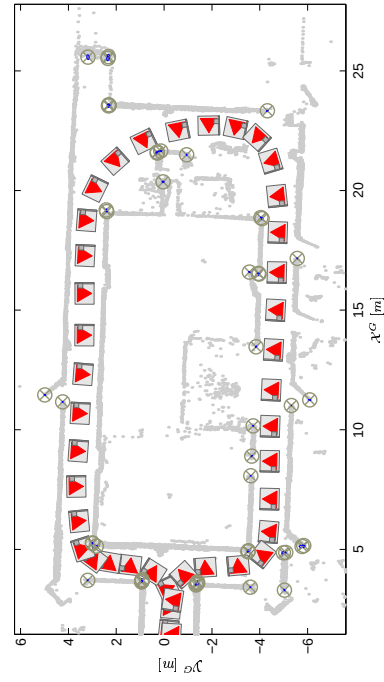
5.3 Evaluation expérimentale

Notre objectif ici est de vérifier la qualité de la carte locale par une approche purement visuelle. En effet, nous ne disposons pas de système de mesure de référence qui permettrait une vérification quantitative précise des résultats obtenus. Notre méthode d'évaluation consistera donc à superposer aux mesures télémétriques brutes, toutes les cartes locales construites durant une expérience de navigation dans un trajet fermé. Cette expérience est la même que celle qui a servi de base pour l'évaluation des approches de détection de points de cassure (section 4.5.2.1). La position du robot est estimée par gyrodométrie. Nous présentons sur la Figure 5.7 les superpositions des images télémétriques et des structures cartographiques obtenues en utilisant des paires d'images $(\mathcal{L}, \mathcal{I})$ acquises à différentes positions du robot. Ainsi, les demi-plans affichés sur la Figure 5.7(a) correspondent bien aux murs de l'environnement réel. L'algorithme de segmentation de l'image télémétrique a employé le détecteur adaptatif pour les points de cassure et le noyau SMF pour l'extraction de droites. Les arêtes de type bord, coin et photométrique, ainsi que leurs ellipses d'incertitude 3σ , sont montrées sur les Figures 5.7(b)-(d). Dans cette superposition de cartes locales, obtenues à différents points de vue, il apparaît que les observations multiples des arêtes sont stables et constantes les unes par rapport aux autres. Nous remarquons aussi qu'en général les incertitudes des indices photométriques sont plus importantes que celles des bords et des coins. Cela est dû au modèle simplifié utilisé pour calculer les directions d'observation des mesures vidéo dans le repère du télémètre (annexe F).

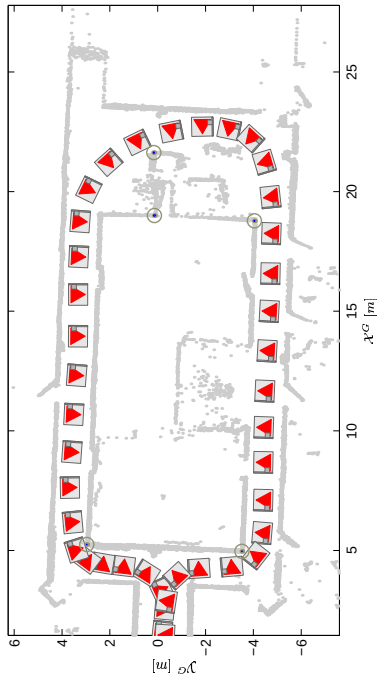
Regardons maintenant plus en détail sur la Figure 5.8 des arêtes photométriques extraites de la Figure 5.7(d). La Figure 5.8(a), met en évidence trois ensembles d'arêtes photométriques (indiqués par A, B et C). Ces structures correspondent bien à trois contours verticaux de l'environnement qui sont les bords d'une porte fermée et l'angle d'un mur (cf. Figure 5.8(b)). Nos remarquons que cet angle traduit ici sous la forme d'une arête photométrique (A), alors que dans la Figure 5.7(b) ce même angle est identifié sur des cartes locales différentes comme un bord. La Figure 5.8(c) nous montre une charnière de porte qui se traduit dans les cartes locales par différentes sous-types



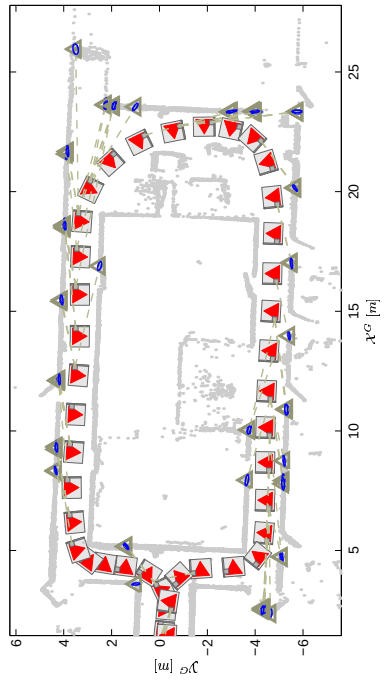
(a) Demi-plans



(b) Arêtes de type bord



(c) Arêtes de type coin



(d) Arêtes de type indice photométrique

FIG. 5.7 – Superposition des structures de toutes les cartes locales extraites durant une expérience suivant un trajet fermé. Les positions du robot correspondantes aux instants d'acquisition d'images comme aussi les images télémétriques sont aussi bien illustrés

d'arêtes (indiqués par D). Et pourtant, en réalité, cette structure est de type coin. Une identification correcte de cette structure n'a pas été possible parce que les deux murs qui lui servent de support n'ont jamais été détectés en même temps sur la même carte locale. Cette analyse nous permet de conclure qu'un objet de l'environnement correspondant à un point 2-D dans la carte locale peut être reconnu soit comme un bord, soit comme un coin, ou encore comme un indice photométrique. Nous verrons par la suite qu'il est fondamental pour la cartographie globale (chapitre 7) de garder cette classification des arêtes en sous-types.

Un autre exemple intéressant est celui de la Figure 5.8(c), où une flèche indique un grand nombre d'arêtes photométriques qui semblent correspondre au même objet. Et pourtant, ces structures correspondent bien à des objets différents, comme nous pouvons le voir sur l'image vidéo de la Figure 5.8(d). En fait, dans cette position du robot nous observons dans l'image vidéo un grand nombre de droites verticales relativement proches. Il est clair que la résolution du télémètre laser n'est pas suffisante pour qu'on puisse bien discerner les structures observées. Ce cas d'observations multiples mérite un traitement spécifique. En effet, nous verrons que les approches de localisation (chapitre 6) et de cartographie (chapitre 7) présentées dans ce manuscrit sont capables de gérer les observations ou correspondances multiples.

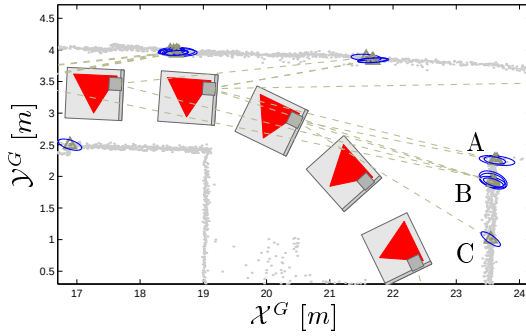
Enfin, le temps de calcul mesuré pour la construction des cartes locales en chacune des 48 positions est indiqué sur la Figure 5.9. Nous soulignons que ce temps comprend la segmentation des données multicapteur et la construction de la carte locale. Ces mesures sont comprises entre 235 *ms* et 350 *ms*.

5.3.1 Contre-exemple de la détection d'arêtes photométriques

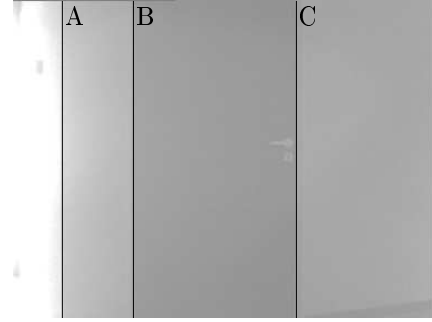
Nous avons vu dans la section précédente qu'un indice photométrique a une incertitude d'observation en général plus importante que celle d'une arête de type bord ou de type coin. Pour le localiser dans la carte télémétrique, nous avons fait l'hypothèse que le contour vertical de l'image correspond à une structure mesurée par le télémètre. On utilise un test statistique sur les mesures télémétriques acquises dans la direction du contour vertical observé dans l'image vidéo. Les paramètres de l'arête sont déterminés par bootstrap, et ensuite rajoutée à la carte locale si son incertitude est inférieure à un seuil.

Cependant, cette méthode de détection n'est pas sans faille. Nous montrons ici un contre-exemple qui illustre bien le fait qu'un test statistique seul ne suffit pas pour confirmer la correspondance entre les mesures télémétriques et le contour vertical. En effet, compte tenu du fait que les champs d'observation des deux capteurs sont décalés suivant une direction verticale, dans certains cas le télémètre balaye une surface de l'espace qui ne correspond pas à celle contenant le contour vertical observé par la caméra. Un tel cas est illustré sur la Figure 5.10. Pour plus de clarté, nous avons montré dans cette illustration les repères télémètre et caméra vidéo dans une configuration différente de celle de notre système multicapteur. Dans ce cas, notre algorithme génère une arête photométrique erronée (numéro 4).

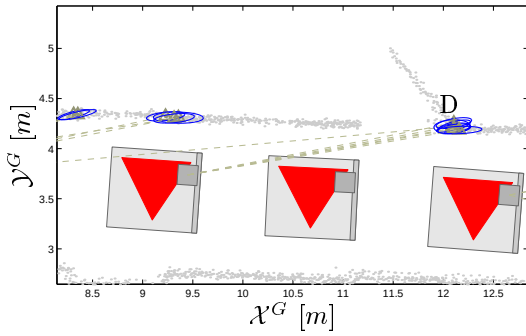
Une solution existe pour pallier ce problème, mais elle n'a pas été implémentée actuellement. En effet, un calibrage plus complet du système multicapteur permettrait de détecter ces cas pour



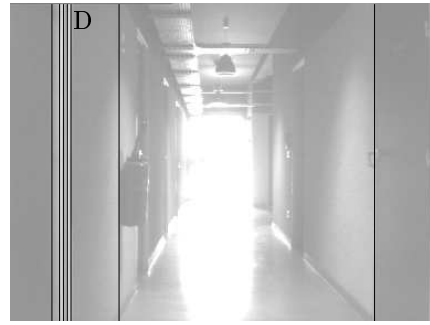
(a) Arêtes photométriques



(b) Image vidéo



(c) Arêtes photométriques



(d) Image vidéo

FIG. 5.8 – Cas de figure concernant les arêtes photométriques : (a)-(b) cas montrant l'intérêt d'associer un sous-type aux arêtes ; (c)-(d) cas qui illustre les observations multiples.

les éviter. A partir de l'intervalle $[v_{\min}, v_{\max}]$ des lignes de l'image vidéo auquel appartiennent les pixels du contour vertical, nous pourrions délimiter l'intervalle de distances $[r_{\min}, r_{\max}]$ auquel les mesures télémétriques doivent appartenir. Une approche semblable a été présentée par Lobo *et al.* [149] dans un système de détection d'indices verticaux avec une paire d'images stéréo.

5.4 Conclusions

Dans ce chapitre nous avons présenté une approche pour la construction de cartes locales par fusion d'indices géométriques obtenus par segmentation de données multisensorielles. La carte locale obtenue est formée par deux structures de base qui sont des demi-plans et des arêtes. Un demi-plan est modélisé non seulement par un segment de droite, mais aussi par un ensemble de paramètres permettant de vérifier si un point donné de l'environnement se trouve dans sa région de visibilité. Les arêtes sont classées en trois sous-types : les bords toujours associés à un demi-plan, les coins associés à deux demi-plans adjacents, et les indices photométriques. Les arêtes photométriques

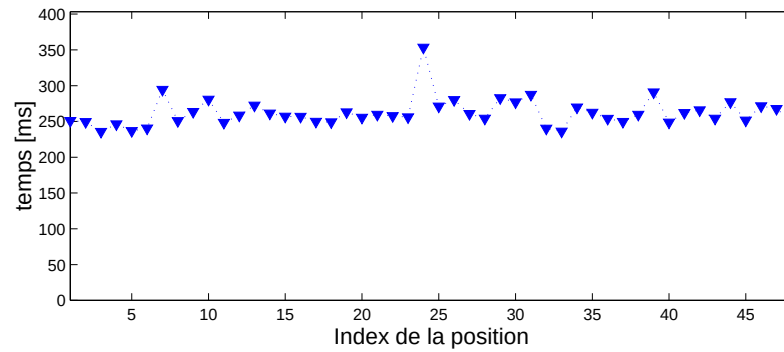


FIG. 5.9 – Temps de calcul utilisé pour la segmentation des données multicapteur et la construction des cartes locales dans l'expérience précédente.

permettent de modéliser des contours verticaux observés par la caméra. Leur position dans la carte est déterminée à l'aide des mesures télémétriques en utilisant une méthode statistique de *bootstrap*.

La détermination de ces structures est accomplie en deux phases : initialisation des demi-plans, bords et coins en utilisant les indices télémétriques, et mise à jour des bords, puis initialisation des arêtes photométriques à l'aide des contours vidéo. Une évaluation expérimentale nous a permis de contrôler visuellement la qualité des structures obtenues, de vérifier la relation entre les différents types d'arêtes et de mettre en évidence le phénomène des observations multiples.

Dans le contexte de notre approche, les cartes locales ont deux objectifs distincts mais fortement corrélés : la localisation et la cartographie globale. Malgré cette corrélation, les problèmes de localisation et de cartographie seront traités de façon indépendante, en s'appuyant sur des outils théoriques récents, comme nous le verrons dans les chapitres 6 et 7.

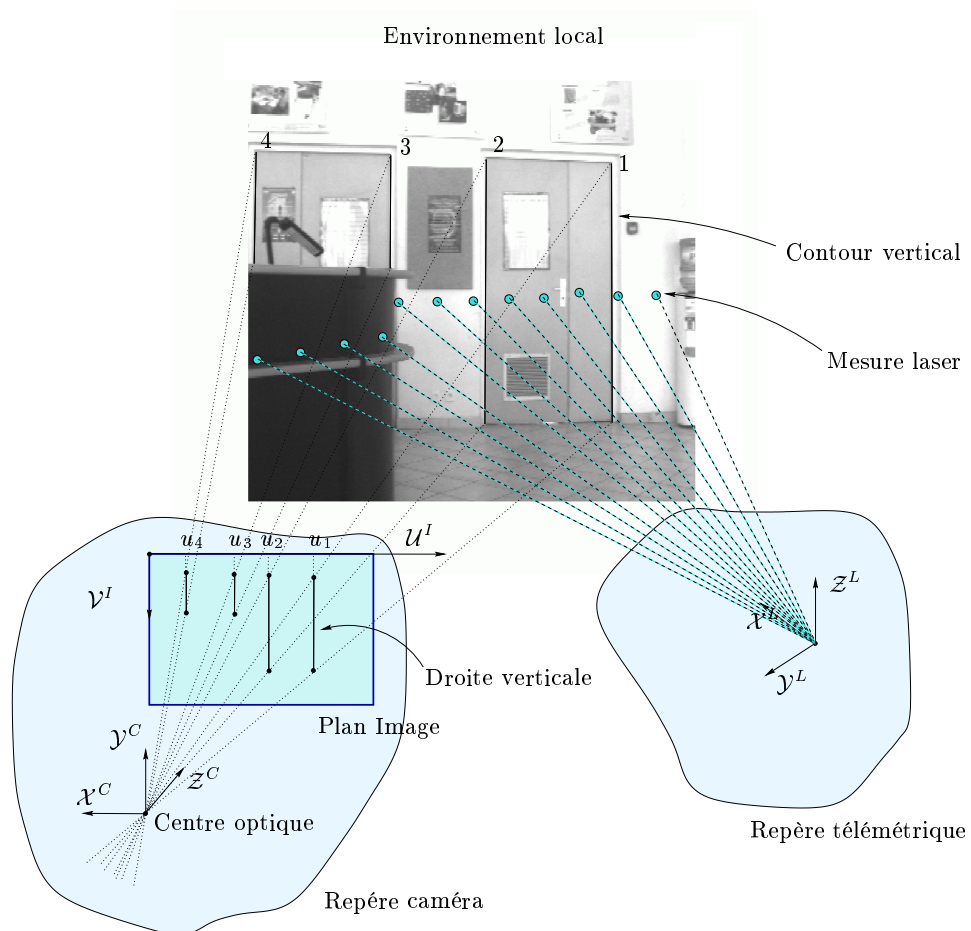


FIG. 5.10 – Contre-exemple de la détection d'arêtes photométriques.

Localisation dynamique robuste

6.1 Introduction

Ce chapitre présente les modules de localisation relative et de recalage, qui composent l'architecture présentée dans la section 1.4 (Figure 1.3). Nous utilisons le concept classique prédiction-correction : la prédiction de position est effectuée par estime à partir des mesures des capteurs proprioceptifs, et la correction utilise la mise en correspondance de cartes géométriques. La contribution majeure de ce chapitre porte sur de nouvelles méthodes robustes de localisation par mise en correspondance de cartes d'environnement. Par le biais d'expériences réelles, elles sont comparées à des méthodes classiques utilisant le filtrage stochastique.

Nous avons implanté ces méthodes dans le cadre de la localisation dynamique, désignée aussi par *on-the-fly localization* [8], qui est adaptée à des robots mobiles se déplaçant à des vitesses non négligeables. Une partie des effets négatifs engendrés par le mouvement du robot sur les mesures capteur sont déjà compensés par les procédures de prétraitement présentées dans la section 3.5.3. Par ailleurs, en prenant en compte le temps ΔT_l écoulé lors de la correction de position (cela comprend la segmentation, la cartographie locale et l'estimation de position), l'estimée $\hat{\mathbf{z}}_k$ résultant de ce cycle fait référence à l'instant $t_{\mathcal{I}}$ associé aux données extéroceptives. La mise à jour de la situation du robot nécessite un recalage temporel avant sa mise à disposition à l'instant $t_{\mathcal{I}} + \Delta T_l$, qui est fait par intégration des mesures proprioceptives acquises pendant ce temps ΔT_l .

6.2 Localisation relative à l'estime

La localisation à l'estime utilise des mesures proprioceptives pour mettre à jour l'estimée $\hat{\mathbf{z}}_k$ à des instants discrets k . Sur le robot Omni, deux types de capteurs servent à cette fin : les codeurs incrémentaux couplés aux axes des roues, et le gyromètre laser. Ces capteurs permettent une mise à jour de $\hat{\mathbf{z}}_k$ à chaque 5 ms.

Dans notre étude, nous exploitons deux modalités de localisation à l'estime : l'odométrie, utilisant seulement les données codeurs, et la gyrodométrie, qui incorpore les mesures du gyromètre laser à l'odométrie. Ces techniques sont basées sur l'intégration du modèle cinématique du robot.

6.2.1 Modèle cinématique du robot Omni

La modélisation cinématique permet d'identifier des relations existant entre les variables de configuration $\mathbf{q}$ du robot, et sa position $\mathbf{z}$ dans l'espace cartésien de navigation. Le robot Omni est équipé de trois roues motrices et directrices non-centrées (cf. Figure 6.1). Nous supposons que ces axes passent par les sommets A_1 , A_2 et A_3 d'un triangle se trouvant sur le plan $\mathcal{X}^R \times \mathcal{Y}^R$. Les paramètres géométriques associés à la roue i sont :

- r_i , son rayon ;
- e_i , la distance du point A_i à l'axe de la roue ;
- Les coordonnées cartésiennes (x_{a_i}, y_{a_i}) ou polaires (l_i, α_i) du point A_i dans le repère robot.

Des mesures effectuées sur Omni nous indiquent les valeurs suivantes pour ces paramètres :

$$\lambda_0 : \begin{cases} r_1 = 0,1 \text{ m} & e_1 = 0,05 \text{ m} & x_{a_1} = 0,325 \text{ m} & y_{a_1} = 0,188 \text{ m} \\ r_2 = 0,1 \text{ m} & e_2 = 0,05 \text{ m} & x_{a_2} = -0,325 \text{ m} & y_{a_2} = 0,188 \text{ m} \\ r_3 = 0,1 \text{ m} & e_3 = 0,05 \text{ m} & x_{a_3} = 0 \text{ m} & y_{a_3} = -0,376 \text{ m} \end{cases} . \quad (6.1)$$

Si nous considérons $\mathbf{r} = (r_1, r_2, r_3)^T$, $\mathbf{e} = (e_1, e_2, e_3)^T$, $\mathbf{x}_a = (x_{a_1}, x_{a_2}, x_{a_3})^T$ et $\mathbf{y}_a = (y_{a_1}, y_{a_2}, y_{a_3})^T$, les paramètres géométriques sont représentées par $\lambda = (\mathbf{r}^T, \mathbf{e}^T, \mathbf{x}_a^T, \mathbf{y}_a^T)^T$. Les variables de configuration sont représentées par :

$$\mathbf{q} = \left(\beta_1 \quad \beta_2 \quad \beta_3 \quad \varphi_1 \quad \varphi_2 \quad \varphi_3 \right)^T, \quad (6.2)$$

où β_i est l'angle de direction de la roue i et φ_i sa position angulaire. La vitesse angulaire $\dot{\varphi}_i$ et l'angle de direction de chaque roue sont commandés indépendamment par des moteurs à courant continu.

Le modèle cinématique inverse est le plus simple à obtenir pour ce genre de robot :

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}, \lambda, \theta) \cdot \dot{\mathbf{z}}. \quad (6.3)$$

Les contraintes cinématiques sont données pour la roue i par les équations suivantes [47] :

$$\begin{pmatrix} \cos(\beta_i) & \sin(\beta_i) & l_i \sin(\beta_i - \alpha_i) \end{pmatrix} \mathbf{R}_h(\theta) \dot{\mathbf{z}} - r_i \dot{\varphi}_i = 0, \quad (6.4)$$

$$\begin{pmatrix} \sin(\beta_i) & -\cos(\beta_i) & e_i - l_i \cos(\beta_i - \alpha_i) \end{pmatrix} \mathbf{R}_h(\theta) \dot{\mathbf{z}} + e_i \dot{\beta}_i = 0, \quad (6.5)$$

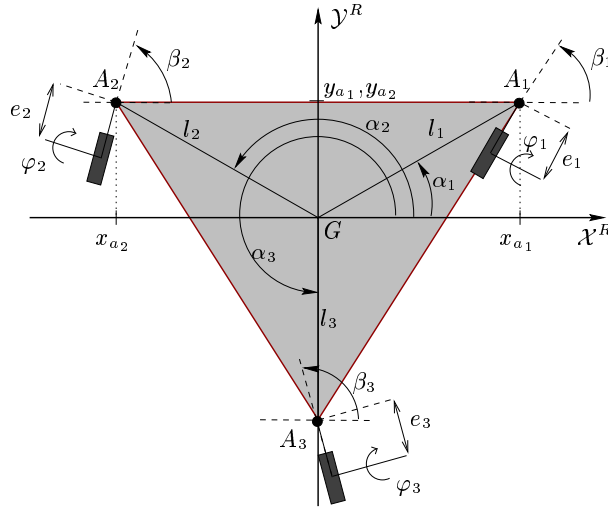


FIG. 6.1 – Variables de configuration du robot Omni

où

$$\mathbf{R}_h(\theta) = \begin{pmatrix} \mathbf{R}(\theta) & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 \end{pmatrix}$$

est la transformation homogène constituée d'une rotation θ représentée par la matrice $\mathbf{R}(\theta)$ (cf. équation (A.2)). $\mathbf{R}_h(\theta)$ est carrée et orthogonale : $\mathbf{R}_h^T(\theta) = \mathbf{R}_h^{-1}(\theta)$.

Dans l'équation (6.3), $\mathbf{J}(\mathbf{q}, \lambda, \theta)$ est la matrice Jacobienne de dimension 6×3 donnée par :

$$\mathbf{J}(\mathbf{q}, \lambda, \theta) = \mathbf{S}(\mathbf{q}, \lambda) \cdot \mathbf{R}_h(\theta), \quad (6.6)$$

avec

$$\mathbf{S}(\mathbf{q}, \lambda) = \begin{pmatrix} \begin{pmatrix} \frac{1}{e_1} & 0 & 0 \\ 0 & \frac{1}{e_2} & 0 \\ 0 & 0 & \frac{1}{e_3} \end{pmatrix} \cdot \begin{pmatrix} -\sin(\beta_1) & \cos(\beta_1) & -e_1 + x_{a1} \cos(\beta_1) + y_{a1} \sin(\beta_1) \\ -\sin(\beta_2) & \cos(\beta_2) & -e_2 + x_{a2} \cos(\beta_2) + y_{a2} \sin(\beta_2) \\ -\sin(\beta_3) & \cos(\beta_3) & -e_3 + x_{a3} \cos(\beta_3) + y_{a3} \sin(\beta_3) \end{pmatrix} \\ \begin{pmatrix} \frac{1}{r_1} & 0 & 0 \\ 0 & \frac{1}{r_2} & 0 \\ 0 & 0 & \frac{1}{r_3} \end{pmatrix} \cdot \begin{pmatrix} \cos(\beta_1) & \sin(\beta_1) & x_{a1} \sin(\beta_1) - y_{a1} \cos(\beta_1) \\ \cos(\beta_2) & \sin(\beta_2) & x_{a2} \sin(\beta_2) - y_{a2} \cos(\beta_2) \\ \cos(\beta_3) & \sin(\beta_3) & x_{a3} \sin(\beta_3) - y_{a3} \cos(\beta_3) \end{pmatrix} \end{pmatrix}.$$

La localisation par odométrie repose sur le modèle cinématique direct, obtenu à partir de l'équation (6.3) :

$$\dot{\mathbf{z}} = \mathbf{J}^\dagger(\mathbf{q}, \lambda, \theta) \dot{\mathbf{q}}, \quad (6.7)$$

dans lequel $\mathbf{J}^\dagger(\mathbf{q}, \theta)$ est la pseudo-inverse de Moore-Penrose de $\mathbf{J}(\mathbf{q}, \theta)$, calculée par

$$\mathbf{J}^\dagger(\mathbf{q}, \lambda, \theta) = (\mathbf{J}^T(\mathbf{q}, \lambda, \theta) \cdot \mathbf{J}(\mathbf{q}, \lambda, \theta))^{-1} \cdot \mathbf{J}^T(\mathbf{q}, \lambda, \theta). \quad (6.8)$$

Etant donné les propriétés de la matrice $\mathbf{R}_h(\theta)$, nous avons

$$\mathbf{J}^\dagger(\mathbf{q}, \lambda, \theta) = \mathbf{R}_h^T(\theta) \cdot \mathbf{S}^\dagger(\mathbf{q}, \lambda). \quad (6.9)$$

$\dot{\mathbf{z}}$ est la solution au sens de moindres-carrés d'un système de 6 équations et 3 inconnues (éq. (6.3)). Le théorème suivant démontré dans [119] illustre une propriété de cette solution :

Théorème 6.1: $\dot{\mathbf{z}} = \mathbf{J}(\mathbf{q}, \lambda, \theta)^\dagger \dot{\mathbf{q}}$ est la solution de norme minimale de $\mathbf{J}(\mathbf{q}, \lambda, \theta) \cdot \dot{\mathbf{z}} = \dot{\mathbf{q}}$ au sens de moindres carrés, c'est-à-dire, la solution $\dot{\mathbf{z}}$ qui minimise

$$V_{MC} = \| \dot{\mathbf{q}} - \mathbf{J}(\mathbf{q}, \lambda, \theta) \cdot \dot{\mathbf{z}} \|^2 . \quad (6.10)$$

◇

Pour déterminer $\dot{\mathbf{z}}$, nous avons besoin de calculer $\mathbf{J}^\dagger(\mathbf{q}, \lambda, \theta)$. Campion *et al.* [47] ont montré que le rang de $\mathbf{J}(\mathbf{q}, \lambda, \theta)$ est toujours 3 pour un robot omnidirectionnel. Ainsi, $\mathbf{J}^\dagger(\mathbf{q}, \lambda, \theta)$ est toujours calculable par l'équation (6.8).

6.2.2 Odométrie

Pour mettre à jour l'estimé $\hat{\mathbf{z}}_k$ de position résultant de la variation de configuration $\Delta \mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_{k-1}$, nous linéarisons au premier ordre le modèle cinématique direct (équation (6.7)) :

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k-1} + \mathbf{J}^\dagger(\mathbf{q}_{k-1}, \lambda, \hat{\theta}_{k-1}) \cdot \Delta \mathbf{q}_k . \quad (6.11)$$

L'estimation de position par odométrie repose sur les hypothèses suivantes :

- I. La configuration géométrique du robot est exactement celle de la Figure 6.1 ;
- II. La discrétisation néglige les composantes d'ordre deux ou supérieur du modèle cinématique ;
- III. Les roues ne glissent pas pendant le déplacement du robot ;
- IV. Le contact de chaque roue avec le sol est ponctuel ;
- V. Le sol sur lequel le robot navigue est parfaitement plan ;
- VI. Les paramètres λ sont connus de manière exacte ;

La non-satisfaction de ces hypothèses contribue à une croissance sans limite de l'erreur de positionnement par odométrie. Or, les hypothèses I-V ne sont jamais parfaitement vérifiées. Dans les sections suivantes, nous nous intéressons à minimiser l'erreur sur $\hat{\theta}_{k-1}$, que ce soit par l'utilisation d'un gyromètre laser, ou par l'identification de λ . En effet, les composantes de $\mathbf{J}^\dagger(\mathbf{q}_{k-1}, \lambda, \hat{\theta}_{k-1})$ étant fortement non-linéaires, des erreurs sur l'estimée $\hat{\theta}_k$ du cap sont re-intégrées dans le calcul à l'instant $k + 1$.

6.2.3 Gyrodométrie

Nous avons utilisé la gyrodométrie comme référence de positionnement pour valider ou comparer les méthodes de localisation absolue développées. En effet, comme l'odométrie, la gyrodométrie repose sur l'intégration du modèle géométrique direct, avec une mesure supplémentaire : le changement de cap fourni par le gyromètre laser. Le modèle cinématique inverse peut être décomposé

de la façon suivante :

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q}, \lambda) \cdot \mathbf{R}_h(\theta) \cdot \dot{\mathbf{z}}, \quad (6.12)$$

$$= \mathbf{S}_{xy}(\mathbf{q}, \lambda) \cdot \mathbf{R}(\theta) \cdot \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} + \mathbf{S}_\theta(\mathbf{q}, \lambda) \cdot \dot{\theta}, \quad (6.13)$$

avec

$$\mathbf{S}_{xy}(\mathbf{q}, \lambda) = \begin{pmatrix} \begin{pmatrix} \frac{1}{e_1} & 0 & 0 \\ 0 & \frac{1}{e_2} & 0 \\ 0 & 0 & \frac{1}{e_3} \end{pmatrix} \cdot \begin{pmatrix} -\sin(\beta_1) & \cos(\beta_1) \\ -\sin(\beta_2) & \cos(\beta_2) \\ -\sin(\beta_3) & \cos(\beta_3) \end{pmatrix} \\ \begin{pmatrix} \frac{1}{r_1} & 0 & 0 \\ 0 & \frac{1}{r_2} & 0 \\ 0 & 0 & \frac{1}{r_3} \end{pmatrix} \cdot \begin{pmatrix} \cos(\beta_1) & \sin(\beta_1) \\ \cos(\beta_2) & \sin(\beta_2) \\ \cos(\beta_3) & \sin(\beta_3) \end{pmatrix} \end{pmatrix}, \quad (6.14)$$

$$\mathbf{S}_\theta(\mathbf{q}, \lambda) = \begin{pmatrix} \begin{pmatrix} \frac{1}{e_1} & 0 & 0 \\ 0 & \frac{1}{e_2} & 0 \\ 0 & 0 & \frac{1}{e_3} \end{pmatrix} \cdot \begin{pmatrix} -e_1 + x_{a_1} \cos(\beta_1) + y_{a_1} \sin(\beta_1) \\ -e_2 + x_{a_2} \cos(\beta_2) + y_{a_2} \sin(\beta_2) \\ -e_3 + x_{a_3} \cos(\beta_3) + y_{a_3} \sin(\beta_3) \end{pmatrix} \\ \begin{pmatrix} \frac{1}{r_1} & 0 & 0 \\ 0 & \frac{1}{r_2} & 0 \\ 0 & 0 & \frac{1}{r_3} \end{pmatrix} \cdot \begin{pmatrix} x_{a_1} \sin(\beta_1) - y_{a_1} \cos(\beta_1) \\ x_{a_2} \sin(\beta_2) - y_{a_2} \cos(\beta_2) \\ x_{a_3} \sin(\beta_3) - y_{a_3} \cos(\beta_3) \end{pmatrix} \end{pmatrix}. \quad (6.15)$$

Par un procédé de linéarisation équivalent à (6.11), nous obtenons les formules suivantes pour la gyrodométrie :

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k-1} + \begin{pmatrix} \left(\mathbf{S}_{xy}(\mathbf{q}_{k-1}, \lambda) \cdot \mathbf{R}(\hat{\theta}_{k-1}) \right)^\dagger & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 6} & 1 \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{q}_k - \mathbf{S}_\theta(\mathbf{q}_{k-1}, \lambda) \cdot \Delta \theta_k \\ \Delta \theta_k \end{pmatrix}, \quad (6.16)$$

où $\Delta \theta_k$ est la mesure du changement du cap fournie par le gyromètre laser entre les instants $k-1$ et k . L'utilisation d'un gyromètre précis et de faible dérive (dans notre cas, 45° par heure) va permettre de minimiser les erreurs de cap.

6.2.4 Identification paramétrique

Pour identifier le vecteur de paramètres λ , nous avons appliqué une technique d'identification stochastique utilisant des mesures fournies par le gyromètre laser. Nous avons appliqué un filtre de Kalman étendu pour identifier l'état du modèle :

$$\lambda_n = \lambda_{n-1}, \quad (6.17)$$

$$\Delta \theta_n = h_n(\lambda_n) + w_n. \quad (6.18)$$

avec

$$h_n(\lambda_n) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{J}^\dagger(\mathbf{q}_{n-1}, \lambda_n, \hat{\theta}_{n-1}) \cdot \Delta \mathbf{q}_n. \quad (6.19)$$

Le modèle de processus donné l'équation (6.17) considère que les paramètres sont constants dans le temps, *i.e.*, sans bruit de processus associé. Les mesures reliées à l'état λ_n par l'équation

(6.18) sont perturbées par un bruit w_n supposé gaussien de moyenne nulle et de variance Λ_{w_n} . Si nous disposons de N mesures $\Delta\theta_n$, les équations du filtre de Kalman pour la n -ième mise à jour sont :

$$\begin{aligned}\hat{\lambda}_n &= \hat{\lambda}_{n-1} + \mathbf{k}_n \cdot (\Delta\theta_n - h_n(\hat{\lambda}_{n-1})), \\ \mathbf{\Lambda}_{\hat{\lambda}_n} &= \left(\mathbf{I}_{12} - \mathbf{k}_n \cdot \nabla_{\hat{\lambda}_{n-1}}^T h_n \right) \cdot \mathbf{\Lambda}_{\hat{\lambda}_{n-1}},\end{aligned}\quad (6.20)$$

où le gain de Kalman $\mathbf{k}_n$ est donné par

$$\mathbf{k}_n = \mathbf{\Lambda}_{\hat{\lambda}_{n-1}} \cdot \nabla_{\hat{\lambda}_{n-1}}^T h_n \cdot \left(\nabla_{\hat{\lambda}_{n-1}} h_n \cdot \mathbf{\Lambda}_{\hat{\lambda}_{n-1}} \cdot \nabla_{\hat{\lambda}_{n-1}}^T h_n + \Lambda_{w_n} \right)^{-1}. \quad (6.21)$$

Afin de minimiser le risque d'inconsistance d'estimation en utilisant le filtre de Kalman étendu, w_n doit représenter non seulement l'incertitude associée à la mesure capteur $\Delta\theta_k$, mais aussi l'incertitude du modèle $h_n(\lambda_n)$ composé de variables dont l'incertitude est pour l'instant inconnue : $\mathbf{q}_{k-1}$ et $\Delta\mathbf{q}_k$. Ainsi, la solution choisie consiste à calculer Λ_{w_n} par la formule suivante

$$\Lambda_{w_n} = \sigma_{\Delta\theta_n}^2 + (\mu_n)^2 \quad (6.22)$$

où μ_n est un paramètre à régler. Cette formulation est équivalente à la forme linéarisée des moindres carrés récursifs de gain normalisé [148], où μ_n détermine la vitesse de convergence de l'estimateur. Néanmoins, la convergence de l'estimateur ne signifie pas obligatoirement convergence des estimées aux vraies valeurs, étant donné que nous utilisons une approximation linéaire du modèle (cf. section E.4).

Nous avons appliqué cet estimateur aux données acquises dans les expériences *Halle robotique - A* (annexe G.2), *Couloir robotique - A* (annexe G.4), *Premier étage extension - A* (annexe G.6) et *Rez-de-chaussée extension - A* (annexe G.8). Ces données correspondent au changement de cap $\Delta\theta_n$ mesuré par le gyromètre, et aux variables de configuration $\mathbf{q}_{n-1}$ et $\Delta\mathbf{q}_n$. $\hat{\theta}_{n-1}$ est intégré en utilisant les mesures du gyromètre, *i.e.*, $\hat{\theta}_n = \hat{\theta}_{n-1} + \Delta\theta_n$. L'initialisation de l'état est donnée par $\hat{\lambda}_0 = \lambda_0$, dont les composantes sont les mesures (6.1), avec une matrice de covariance $\mathbf{\Lambda}_{\hat{\lambda}_0} = (0,005\text{ m})^2 \cdot \mathbf{I}_{12}$. Cette initialisation de $\mathbf{\Lambda}_{\hat{\lambda}_0}$ est celle qui nous a donné les meilleurs résultats. Avec $\mu_n = 160 \cdot \sigma_{\Delta\theta_n}$, nous avons obtenu :

$$\hat{\lambda} : \begin{cases} \hat{r}_1 = 0,0988\text{ m} & \hat{e}_1 = 0,0443\text{ m} & \hat{x}_{a_1} = 0,3143\text{ m} & \hat{y}_{a_1} = 0,1916\text{ m} \\ \hat{r}_2 = 0,0989\text{ m} & \hat{e}_2 = 0,0424\text{ m} & \hat{x}_{a_2} = -0,3302\text{ m} & \hat{y}_{a_2} = 0,1926\text{ m} \\ \hat{r}_3 = 0,1012\text{ m} & \hat{e}_3 = 0,0446\text{ m} & \hat{x}_{a_3} = 0,0170\text{ m} & \hat{y}_{a_3} = -0,3816\text{ m} \end{cases}, \quad (6.23)$$

Nous présentons sur la Figure 6.2 des courbes de validation des paramètres estimés $\hat{\lambda}$, qui correspondent aux écarts de position et d'orientation des trajectoires obtenues par odométrie par rapport à celles obtenues par gyrodométrie. Nous avons utilisé pour la gyrodométrie les paramètres λ_0 , qui sont ceux utilisés pour superposer les images télémétriques de l'annexe G. Les données de validation sont différentes de celles utilisées par l'identification, et sont celles correspondant aux expériences *Couloir robotique - B* (section G.5) et *Premier étage extension - B* (section G.7). Nous présentons aussi les écarts obtenus avec l'odométrie en utilisant les paramètres mesurés λ_0 . Nous observons que les erreurs de positionnement sont beaucoup moins importants en utilisant les paramètres estimés.

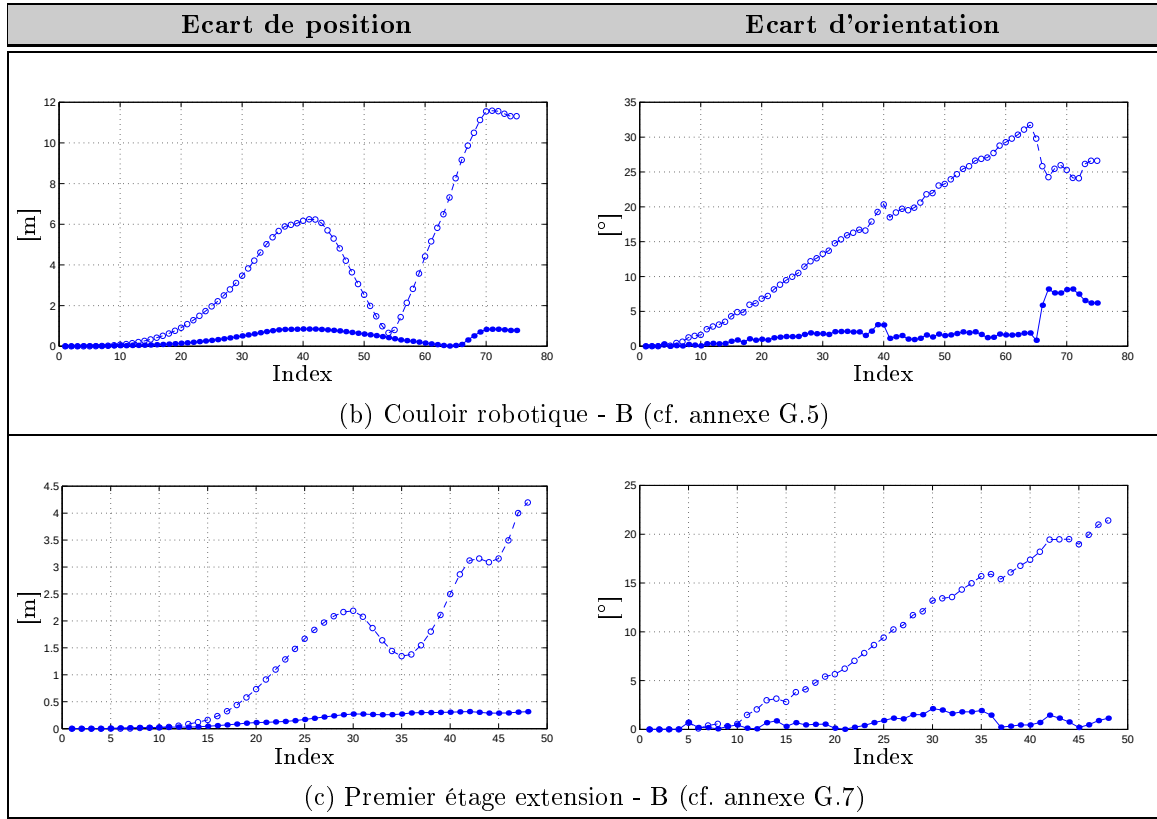


FIG. 6.2 – Validation de l'identification de paramètres géométriques : écarts de position $X \times Y$ et d'orientation θ de l'odométrie en utilisant les paramètres mesurés λ_0 (courbes en -o-) et les paramètres estimés $\hat{\lambda}$ (courbes en -●-).

6.2.5 Modèle de propagation de l'erreur

De façon générale, nous pouvons écrire les formules de mise à jour de l'estimée $\hat{\mathbf{z}}_k$ par odométrie (équation (6.11)) ou par gyrodométrie (équation (6.16)), sous la forme suivante :

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k-1} + \Delta \hat{\mathbf{z}}_k(\hat{\theta}_{k-1}, \mathbf{q}_{k-1}, \Delta \mathbf{q}_k, \Delta \theta_k, \hat{\lambda}) \quad (6.24)$$

$$= \mathbf{f}(\hat{\mathbf{z}}_{k-1}, \mathbf{q}_{k-1}, \Delta \mathbf{q}_k, \Delta \theta_k, \hat{\lambda}), \quad (6.25)$$

où le terme $\Delta \theta_k$ ne serait considéré que pour la gyrodométrie. Pour la propagation d'incertitude, nous considérons des erreurs supposées gaussiennes, de moyenne nulle, et de dispersions représentées par :

- $\Lambda_{\hat{\mathbf{z}}_{k-1}}$: la matrice de covariance de $\hat{\mathbf{z}}_{k-1}$, calculée lors de la mise à jour précédente de la position ;
- $\sigma_{\Delta \theta_k}^2$: la variance de $\Delta \theta_k$, donnée par le fabricant du capteur¹ ;
- $\sigma_{r_i}^2$: la variance de l' i -ième composante de $\mathbf{r}_k = \Delta \mathbf{q}_k - \mathbf{J}(\mathbf{q}_{k-1}, \hat{\lambda}, \hat{\theta}_{k-1}) \cdot (\hat{\mathbf{z}}_k - \hat{\mathbf{z}}_{k-1})$ ². En accord avec le théorème 6.1, $\mathbf{r}_k$ est le vecteur des résidus d'estimation dans l'espace $\Delta \mathbf{q}_k$.

¹Pour la gyrodométrie seulement.

²Pour l'odométrie seulement.

Si $\Delta \mathbf{q}_k$ n'est pas cohérent avec le modèle, ces résidus seront importants (par exemple, lors du glissement d'une roue). En supposant les composantes de $\mathbf{r}_k$ indépendantes, la variance empirique non-biaisée de chacune d'elles est donnée par

$$\sigma_{r_i}^2 = \frac{1}{5} \cdot \mathbf{r}_k^T \cdot \mathbf{r}_k. \quad (6.26)$$

- $\Lambda_{\hat{\lambda}}$: la matrice de covariance de $\hat{\lambda}$. Elle ne peut pas être celle déterminée par identification car, pour faire converger l'algorithme employé, nous avons injecté du bruit artificiel sur les mesures (cf. équation 6.22). Dans ces conditions, la matrice de covariance finale n'est pas la vraie covariance de $\hat{\lambda}$ (cf. section E.4). Ainsi, nous lui avons donné la valeur suivante

$$\Lambda_{\hat{\lambda}} = (0,03 \text{ m})^2 \cdot \mathbf{I}_{12}, \quad (6.27)$$

qui semble intuitivement valide. Elle correspond à un écart-type de 3 cm pour les composantes de $\hat{\lambda}$.

Si nous considérons les sources d'incertitude ci-dessus, la matrice de covariance $\Lambda_{\hat{\mathbf{z}}_k}$ est déterminée par propagation directe :

$$\begin{aligned} \Lambda_{\hat{\mathbf{z}}_k} = & (\nabla_{\hat{\mathbf{z}}_{k-1}} \mathbf{f}) \cdot \Lambda_{\hat{\mathbf{z}}_{k-1}} \cdot (\nabla_{\hat{\mathbf{z}}_{k-1}} \mathbf{f})^T + (\nabla_{\Delta \theta_k} \mathbf{f}) \cdot \sigma_{\Delta \theta_k}^2 \cdot (\nabla_{\Delta \theta_k} \mathbf{f})^T \\ & + (\nabla_{\mathbf{r}_k} \mathbf{f}) \cdot \mathbf{Diag}(\sigma_{r_1}^2, \dots, \sigma_{r_6}^2) \cdot (\nabla_{\mathbf{r}_k} \mathbf{f})^T + \mathbf{\Gamma} \cdot (\nabla_{\hat{\lambda}} \mathbf{f}) \cdot \Lambda_{\hat{\lambda}} \cdot (\nabla_{\hat{\lambda}} \mathbf{f})^T \cdot \mathbf{\Gamma}^T, \end{aligned} \quad (6.28)$$

où $\mathbf{\Gamma}$ est une matrice diagonale permettant d'agrandir artificiellement la portion d'incertitude due à $\hat{\lambda}$. En effet, si $\mathbf{\Gamma} = \mathbf{Diag}(\gamma_x, \gamma_y, \gamma_\theta)$, nous pouvons amplifier indépendamment les axes de l'incertitude sur x , y et θ . Ainsi, $\mathbf{\Gamma}$ doit être déterminé avec des données expérimentales, de sorte que $\Lambda_{\hat{\mathbf{z}}_k}$ soit une estimée consistante, voir pessimiste, de l'incertitude sur $\hat{\mathbf{z}}_k$. En robotique mobile, ce genre de réglage empirique de l'incertitude de l'estime est très utilisé [8, 55, 56]. Nous avons obtenu des résultats satisfaisants avec les composantes suivantes de $\mathbf{\Gamma}$:

$$\text{Pour l'odométrie} : \quad \gamma_x = 10, \quad \gamma_y = 10, \quad \gamma_\theta = 5, \quad (6.29)$$

$$\text{Pour la gyrodométrie} : \quad \gamma_x = 2, \quad \gamma_y = 2, \quad \gamma_\theta = 0, \quad (6.30)$$

$\gamma_\theta = 0$ pour la gyrodométrie signifie que l'incertitude de la mesure $\Delta \theta_k$ du gyromètre est déjà suffisante pour une consistance satisfaisante de $\hat{\theta}_k$.

Pour conclure, nous avons observé que c'est essentiellement l'erreur d'estimation du cap qui rend l'odométrie moins sûre sur le robot Omni. On peut considérer qu'elle n'est fiable que pour une très courte période de temps. La fiabilité de la gyrodométrie dépend de la précision du gyromètre employé. Cependant, les imperfections du sol restent des sources non contrôlables d'erreur pour ces techniques.

6.3 Localisation référencée environnement

6.3.1 Principe et difficultés

La localisation référencée environnement est basée sur la mise en correspondance de la carte locale $\mathcal{M}^R$ avec la carte globale $\mathcal{M}^G$. La carte locale étant acquise par le robot à la position Z ,

les structures de $\mathcal{M}^R$ sont en fait des observations locales des structures correspondantes de $\mathcal{M}^G$. Les correspondances sont déterminées par l'approche de mise en correspondance locale de cartes, décrite dans l'annexe D.5. Pour cela, une estimée *a priori* $\hat{Z}_{k|k-1} = \{\hat{\mathbf{z}}_{k|k-1}, \mathbf{\Lambda}_{\hat{\mathbf{z}}_{k|k-1}}\}$ de la position est fournie par l'estime.

Dans le processus de localisation, nous ne nous intéressons qu'aux primitives géométriques des structures en correspondance. Ainsi, soit

$$\mathcal{C}^L : \{C_n^L = \{\mathbf{l}_n^G, \mathbf{l}_n^R\} | n = 1, \dots, N_L\} \quad (6.31)$$

l'ensemble de N_L paires de droites en correspondance, où C_n^L est la n -ième correspondance entre une droite de paramètres $\mathbf{l}_n^G$ de la carte globale et une droite de paramètres $\mathbf{l}_n^R$ de la carte locale. De façon équivalente

$$\mathcal{C}^P : \{C_n^P = \{\mathbf{p}_n^G, \mathbf{p}_n^R\} | n = 1, \dots, N_P\} \quad (6.32)$$

est l'ensemble de N_P paires de points en correspondance, où C_n^P est la n -ième correspondance entre un point de paramètres $\mathbf{p}_n^G$ de la carte globale et un point de paramètres $\mathbf{p}_n^R$ de la carte locale.

Les primitives correspondantes sont reliées entre elles par des équations qui dépendent de la position du robot. Ainsi, à partir de ces équations, il est possible d'estimer, complètement ou partiellement, la position du robot. Les techniques d'estimation que nous présentons dans cette section sont basées sur le filtrage stochastique et sur l'optimisation non-linéaire. En plus de leurs propres limitations, ces techniques doivent faire face à des difficultés liées aux conditions d'utilisation, notamment :

- I. La présence de fausses correspondances, résultant de :
 - Une estime biaisée ;
 - Une incertitude trop importante sur $\hat{Z}_{k|k-1}$;
 - Une erreur de segmentation sur la carte locale ;
- II. La présence d'obstacles fixes ou mobiles considérés comme des structures cartographiques ;
- III. La présence de correspondances multiples. Cela arrive très souvent dans des environnements encombrés, quand l'incertitude sur $\hat{Z}_{k|k-1}$ est suffisamment importante.

6.3.2 Approches par filtrage stochastique

Les approches par filtrage stochastique sont les plus utilisées pour la localisation par mise en correspondance de cartes géométriques, car elles permettent de formuler le problème de manière élégante et générique en autorisant l'intégration de primitives géométriques de différents types. On doit alors estimer l'état d'un système stochastique non-linéaire, c'est à dire, la position du robot considérée comme une variable aléatoire de distribution gaussienne. Les correspondances sont utilisées pour construire les fonctions de mesure :

$$\mathbf{y}_k^R = \mathbf{h}(\mathbf{z}_k, \mathbf{y}_k^G) + \mathbf{w}_k, \quad (6.33)$$

où $\mathbf{y}_k^R$ est composé des correspondances locales, $\mathbf{y}_k^G$ est composé des correspondances globales et $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\mathbf{w}_k})$. $\mathbf{h}$ est une fonction non-linéaire, construite à partir des équations qui relient les correspondances :

- Primitives de type point (cf. équation C.8) :

$$\mathbf{p}_n^R = \mathbf{h}_P(\mathbf{z}, \mathbf{p}_n^G) = \mathbf{R}(\theta_k) \cdot (\mathbf{p}_n^G - \mathbf{t}). \quad (6.34)$$

- Primitives de type droite (cf. équation C.17) :

$$\mathbf{l}_n^R = \mathbf{h}_L(\mathbf{z}, \mathbf{l}_n^G) = \mathbf{l}_n^G - \begin{pmatrix} \beta^T(\alpha_n^G) \cdot \mathbf{t} \\ \theta_k \end{pmatrix}. \quad (6.35)$$

Dans ces équations, $\beta(\phi) = (\cos(\phi) \quad \sin(\phi))^T$ pour tout angle ϕ et $\mathbf{t} = (x(k), y(k))^T$.

L'estimation de l'état est alors résolue en utilisant le filtrage de Kalman. Nous présentons trois formulations qui diffèrent par la façon dont les mesures sont traitées.

6.3.2.1 FKE parallèle

En appliquant directement les équations du filtre de Kalman étendu (*FKE*, annexe E.4.1), l'estimée de position du robot peut être corrigée par l'ensemble des mesures de correspondance. On intègre alors en une seule fois toutes les mesures acquises à l'instant k , de sorte que $\mathbf{y}_k^R$, $\mathbf{y}_k^G$ et $\mathbf{h}(\hat{\mathbf{z}}_{k|k-1}, \mathbf{y}_k^G)$ sont des vecteurs de dimension $2(N_L + N_P)$ donnés par

$$\mathbf{y}_k^R = \begin{pmatrix} \mathbf{l}_1^R \\ \vdots \\ \mathbf{l}_{N_L}^R \\ \mathbf{p}_1^R \\ \vdots \\ \mathbf{p}_{N_P}^R \end{pmatrix}, \quad \mathbf{y}_k^G = \begin{pmatrix} \mathbf{l}_1^G \\ \vdots \\ \mathbf{l}_{N_L}^G \\ \mathbf{p}_1^G \\ \vdots \\ \mathbf{p}_{N_P}^G \end{pmatrix}, \quad \mathbf{h}(\hat{\mathbf{z}}_{k|k-1}, \mathbf{y}_k^G) = \begin{pmatrix} \mathbf{h}_L(\hat{\mathbf{z}}_{k|k-1}, \mathbf{l}_1^G) \\ \vdots \\ \mathbf{h}_L(\hat{\mathbf{z}}_{k|k-1}, \mathbf{l}_{N_L}^G) \\ \mathbf{h}_P(\hat{\mathbf{z}}_{k|k-1}, \mathbf{p}_1^G) \\ \vdots \\ \mathbf{h}_P(\hat{\mathbf{z}}_{k|k-1}, \mathbf{p}_{N_P}^G) \end{pmatrix}. \quad (6.36)$$

Cette formulation, connue sous le nom de *FKE* parallèle, a été initialement employée pour pouvoir considérer les inter-corrélations des composantes du terme d'innovation $\nu = \mathbf{y}_k^R - \mathbf{h}(\hat{\mathbf{z}}_{k|k-1}, \mathbf{y}_k^G)$ [40][105][139]. En effet, les correspondances de la carte locale présentent des inter-corrélations [105] qui sont approximées par les termes extra diagonaux de la matrice :

$$\mathbf{\Lambda}_\nu = \nabla_{\hat{\mathbf{z}}_{k|k-1}} \mathbf{h} \cdot \mathbf{\Lambda}_{\hat{\mathbf{z}}_{k|k-1}} \cdot \nabla_{\hat{\mathbf{z}}_{k|k-1}}^T \mathbf{h} + \mathbf{\Lambda}_{\mathbf{w}_k}, \quad (6.37)$$

L'estimation avec le *FKE* parallèle requiert l'inversion de la matrice $\mathbf{\Lambda}_\nu$, de taille proportionnelle au nombre de mesures. Si le nombre de correspondances disponibles à un moment donné est très important, cela se répercute sur le temps de calcul.

6.3.2.2 *FKE* séquentiel

L'approche *FKE* séquentiel (*FKES*) [8] [108][170] considère que les mesures sont indépendantes, ce qui induit une grande réduction du temps de calcul et de la mémoire nécessaire pour stocker les variables du filtre. Cela revient à appliquer les formules de mise à jour de l'équation (E.32) comme si seulement une seule correspondance était disponible à la fois. L'ordre d'utilisation des correspondances a beaucoup d'importance. En fait, ce sont les premières correspondances qui auront le plus d'impact sur la réduction de l'incertitude de l'estimée, représentée par la matrice $\mathbf{\Lambda}_{\hat{\mathbf{z}}_k}$. Ainsi, au fur et à mesure que les correspondances sont traitées par le filtre, $\mathbf{\Lambda}_{\hat{\mathbf{z}}_k}$ décroît et les dernières correspondances restantes ont peu d'influence sur l'estimée $\hat{\mathbf{z}}_k$. Ainsi, pour minimiser le risque de divergence de ce filtre, il vaut mieux appliquer en premier les correspondances les plus *fiables*. Dans nos travaux, nous avons choisi d'utiliser la méthodologie employée dans [170][178], qui consiste à intégrer les correspondances selon l'ordre croissant des distances de Mahalanobis.

6.3.2.3 *FKEI* séquentiel

Le filtre de Kalman étendu itératif (*FKEI*, annexe E.4.2) a été utilisé par Kleeman [122] pour la localisation à partir de mesures sonar. Ces mesures étant effectuées à des instants différents, elles sont intégrées de façon indépendante par le *FKEI*. Dans notre cas, les mesures de correspondance sont toutes obtenues à un instant précis, ce qui requiert un choix de l'ordre d'intégration de ces mesures. Nous avons choisi de procéder comme pour le *FKE* séquentiel : les mesures sont considérées comme non-corrélées et elles sont intégrées à chaque itération du *FKEI*, par ordre croissant des distances de Mahalanobis.

6.3.3 Discussion à propos des approches *FKE*

Comme nous l'avons vu, la localisation dynamique repose sur le concept prédiction-correction, couramment employé pour le suivi de cibles [21]. Cependant, si une erreur se produit lors de la correction à l'instant $k - 1$, cette erreur sera intégrée par l'estime et la prédiction à l'instant k sera biaisée. Ce biais va augmenter par la suite et conduire à un échec de la localisation.

Les approches de localisation par filtrage de Kalman ont des limitations qu'il faut bien connaître pour mieux les utiliser (cf. annexe E.4). Pour cette discussion, nous faisons référence au *FKEI*. Il a été montré que ce filtre est équivalent à l'application de la méthode itérative de Gauss-Newton pour maximiser $f(\mathbf{z}|\mathbf{y}_k)$, la fonction de vraisemblance de $\hat{\mathbf{x}}_k$ étant donné les mesures $\mathbf{y}_k$ [18]. Or, Gauss-Newton présente une convergence d'ordre deux si la prédiction $\hat{\mathbf{x}}_{k|k-1}$ est *suffisamment proche* de la vraie position $\mathbf{x}_k$ [150]. Cela signifie que, étant donné le degré de non-linéarité de $\mathbf{h}(\hat{\mathbf{z}}_{k|k-1}, \mathbf{y}_k^G)$, une seule itération ne suffirait pas pour assurer la convergence du *FKEI*. Le *FKE* étant équivalent au *FKEI* avec une itération [18], il ne présente pas de garantie de convergence à un mode de $f(\mathbf{z}|\mathbf{y}_k)$.

L'approche *FKEI* séquentiel est aussi très sensible à des maximums locaux de la fonction de vraisemblance. Si cette fonction est multimodale, ce qui est fréquent lorsque les correspondances impliquent de multiples hypothèses de positionnement, l'estimée $\hat{\mathbf{x}}_k$ a tendance à se rapprocher du

mode le plus proche de la prédiction $\hat{\mathbf{x}}_{k|k-1}$. Comme la moyenne d'une distribution multimodale peut ne pas correspondre aux modes, *FKEI* risque de présenter une estimation biaisée.

Un autre point important pour les approches séquentielles, est le critère de détermination de la séquence d'intégration des correspondances. Les solutions déjà proposées, utilisent comme critère soit la trace de la covariance *a posteriori* $\mathbf{A}_{\hat{\mathbf{x}}_k}$ après intégration [9], soit la distance de Mahalanobis [170], ou encore l'incertitude des mesures [8]. [108] n'utilise pas de critère particulier. De notre point de vue, il n'est pas évident d'établir un tel critère, car la *fiabilité* d'une correspondance ne doit pas être évaluée seulement localement. En fait, nous pensons que seule une analyse sur l'ensemble de correspondances doit permettre de déterminer celles qui sont aberrantes. Des critères individuels (ou locaux) sont très sensibles aux biais dus à des conditions d'observation différentes. Cela affecte principalement les critères reposant seulement dans l'incertitude des mesures. De plus, une estime mal réglée peut biaiser la prédiction $\hat{\mathbf{z}}_{k|k-1}$ et affecter ainsi directement les résidus de correspondance ainsi que leur distance de Mahalanobis.

6.3.4 Nouvelles approches par optimisation non-linéaire

Dans le domaine de la vision par ordinateur, la reconstruction 3-D à partir de primitives géométriques requiert l'utilisation d'estimateurs capables de fournir des estimées fiables de changement de repère qui relient des correspondances géométriques. Nous avons remarqué qu'il existe un grand nombre d'approches basées sur l'optimisation. Ces méthodes cherchent à minimiser une fonction de coût exprimée à partir des N correspondances et des paramètres $\mathbf{z}$ du changement de repère³ :

$$V = \sum_{n=1}^N \mu_n \cdot d^2(\mathbf{C}_n, \hat{\mathbf{z}}), \quad (6.38)$$

où $d^2(\cdot)$ est une mesure de distance et μ_n est un poids associé à la n -ième correspondance. Ces solutions sont en général restreintes à des correspondances de même type, par exemple entre points [208][230][92], segments de droite [228][120] ou droites infinies [221][160]. Sauf dans quelques cas [22][171][198], ces approches n'ont pas été utilisées pour la localisation de robots mobiles. Cela est peut être dû au fait qu'elles ne peuvent être appliquées qu'à des primitives de même type. Une méthode pour des correspondances de types différents est présenté dans [177]. Des correspondances de type point-point et de type droite-droite y sont utilisées dans un critère hétérogène composé de deux termes, un pour chaque type de primitive. Cependant, ce critère hétérogène est difficile à exploiter car il mélange de termes de dimensions différentes, comme par exemple la somme des écarts d'orientation (en degrés) et les écarts de position (en mètres). Ces termes doivent être pondérés différemment et le calcul des paramètres de pondération est délicat.

Nous proposons dans cette section de nouvelles approches pour la localisation de robots mobiles. Au départ, nous nous sommes inspirés du concept des C-moyennes floues (cf. section 4.3.4.4) et des C-moyennes possibilistes [65] pour proposer une approche d'estimation de position et de mise en correspondance simultanée, en utilisant seulement des droites [37]. Or, les primitives de

³En vision par ordinateur, $\mathbf{z}$ est constitué de trois angles et trois translations (cas 3-D)

type point sont des indices intéressants pour la localisation. Nous avons donc développé une version qui considère simultanément des correspondances de type point-point et de type droite-droite [31]. Grâce à un changement de représentation des primitives de type point et de type droite, il est possible d'utiliser un critère homogène (équation (6.38)). Cette représentation, intitulée *représentation unifiée*, est présentée dans la section suivante. Enfin, en utilisant des concepts courants de la vision par ordinateur et des statistiques robustes, nous avons proposé un estimateur optimal de type moindres carrés pondérés (*MCP*) [36] [33][35]. Cet algorithme fournit des estimées qui minimisent le critère d'optimisation de façon non itérative et optimale (minimum garanti). Plus récemment, nous avons développé des variantes du *MCP*, qui sont le *MCP* incomplet et un M-estimateur itératif. Les versions présentées diffèrent souvent par le calcul des poids μ_n , et ont pour objectif de réduire la sensibilité de l'estimateur à des correspondances incorrectes, et d'augmenter ainsi sa robustesse. Après le développement mathématique de la procédure de minimisation et de ses propriétés, nous présentons les différentes approches de la méthode.

6.3.4.1 Représentation unifiée de primitives géométriques

Pour représenter de manière unifiée les primitives géométriques 2-D⁴, nous utilisons un vecteur $\mathbf{u} = (u_x, u_y)^T$ avec :

- $\mathbf{u} = \mathbf{p}$ une primitive de type point, de paramètres $\mathbf{p}$;
- $\mathbf{u} = \rho \cdot \beta(\alpha)$ pour une primitive de type droite, de paramètres $\mathbf{l} = (\rho, \alpha)^T$. En fait, $\mathbf{u}$ contient les coordonnées cartésiennes du point sur la droite qui se trouve le plus proche de l'origine du repère de représentation.

Il faut exprimer les équations qui relient $\mathbf{u}^G$, primitive unifiée de la carte globale, à $\mathbf{u}^R$ sa correspondante dans la carte locale. Cette relation dépend du type de primitive et de la position $\mathbf{z}$. Pour un point, la relation est celle donnée par l'équation (6.34) :

$$\mathbf{u}^R = \mathbf{R}(\theta) \cdot (\mathbf{u}^G - \mathbf{t}) . \quad (6.39)$$

Pour une droite, nous utilisons l'équation qui exprime tout point $\mathbf{p}^G$ sur la droite $\mathbf{l}^G$ de paramètres unifiés $\mathbf{u}^G$ par

$$\mathbf{p}^G(k) = \mathbf{u}^G + k \cdot \mathbf{R}(\pi/2) \cdot \beta(\alpha^G), \quad (6.40)$$

où k est un scalaire. Remarquons que $\mathbf{u}^G = \rho^G \cdot \beta(\alpha^G)$ et $\mathbf{R}(\pi/2) \cdot \beta(\alpha^G)$ sont orthogonaux, *i.e.*, $\beta^T(\alpha^G) \cdot \mathbf{R}^T(\pi/2) \cdot \mathbf{u}^G = 0$, et que le pied de la droite $\mathbf{u}^G$ est le point $\mathbf{p}^G(k)$ le plus proche de l'origine (dans ce cas, avec $k = 0$). La transformation de coordonnées étant appliquée, tous les points $\mathbf{p}^G(k)$ se trouvant sur la droite ont des représentations locales reliées par l'équation (6.39), *i.e.*,

$$\mathbf{p}^R(k) = \mathbf{R}(\theta) \cdot (\mathbf{p}^G(k) - \mathbf{t}) = \mathbf{R}(\theta) \cdot (\mathbf{u}^G + k \cdot \mathbf{R}(\pi/2) \cdot \beta(\alpha^G) - \mathbf{t}) . \quad (6.41)$$

⁴Cette représentation ne sert qu'au processus d'estimation par optimisation. Il n'est pas destiné à remplacer les représentations précédentes des primitives des structures cartographiques.

Comme le pied de la droite locale formée par les points $\mathbf{p}^R(k)$ est donné par le point le plus proche de l'origine, la valeur de k correspondante à ce point vérifie

$$\begin{aligned} k^* &= \arg \min_k \|\mathbf{p}^R(k)\|^2, \\ &= \arg \min_k \|\mathbf{R}(\theta) \cdot (\mathbf{u}^G + k \cdot \mathbf{R}(\pi/2) \cdot \beta(\alpha^G) - \mathbf{t})\|^2, \\ &= \arg \min_k \|\mathbf{u}^G + k \cdot \mathbf{R}(\pi/2) \cdot \beta(\alpha^G) - \mathbf{t}\|^2, \end{aligned} \quad (6.42)$$

Nous pouvons vérifier que $k^* = \beta^T(\alpha^G) \cdot \mathbf{R}^T(\pi/2) \cdot \mathbf{t}$ est la solution cherchée. L'application de ce résultat à l'équation (6.41) donne

$$\mathbf{p}^R(k^*) = \mathbf{u}^R = \mathbf{R}(\theta) \cdot \left(\mathbf{u}^G - (\mathbf{I}_2 - \mathbf{R}(\pi/2) \cdot \beta(\alpha^G) \cdot \beta^T(\alpha^G) \cdot \mathbf{R}^T(\pi/2)) \cdot \mathbf{t} \right), \quad (6.43)$$

où

$$\mathbf{I}_2 - \mathbf{R}(\pi/2) \cdot \beta(\alpha^G) \cdot \beta^T(\alpha^G) \cdot \mathbf{R}^T(\pi/2) = \begin{pmatrix} \cos^2(\alpha^G) & \sin(\alpha^G) \cos(\alpha^G) \\ \sin(\alpha^G) \cos(\alpha^G) & \sin^2(\alpha^G) \end{pmatrix}. \quad (6.44)$$

La similarité entre les équations (6.39) et (6.43) nous permet d'exprimer la transformation de coordonnées sous une forme générale, donnée par

$$\mathbf{u}^R = \mathbf{R}(\theta) \cdot \{ \mathbf{u}^G - \mathbf{E}^G \cdot \mathbf{t} \}, \quad (6.45)$$

avec

$$\mathbf{E}^G = \begin{pmatrix} (a^G)^2 & c^G \\ c^G & (b^G)^2 \end{pmatrix}.$$

Les composantes de la matrice $\mathbf{E}^G$ dépendent du type de primitive : $a^G = b^G = 1$ et $c^G = 0$ pour une primitive de type point, et $a^G = \cos(\alpha^G)$, $b^G = \sin(\alpha^G)$ et $c^G = a^G \cdot b^G$ pour une primitive de type droite.

6.3.4.2 Estimation optimale au sens des moindres carrés pondérés

Dans cette section nous présentons l'algorithme permettant d'obtenir des estimées optimales de $\hat{\mathbf{t}} = (\hat{x}, \hat{y})^T$ et $\hat{\theta}$, étant donné les correspondances $\mathcal{C}^L$ et $\mathcal{C}^P$. Nous définissons

$$\mathcal{U} : \{U_n = \{\mathbf{u}_n^G, \mathbf{u}_n^R\} | n = 1, \dots, N_U\} \quad (6.46)$$

comme l'ensemble de $N_U = N_L + N_P$ correspondances unifiées, calculées à partir de $\mathcal{C}^L$ et $\mathcal{C}^P$. Notre critère d'optimisation dévient

$$J(\hat{\mathbf{z}}, \mathcal{U}) = \sum_{n=1}^{N_U} \mu_n \cdot d^2(\mathbf{u}_n^R, \mathbf{u}_n^G, \hat{\mathbf{z}}). \quad (6.47)$$

où la fonction de distance $d^2(\mathbf{u}_n^R, \mathbf{u}_n^G, \hat{\mathbf{z}})$ est donnée par

$$d^2(\mathbf{u}_n^R, \mathbf{u}_n^G, \hat{\mathbf{z}}) = \|\mathbf{r}_n\|_2^2. \quad (6.48)$$

Dans l'équation (6.48),

$$\mathbf{r}_n = \mathbf{u}_n^R - \mathbf{R}(\hat{\theta}) \cdot \{\mathbf{u}_n^G - \mathbf{E}_n^G \cdot \hat{\mathbf{t}}\} \quad (6.49)$$

est le n -ième résidu d'estimation. Les correspondances sont pondérées par des constantes réelles positives μ_n . $\mathbf{r}_n$ étant une fonction non-convexe de $\hat{\mathbf{z}}$, $J(\hat{\mathbf{z}}, \mathcal{U})$ peut avoir plusieurs minima. Ainsi, nous avons adopté la stratégie suivante pour minimiser cette fonction de coût : trouver tous les minima locaux $\hat{\mathbf{z}}^*$ de $J(\hat{\mathbf{z}}, \mathcal{U})$ et déterminer l'estimée $\hat{\mathbf{z}}$ comme étant le $\hat{\mathbf{z}}^*$ qui satisfait un critère de sélection. Ainsi, comme première condition, le minimas locaux de $J(\hat{\mathbf{z}}, \mathcal{U})$ satisfont à

$$\frac{\partial J(\hat{\mathbf{z}}, \mathcal{U})}{\partial \hat{x}} = \frac{\partial J(\hat{\mathbf{z}}, \mathcal{U})}{\partial \hat{y}} = \frac{\partial J(\hat{\mathbf{z}}, \mathcal{U})}{\partial \hat{\theta}} = 0, \quad (6.50)$$

ce qui nous permet d'obtenir

$$\mathbf{M} \cdot \hat{\mathbf{t}} - \mathbf{N} \cdot \beta(\hat{\theta}) = \mathbf{q}, \quad (6.51)$$

$$\beta^T(\hat{\theta}) \cdot (\mathbf{r} + \mathbf{R}(\pi/2) \cdot \mathbf{N}^T \cdot \hat{\mathbf{t}}) = 0, \quad (6.52)$$

avec

$$\mathbf{M} = \sum_{n=1}^{N_U} \mu_n \begin{pmatrix} (c_n^G)^2 + (a_n^G)^4 & c_n^G \cdot ((a_n^G)^2 + (b_n^G)^2) \\ c_n^G \cdot ((a_n^G)^2 + (b_n^G)^2) & (c_n^G)^2 + (b_n^G)^4 \end{pmatrix}, \quad (6.53)$$

$$\mathbf{N} = \sum_{n=1}^{N_U} \mu_n \begin{pmatrix} -(a_n^G)^2 \cdot u_{x,n}^R - (c_n^G) \cdot (u_{y,n}^R) & (a_n^G)^2 \cdot (u_{y,n}^R) - (c_n^G) \cdot (u_{x,n}^R) \\ -(c_n^G) \cdot (u_{x,n}^R) - (b_n^G)^2 \cdot (u_{y,n}^R) & (c_n^G) \cdot (u_{y,n}^R) - (b_n^G)^2 \cdot (u_{x,n}^R) \end{pmatrix}, \quad (6.54)$$

$$\mathbf{q} = \sum_{n=1}^{N_U} \mu_n \begin{pmatrix} (c_n^G) \cdot (u_{y,n}^G) + (a_n^G)^2 \cdot (u_{x,n}^G) \\ (c_n^G) \cdot (u_{x,n}^G) + (b_n^G)^2 \cdot (u_{y,n}^G) \end{pmatrix}, \quad (6.55)$$

$$\mathbf{r} = \sum_{n=1}^{N_U} \mu_n \begin{pmatrix} -(u_{x,n}^G) \cdot (u_{y,n}^R) + (u_{y,n}^G) \cdot (u_{x,n}^R) \\ -(u_{y,n}^G) \cdot (u_{y,n}^R) - (u_{x,n}^G) \cdot (u_{x,n}^R) \end{pmatrix}, \quad (6.56)$$

A partir des équations (6.51) et (6.52) nous obtenons

$$\beta^T(\hat{\theta}) \cdot \Phi \cdot \beta(\hat{\theta}) + \beta^T(\hat{\theta}) \cdot \zeta = 0, \quad (6.57)$$

où

$$\Phi = \begin{pmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{pmatrix} = \mathbf{R}(\pi/2) \cdot \mathbf{N}^T \cdot \mathbf{M}^{-1} \cdot \mathbf{N}, \quad (6.58)$$

$$\zeta = \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = \mathbf{r} + \mathbf{R}(\pi/2) \cdot \mathbf{N}^T \cdot \mathbf{M}^{-1} \cdot \mathbf{q}. \quad (6.59)$$

Comme $\mathbf{M}$ est une matrice symétrique, $\mathbf{N}^T \cdot \mathbf{M}^{-1} \cdot \mathbf{N}$ l'est aussi. Ainsi, étant donné la forme de la matrice $\mathbf{R}(\pi/2)$, nous avons $\Phi_{11} = -\Phi_{22}$. Ce résultat permet d'écrire l'équation (6.57) ainsi :

$$\Phi_{11} \cos(2\hat{\theta}) + \frac{(\Phi_{12} + \Phi_{21})}{2} \sin(2\hat{\theta}) + \zeta_1 \cos(\hat{\theta}) + \zeta_2 \sin(\hat{\theta}) = 0. \quad (6.60)$$

En fonction de Φ et ζ , l'équation (6.60) peut avoir jusqu'à quatre solutions $\hat{\theta}_i^*$, $i = 1, \dots, 4$, qui sont les composantes en rotation des minima $\hat{\mathbf{z}}_i^*$. En utilisant Maple^{©5}, les racines $\hat{\theta}_i^*$ sont calculées de la façon suivante :

$$\hat{\theta}_i^* = \arctan\left(-\frac{2\Phi_{11}p_i^2 - \Phi_{11} + \zeta_1 p_i}{(\Phi_{12} + \Phi_{21})p_i + \zeta_2}, p_i\right), \quad (6.61)$$

où p_i sont les racines réelles du polynôme

$$\alpha_4 p^4 + \alpha_3 p^3 + \alpha_2 p^2 + \alpha_1 p + \alpha_0 = 0, \quad (6.62)$$

avec les coefficients

$$\begin{aligned} \alpha_0 &= \Phi_{11}^2 - \zeta_2^2, & \alpha_1 &= -2(\zeta_1 \Phi_{11} + \zeta_2(\Phi_{12} + \Phi_{21})), \\ \alpha_2 &= -4\Phi_{11}^2 + \zeta_2^2 + \zeta_1^2 - (\Phi_{12} + \Phi_{21})^2, & \alpha_3 &= 4\zeta_1 \Phi_{11} + 2\zeta_2(\Phi_{12} + \Phi_{21}), \\ \alpha_4 &= 4\Phi_{11}^2 + (\Phi_{12} + \Phi_{21})^2. \end{aligned}$$

Dans notre implémentation, les racines de (6.62) ont été calculées en utilisant une approche basée sur les valeurs propres d'une matrice dont l'équation (6.62) est le polynôme caractéristique ([179], chapitre 9.5).

Les paramètres de position $\hat{\mathbf{t}}_i^*$ associés à chaque $\hat{\theta}_i^*$ sont déterminés à partir de l'équation (6.51) :

$$\hat{\mathbf{t}}_i^* = \mathbf{M}^{-1} \cdot (\mathbf{N} \cdot \beta(\hat{\theta}_i^*) + \mathbf{q}), \quad (6.63)$$

A ce stade, nous savons que le critère (6.47) peut avoir jusqu'à quatre solutions $\hat{\mathbf{z}}_i^*$. En effet, comme ces solutions satisfont l'équation (6.50), elles sont les minima et les maxima locaux du critère J . Ainsi, comme condition restrictive pour déterminer les minima locaux, nous gardons les solutions $\hat{\mathbf{z}}_i^*$ qui satisfont à

$$\frac{\partial^2 J(\hat{\mathbf{z}}, \mathcal{U})}{\partial(\hat{\theta}_i^*)^2} > 0. \quad (6.64)$$

Dans ce calcul, il faut considérer $\hat{\mathbf{t}}_i^*$ comme une fonction de $\hat{\theta}_i^*$ (cf. équation 6.63). Cette analyse nous montre que la fonction de coût peut avoir au maximum deux minima qui sont les candidats à la solution $\hat{\mathbf{z}}$. Si nous avons seulement un minimum, celui ci est utilisé comme estimée $\hat{\mathbf{z}}$. Dans le cas où nous avons deux minima, l'estimée $\hat{\mathbf{z}}$ est le minimum ayant la composante $\hat{\theta}_i^*$ la plus proche de la prédiction $\hat{\theta}_{k|k-1}$. Si on ne dispose pas de l'odométrie, un autre critère de sélection consiste à vérifier si les composantes $\hat{\mathbf{t}}_i^*$ des minima correspondent à une position à laquelle tous les demi-plans en correspondance de la carte globale sont visibles.

Estimation de l'incertitude L'estimation de l'incertitude sur $\hat{\mathbf{z}}$ se fait par propagation des diverses sources d'incertitude sur le processus d'estimation, reliées par le critère $J(\hat{\mathbf{z}}, \mathcal{U})$. Nous

⁵Logiciel de calcul symbolique produit par *Waterloo Maple, Inc.*

supposons que les résidus $\mathbf{r}_n$ sont indépendants et qu'ils suivent $\mathcal{N}(\mathbf{0}, \Lambda_{\mathbf{r}})$, dont la matrice de covariance est estimée par

$$\Lambda_{\mathbf{r}} = \frac{\sum_{n=1}^{N_U} \mu_n (\mathbf{r}_n \cdot \mathbf{r}_n^T)}{\sum_{n=1}^{N_U} \mu_n}.$$

Pour considérer les résidus d'estimation, l'incertitude de l'estimée prend en compte toutes les primitives en correspondance y compris celles qui ne se superposent pas parfaitement. Cela peut être dû au biais du capteur, ou à de fausses correspondances.

Les incertitudes des primitives en correspondance sont également considérées sous la forme de matrices de covariance $\Lambda_{\mathbf{u}_n^R}$ et $\Lambda_{\mathbf{u}_n^G}$, obtenues par propagation des covariances des primitives d'origine. En utilisant la technique proposée par Haralick [93], décrite dans l'annexe B.4.1.1, on obtient :

$$\Lambda_{\hat{\mathbf{z}}} = \left(\frac{\partial \mathbf{g}}{\partial \hat{\mathbf{z}}} \right)^{-1} \cdot (\Sigma_{\mathbf{r}} + \Sigma_{\mathbf{u}^G} + \Sigma_{\mathbf{u}^R}) \cdot \left(\frac{\partial \mathbf{g}}{\partial \hat{\mathbf{z}}} \right)^{-1}, \quad (6.65)$$

où $\mathbf{g} = \partial J(\hat{\mathbf{z}}, \mathcal{U}) / \partial \hat{\mathbf{z}}$ et

$$\Sigma_{\mathbf{u}^G} = \sum_{n=1}^{N_U} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}_n^G} \right) \cdot \Lambda_{\mathbf{u}_n^G} \cdot \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}_n^G} \right)^T, \quad \Sigma_{\mathbf{u}^R} = \sum_{n=1}^{N_U} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}_n^R} \right) \cdot \Lambda_{\mathbf{u}_n^R} \cdot \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}_n^R} \right)^T, \quad (6.66)$$

$$\Sigma_{\mathbf{r}} = \sum_{n=1}^{N_U} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{r}_n} \right) \cdot \Lambda_{\mathbf{r}} \cdot \left(\frac{\partial \mathbf{g}}{\partial \mathbf{r}_n} \right)^T. \quad (6.67)$$

$\partial \mathbf{g} / \partial \hat{\mathbf{z}}$ étant la hessienne de $J(\hat{\mathbf{z}}, \mathcal{U})$ par rapport à $\hat{\mathbf{z}}$, elle est forcément non-singulière sur les minima locaux du critère d'optimisation. En conséquence, elle est toujours inversible.

6.3.4.3 Propriétés et existence de solutions

Le processus d'estimation *MCP* présente les propriétés suivantes :

Propriété 6.1: *La minimisation du critère $J(\hat{\mathbf{z}}, \mathcal{U})$ est obtenue de façon non itérative, avec la garantie que $\hat{\mathbf{z}}$ soit le minimum global. Ainsi, l'estimateur présenté est optimal au sens des moindres carrés. Cela ne serait pas le cas si l'on avait utilisé des techniques itératives de minimisation, telles que le gradient descendant ou Gauss-Newton.* $\diamond$

Propriété 6.2: *Cet estimateur n'est pas récursif; il ne permet pas de traiter les correspondances une à une.* $\diamond$

Propriété 6.3: *La matrice $\mathbf{M}$ (équation (6.53)) joue un rôle équivalent à la matrice de persistance d'excitation des moindres carrés linéaires. Si cette matrice n'est pas inversible, il n'y a pas de solution complète au problème d'estimation. En effet, si $\mathbf{M}$ est de rang 1, il existe un nombre infini de solutions $\hat{\mathbf{t}}_i^*$ pour chaque candidat $\hat{\mathbf{t}}_i^*$.* $\diamond$

La propriété 6.3 établit que, le processus *MCP* ne peut fournir une estimation complète (i.e., position et orientation) que si la matrice $\mathbf{M}$ est inversible. En fait, les composantes de cette matrice sont calculées en fonction seulement des correspondances globales⁶. Cela veut dire que, si la

⁶i.e., les primitives de la carte globale ayant de correspondance dans la carte locale.

configuration des correspondances globales est susceptible de permettre une solution, le rang de $\mathbf{M}$ sera complet. L'influence des correspondances locales⁷ ne joue que lors du calcul des matrices Φ et ζ , qui vont déterminer les coefficients du polynôme (6.62).

Soit $\Xi : \{\xi(\hat{\mathbf{z}}) = 0\}$ un sous-espace de $\mathbb{R}^3$, qui représente un ensemble de solutions au problème d'estimation par minimisation de $J(\hat{\mathbf{z}}, \mathcal{U})$. Les exemples suivants illustrent un cas où Ξ correspond à une solution incomplète, et un cas de solution indéterminée :

Exemple 6.1 (Solution incomplète): *Considérons le cas illustré sur la Figure 6.3(a), où le robot se trouve dans un couloir, et seuls les deux murs de ce couloir sont représentés sur la carte locale $\mathcal{M}^R$ sous forme de demi-plans D_1^R et D_2^R . Ces murs ont comme correspondants dans la carte globale $\mathcal{M}^G$ les demi-plans D_1^G et D_2^G . Supposons que D_1^R et D_2^R et D_1^G et D_2^G soient parallèles. Cette configuration ne permet pas une estimation unique de la position du robot. En effet, n'importe quelles estimées $\hat{x}$ et $\hat{y}$ dans un sous-espace affine*

$$\Xi : \{a\hat{x} + b\hat{y} + c = 0\},$$

sont des solutions qui minimisent le critère $J(\hat{\mathbf{z}}, \mathcal{U})$. Pour ce cas, il n'est possible de déterminer qu'une estimée $\hat{\theta}$ du cap du robot, celle qui permet le meilleur alignement des demi-plans. $\diamond$

Exemple 6.2 (Solution indéterminée): *Soit le cas illustré sur la Figure 6.3(b), où le robot se trouve dans une configuration telle que seule une arête de l'environnement est présente sur la carte locale $\mathcal{M}^R$. Cette arête A_1^R a une correspondance A_1^G dans la carte globale $\mathcal{M}^G$. Pour ce cas, il n'existe pas de solution unique pour l'estimation de position $(\hat{x}, \hat{y})$, ni pour celle du cap $\hat{\theta}$. En fait, n'importe quel $\hat{\theta}$ aurait une estimée $(\hat{x}, \hat{y})$ respectant le sous-espace circulaire*

$$\Xi : \{(\hat{x} - x)^2 + (\hat{y} - y)^2 - r^2 = 0\},$$

et minimisant le critère $J(\hat{\mathbf{z}}, \mathcal{U})$. $\diamond$

6.3.4.4 Estimation étendue à des cas incomplets

Un des avantages du *FKE* séquentiel pour la localisation est que, à mesure que chaque correspondance est intégrée, elle contribue à réduire l'incertitude d'estimation sur $\hat{\mathbf{z}}$ suivant les axes les plus concernés par les mesures. Lorsque ce filtre est appliqué au cas de l'exemple 6.1, la matrice de covariance de prédiction verra son incertitude réduite suivant l'axe orthogonal au sous-espace $a\hat{x} + b\hat{y} + c = 0$ (Figure 6.3(a)). Par contre, la prédiction ne sera pas modifiée dans la direction de ce sous-espace. Pour cela, lorsqu'un robot se déplace dans un couloir en se recalant par rapport aux murs, l'incertitude de positionnement augmente sans cesse dans la direction du couloir.

Pour pouvoir tirer partie de notre procédure d'estimation optimale (section 6.3.4.2) dans des cas de solution incomplète, nous avons proposé les modifications suivantes :

⁷i.e., les primitives de la carte locale ayant de correspondance dans la carte globale.

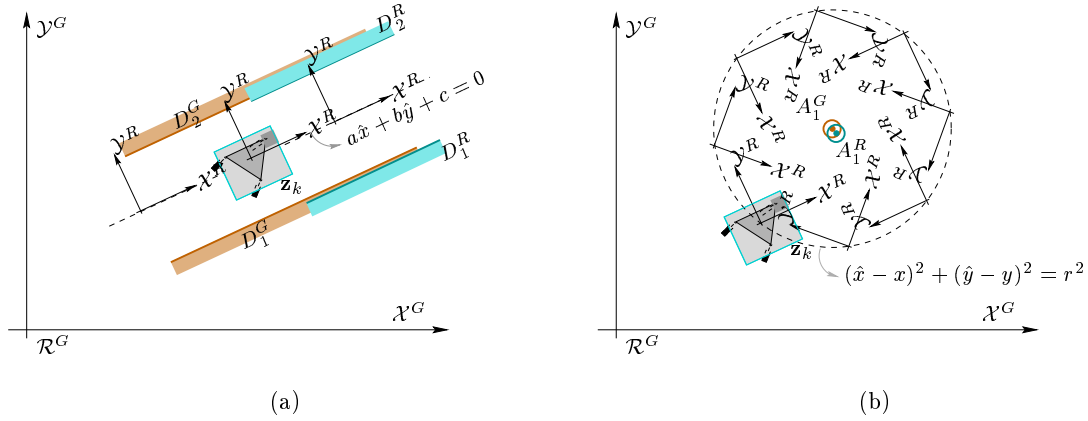


FIG. 6.3 – Cas de données incomplètes

I. *Inversion de $\mathbf{M}$ par décomposition en valeurs singulières* : Si nous écrivons $\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, alors $\mathbf{M}^{-1} = \mathbf{V}\mathbf{S}'\mathbf{U}^T$, où les composantes de la diagonale de $\mathbf{S}'$ sont égales à l'inverse des composantes de $\mathbf{S}$. Si une composante de $\mathbf{S}$ est nulle, ou inférieure à la précision numérique relative de la machine, alors la composante correspondante de $\mathbf{S}'$ est aussi nulle. Ainsi, la solution en position $\hat{\mathbf{t}}_i^*$ correspondant à la rotation $\hat{\theta}_i^*$ est donnée par :

$$\hat{\mathbf{t}}_i^* = \mathbf{V} \cdot \mathbf{S}' \cdot \mathbf{U}^T \cdot (\mathbf{N} \cdot \beta(\hat{\theta}_i^*) + \mathbf{q}). \quad (6.68)$$

Si $\mathbf{M}$ est de rang complet, l'équation (6.68) donne l'unique solution pour $\hat{\mathbf{t}}_i^*$. Mais, si $\mathbf{M}$ est de rang 1, il existe une valeur singulière (*i.e.*, une composante de la diagonale de $\mathbf{S}$) qui est nulle. Dans ce cas, nous avons un nombre infini de solutions pour $\hat{\mathbf{t}}_i^*$, dont l'équation (6.68) nous donne celle de norme minimale. Pour ce problème d'estimation, cette solution est celle qui place le robot le plus près de l'origine du repère global. Toute autre solution est donnée par ([179], chapitre 2.6) :

$$\hat{\mathbf{t}}_i^* = \mathbf{V} \cdot \mathbf{S}' \cdot \mathbf{U}^T \cdot (\mathbf{N} \cdot \beta(\hat{\theta}_i^*) + \mathbf{q}) + \lambda \cdot \mathbf{v} \quad (6.69)$$

où $\mathbf{v}$ est la colonne de $\mathbf{V}$ correspondant à la valeur singulière nulle de $\mathbf{S}$, et $\lambda \in \mathbb{R}$;

II. *Fusion avec la prédiction $\hat{\mathbf{z}}_{k|k-1}$* : Pour des cas de données incomplètes, les composantes de la covariance $\mathbf{\Lambda}_{\hat{\mathbf{z}}}$ (équation 6.65) qui sont associées aux variables indéterminées, tendent vers l'infini. Cela pose des problèmes d'ordre numérique lors de la mise en correspondance dans les cycles suivants de localisation. La prédiction $\hat{\mathbf{z}}_{k|k-1}$ apporte des informations complémentaires, qui peuvent être exploitées. Ainsi, l'estimée $\hat{\mathbf{z}}$ et la prédiction $\hat{\mathbf{z}}_{k|k-1}$ sont fusionnés en utilisant le filtre de Kalman linéaire :

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k|k-1} + \mathbf{K} \cdot (\hat{\mathbf{z}} - \hat{\mathbf{z}}_{k|k-1}), \quad \mathbf{\Lambda}_{\hat{\mathbf{z}}_k} = (\mathbf{I}_3 - \mathbf{K}) \cdot \mathbf{\Lambda}_{\hat{\mathbf{z}}_{k|k-1}}, \quad (6.70)$$

$$\mathbf{K} = \mathbf{I}_3 - \left(\mathbf{\Lambda}_{\hat{\mathbf{z}}_{k|k-1}}^{-1} + \mathbf{\Lambda}_{\hat{\mathbf{z}}}^{-1} \right)^{-1} \cdot \mathbf{\Lambda}_{\hat{\mathbf{z}}_{k|k-1}}^{-1}. \quad (6.71)$$

Remarquons que même si ces équations diffèrent de celles présentées dans l'annexe E.3.1, elles sont équivalentes, sauf qu'ici nous calculons le gain de Kalman en utilisant seulement les matrices d'information $\mathbf{\Lambda}_{\hat{\mathbf{z}}_{k|k-1}}^{-1}$ et $\mathbf{\Lambda}_{\hat{\mathbf{z}}}^{-1}$. En effet, cette formulation est adaptée lorsque $\mathbf{\Lambda}_{\hat{\mathbf{z}}}$ a des composantes importantes (dues à des cas incomplets). La matrice d'information $\mathbf{\Lambda}_{\hat{\mathbf{z}}}^{-1}$ doit être calculée par :

$$\mathbf{\Lambda}_{\hat{\mathbf{z}}}^{-1} = \left(\frac{\partial \mathbf{g}}{\partial \hat{\mathbf{z}}} \right) \cdot (\mathbf{\Sigma}_{\mathbf{r}} + \mathbf{\Sigma}_{\mathbf{u}^G} + \mathbf{\Sigma}_{\mathbf{u}^R})^{-1} \cdot \left(\frac{\partial \mathbf{g}}{\partial \hat{\mathbf{z}}} \right)^T, \quad (6.72)$$

où l'inversion est obtenue par décomposition en valeurs singulières.

Remarque 6.1: *Pour des cas de solution incomplète, cette nouvelle formulation présente une garantie de convergence dans le sous-espace des solutions, où seuls les axes de $\mathbf{\Lambda}_{\hat{\mathbf{z}}_{k|k-1}}$ associés à cette solution sont affectés par la procédure de fusion. Lorsque les solutions sont complètes, elles sont équivalentes à celles données par la formulation initiale (section 6.3.4.2), et le résultat obtenu est fusionné avec la prédiction $\hat{\mathbf{z}}_{k|k-1}$ en utilisant le filtre de Kalman linéaire. Pour des cas de solution indéterminée (cf. exemple 6.2), aucune estimation n'est possible.* $\diamond$

Les cas de solution indéterminée sont détectés si au moins une des conditions suivantes est vérifiée :

- I. Le rang de $\mathbf{M}$ est zéro. Cela n'est possible que si tous les poids μ_n sont nuls ;
- II. Tous les coefficients du polynôme (6.62) sont proches de zéro.

Nous remarquons que les propriétés 6.1 et 6.2 (section 6.3.4.3) sont encore valables pour cette version étendue. La fusion utilisant le filtre de Kalman linéaire hérite de tous les avantages de cette méthode (cf. section E.3.1), lorsqu'elle est appliquée à des modèles stochastiques connus. Ainsi, si l'estimée $\hat{\mathbf{z}}$ et la prédiction $\hat{\mathbf{z}}_{k|k-1}$ sont consistantes ou pessimistes, le résultat de la fusion sera consistant ou pessimiste (cf. théorème E.1). Cependant, si ces variables sont trop optimistes ou inconsistantes, on ne peut pas garantir que l'estimée $\hat{\mathbf{z}}_{k|k-1}$ soit consistante ou pessimiste.

6.3.4.5 Formulations MCP pour la localisation

Nous présentons dans cette section différentes formulations de la procédure d'estimation optimale MCP, étendue aux cas incomplets (MCP incomplet). Elles diffèrent souvent pour la façon dont les mesures de correspondance μ_n sont calculées.

Estimateurs MCP incomplet Nous présentons ici deux formulations qui réalisent l'estimation de position par la procédure MCP incomplet. Les poids associés aux correspondances peuvent être calculés par une des fonctionnelles suivantes :

- MCP incomplet (μ -Mahalanobis) :

$$\mu_n = \exp \left(-\frac{d_n}{\eta} \right), \quad (6.73)$$

où d_n est la distance de Mahalanobis entre les primitives géométriques de la n -ième correspondance, calculée comme pour la mise en correspondance (annexe D.5). Avec $\eta = -\frac{Q_2(0,95)}{\ln(0,1)}$, nous avons $0,1 \leq \mu_n \leq 1$, où $\mu_n = 0,1$ est associé à des correspondances qui sont à la limite du seuil $Q_2(0,95)$ (cf. annexe B.5) ;

- *MCP* incomplet (μ -Médiane) :

$$\mu_n = \exp \left(-\frac{\|\mathbf{r}_n\|}{\eta} \right), \quad (6.74)$$

où $\eta = -\frac{2,6}{\ln(0,1)} \cdot MED(\|\mathbf{r}_1\|, \dots, \|\mathbf{r}_{N_U}\|)$ et MED représente la médiane, qui est une estimation robuste de la moyenne. Cette fonction de pondération permet d'écarter toute correspondance ayant de résidus aberrants.

M-estimateur Les M-estimateurs sont beaucoup utilisés dans des problèmes d'estimation paramétrique en présence de mesures aberrantes [97][152], notamment en recherche en vision par ordinateur [20]. Citons par exemple des applications à l'estimation de position à partir de points 3-D [92], à l'extraction de primitives géométriques dans des images [80] [64], et au *clustering* [65][112]. Appliqué à l'estimation de position, un M-estimateur⁸ cherche à résoudre le problème par minimisation d'une fonction de coût de la forme [100]

$$\sum_n^{N_U} \rho(r_n^*, \hat{\mathbf{z}}) \quad (6.75)$$

où $\rho(\cdot)$ est fonction d'une mesure de distance mise à l'échelle r_n^* (de fonction équivalente aux résidus $\mathbf{r}_n$) étant donné l'estimée $\hat{\mathbf{z}}$. Si $\rho(\cdot) = \|r_n^*\|^2$, avec $r_n^* = \sqrt{\mathbf{r}_n^T \cdot \mathbf{r}_n}$ nous sommes dans le cas des moindres carrés classiques non-pondérés, qui sont des estimateurs non robustes. En effet, pour de tels estimateurs, un seul point aberrant, qui ne respecte pas le modèle, peut fausser l'estimation. Les M-estimateurs sont sensés "tolérer" des points aberrants, et présenter une estimée qui est peu sensible à ces points. Il existe différentes formes de M-estimateurs, qui se caractérisent par le choix sur la fonction $\rho(\cdot)$.

En général, la fonction de coût (6.75) ne conduit pas à une solution analytique, et sa minimisation s'effectue par des techniques itératives de recherche de minimum. Une autre approche de minimisation de (6.75) consiste à formuler le problème en utilisant les moindres carrés pondérés itératif [92][13]. Cela veut dire que l'on exécute la procédure d'estimation *MCP* incomplet à chaque itération de l'algorithme, et que les poids sont calculés en utilisant une fonction de pondération robuste. Etant donné le grand nombre de fonctions de pondération robuste [20], nous proposons deux formulations de M-estimateurs utilisant les fonctions les plus connues :

- M-estimateur (μ -Tukey) : A chaque itération, les poids des correspondances sont actualisés par la fonction *Tukey's biweight*, donnée par [92] :

$$\mu_n = \begin{cases} \left(1 - \left(\frac{\|\mathbf{r}_n\|}{c \cdot s} \right)^2 \right)^2, & \text{si } \|\mathbf{r}_n\| \leq c \cdot s \\ 0 & \text{sinon.} \end{cases}, \quad (6.76)$$

⁸Le 'M' vient de *maximum likelihood* [100].

où s est un estimateur d'échelle, et c est une constante qui commence à la valeur 12 pour la première itération, et décroît de façon linéaire à 9 pour la dernière itération (voir par exemple [92]) ;

- M-estimateur (μ -Huber) : A chaque itération, les poids des correspondances sont actualisés par la fonction de Huber [97] :

$$\mu_n = \min(1, \frac{c \cdot s}{\|\mathbf{r}_n\|}), \quad (6.77)$$

où $c = 1,345$ est une valeur optimale pour distributions gaussiennes [13] et s est toujours un estimateur d'échelle.

Pour ces estimateurs, s est la médiane des déviations absolues :

$$s = MED(\|\mathbf{r}_1\| - \bar{r}, \dots, \|\mathbf{r}_{N_U}\| - \bar{r}), \quad (6.78)$$

où $\bar{r} = MED(\|\mathbf{r}_1\|, \dots, \|\mathbf{r}_{N_U}\|)$. Bien entendu, à chaque itération les résidus sont calculés en utilisant l'estimée de la dernière itération. Pour la première itération, les résidus sont déterminés avec la prédiction $\hat{\mathbf{z}}_{k|k-1}$ donnée par l'estime.

Pour qu'un M-estimateur soit efficace, il faut prévoir un nombre important d'itérations. Pour ces formulations nous l'avons implanté avec 50 itérations. On remarque que la formulation *MCP* incomplet (μ -Médiane), de la section précédente, utilise elle aussi une estimation robuste de la moyenne, qui est la médiane. Mais, avec une seule itération, dans certains cas les correspondances aberrantes peuvent ne pas être totalement écartées (*i.e.*, $\mu_n \rightarrow 0$). Ce n'est qu'avec des itérations consécutives que l'on réduit les poids donnés à ces correspondances, et que l'on obtient une estimation plus robuste en leur présence. Cependant, si la proportion de mesures aberrantes parmi les N_U correspondances est très importante, l'estimateur ne pourra pas écarter les mauvaises correspondances. Même dans ces cas, il reste encore plus robuste qu'un moindre carré classique qui considère également toutes les mesures. Pour les cas des approches par filtrage de Kalman étendu, même si les correspondances sont intrinsèquement pondérées par l'inverse de la covariance de l'innovation (voir calcul du gain dans l'équation (E.33)), on a encore un estimateur non-robuste. Cela est dû au fait que cette matrice de covariance ne prend en compte que les covariances des mesures et de la prédiction, et ne dépend pas de l'écart entre les correspondances. C'est en effet cet écart qui mesure le degré d'aberration d'une correspondance par rapport aux autres.

6.3.5 Analyse de complexité

Dans cette section, nous présentons la complexité des approches de localisation en fonction du nombre $N = N_L + N_P$ de structures en correspondance. Cette complexité est évaluée en utilisant des techniques classiques. Nous utilisons la notation O pour représenter la complexité de calcul des algorithmes. Par exemple, $O(N^4)$ veut dire que le temps de calcul est donné par un polynôme au degré quatre en N .

Ainsi, nous pouvons vérifier que l'algorithme *FKE* parallèle présente une complexité $O(N^3)$, à cause de l'inversion de la matrice (6.37) lors de la mise à jour [179]. Par contre, comme l'approche *FKE* séquentiel intègre chacune des correspondances séparément, sa complexité est $O(N)$. Il en

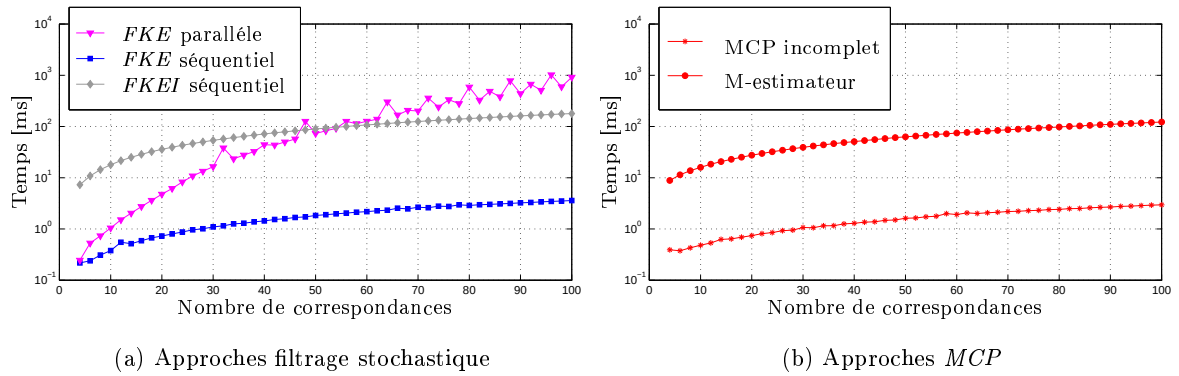


FIG. 6.4 – Temps de calcul des différentes approches de localisation.

est de même pour l'approche *FKEI* séquentiel, si le nombre d'itérations est fixé à l'avance. Le processus de calcul du *MCP* incomplet présente une complexité linéaire $O(N)$. Dans cet algorithme, la dimension de la matrice $\mathbf{M}$, et l'ordre du polynôme (6.62) ne dépendent pas de N . Les seules opérations qui dépendent du nombre de correspondances sont le calcul des matrices des équations (6.53)-(6.56), et des poids de correspondance par les équations (6.73) ou (6.74), qui se font en $O(N)$. Les M-estimateurs de la section 6.3.4.5 exécutent le *MCP* incomplet de façon itérative. Pour cela, ils ont aussi la même complexité de ce dernier, i.e., $O(N)$.

L'ordre de complexité ne permettant de juger que du taux de croissance du temps de calcul, il nous faut une idée approximative du temps d'exécution. A des fins de comparaison, nous avons réalisé une simulation où le temps de calcul a été mesuré, pour ces différentes approches avec des correspondances générées de façon aléatoire. Nous avons ainsi, sur les Figures 6.4(a) et 6.4(b), les courbes du temps d'exécution mesuré en fonction du nombre de correspondances (droites et points). Pour une meilleure visualisation, le temps est mis en échelle logarithmique. On remarque que les approches de *FKE* séquentiel et *MCP* incomplet sont celles qui s'exécutent le plus vite, et présentent à peu près le même temps de calcul. Les approches itératives *FKEI* séquentiel et le M-estimateur, qui réalisent 50 itérations, présentent des courbes décalées par rapport à celles des approches *FKE* séquentiel et *MCP* incomplet. Bien entendu, ce décalage étant constant dans l'échelle de logarithmique, cela signifie que les approches itératives ont un temps de calcul multiple des approches non itératives. Enfin, la localisation en utilisant le *FKE* parallèle est la technique la plus coûteuse en temps de calcul pour plus de 50 correspondances. Ce grand nombre de correspondances se rencontre seulement dans des environnements très encombrés.

Nous remarquons que, dans un système de navigation, les tâches de localisation ont en général des priorités moins hautes que celles associées à la commande et à l'acquisition de données. Cela signifie que ces temps d'exécution peuvent être encore plus importants.

6.4 Etudes en simulation

Nous ne présentons pas de simulation comparative des approches de localisation. En effet, les simulations ne permettent pas de prendre en considération la grande diversité des difficultés rencontrées en pratique. Nous présentons une analyse de cas qui permet de mieux comprendre le fonctionnement du processus d'estimation *MCP* incomplet pour la localisation. Nous rappelons que ce processus est le noyau des méthodes non-itératives et des M-estimateurs de la section 6.3.4.5.

Les simulations effectuées diffèrent en fonction des primitives géométriques des structures en correspondance, obtenues toujours avec le robot à la position absolue $\mathbf{z} = (2, 3, 120^\circ)^T$. Les primitives capturées dans la carte locale sont perturbées indépendamment par des bruits gaussiens d'écart-type σ_ε . Pour les droites, $\sigma_\varepsilon = 3 \text{ cm}$ pour le paramètre ρ , et $\sigma_\varepsilon = 0,5^\circ$ pour le paramètre α . Les coordonnées des points sont perturbées par des bruits d'écart-type $\sigma_\varepsilon = 3 \text{ cm}$. Nous ne nous intéressons pas à la qualité de l'estimation fournie par l'algorithme *MCP* incomplet, mais par les minimums $\hat{\mathbf{z}}_i^*$ calculés.

Nous considérons sept cas de figure illustrés sur les Figures 6.5 et 6.6. Sur ces figures, nous avons à gauche la configuration environnement avec les structures en correspondance auxquelles des numéros sont associés. Les solutions $\hat{\mathbf{z}}_i^*$ calculées pour $\mathbf{z}$, le rang de la matrice $\mathbf{M}$ et la somme des valeurs absolues des coefficients du polynôme (6.62) sont indiqués. Les courbes de droite représentent la fonction de coût $J(\hat{\mathbf{z}}, \mathcal{U})$ en fonction de $\hat{\theta}$. Ces sept cas sont analysés ci-dessous :

- *Deux droites et un point I (Fig. 6.5(a)).* Ce problème, présenté dans [35], ne peut pas être complètement résolu par des techniques d'optimisation appliquées à des correspondances de même type. Avec l'estimateur *MCP* incomplet, comme avec le simple *MCP*, il n'existe qu'une seule solution, comme nous pouvons le voir sur la figure ;
- *Deux droites et un point II (Fig. 6.5(b)).* Si nous plaçons la primitive point à l'intersection des deux droites, nous obtenons un cas avec deux solutions candidates $\hat{\mathbf{z}}_1^*$ et $\hat{\mathbf{z}}_2^*$, illustré sur la Figure 6.5(b). En effet, si l'on considère les droites infinies, la symétrie de la configuration des primitives permet les deux solutions présentées ;
- *Deux droites non parallèles (Fig. 6.5(c)).* Dans cette configuration, nous avons une symétrie qui est équivalente à celle de la Figure 6.5(b). Ainsi, il existe deux solutions candidates $\hat{\mathbf{z}}_1^*$ et $\hat{\mathbf{z}}_2^*$;
- *Deux droites parallèles (Fig. 6.6(a)).* Ceci est un cas incomplet, où nous pouvons déterminer seulement une rotation candidate $\hat{\theta}_1^*$. $\mathbf{M}$ n'étant pas de rang complet, il existe un nombre infini de solutions $\hat{\mathbf{t}}_1^*$. L'estimée représentée sur la Figure 6.6(a) utilise comme composante $\hat{\mathbf{t}}_1^*$ la solution de norme minimale, calculée par l'équation (6.68). Les autres solutions sont représentées dans un ton plus clair. La matrice d'information obtenue est

$$\mathbf{\Lambda}_{\hat{\mathbf{z}}_1^*}^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1,070 & 0,021 \\ 0 & 0,021 & 2,262 \end{bmatrix} \cdot 10^3, \quad (6.79)$$

ce qui indique que seules la position $\hat{y}$ et le cap $\hat{\theta}$ sont complètement déterminés. Ainsi, lors de la fusion avec la prédiction $\hat{\mathbf{z}}_{k|k-1}$, l'estimée suivant $\hat{x}$ est celle de l'estime ;

- *Une droite (Fig. 6.6(b)).* Ceci est encore un cas incomplet, mais qui présente deux rotations candidates $\hat{\theta}_1^*$ et $\hat{\theta}_2^*$, et, étant donné que $\mathbf{M}$ n'est pas de rang complet, un nombre infini de solutions $\hat{\mathbf{t}}_1^*$ et $\hat{\mathbf{t}}_2^*$. Les estimées sont représentées sur la Figure 6.6(b). Les matrices d'information obtenues sont identiques :

$$\Lambda_{\hat{\mathbf{z}}_1^*}^{-1} = \Lambda_{\hat{\mathbf{z}}_2^*}^{-1} = \begin{bmatrix} 1,942 & 2,649 & 0 \\ 2,649 & 3,612 & 0 \\ 0 & 0 & 9,140 \end{bmatrix} \cdot 10^2, \quad (6.80)$$

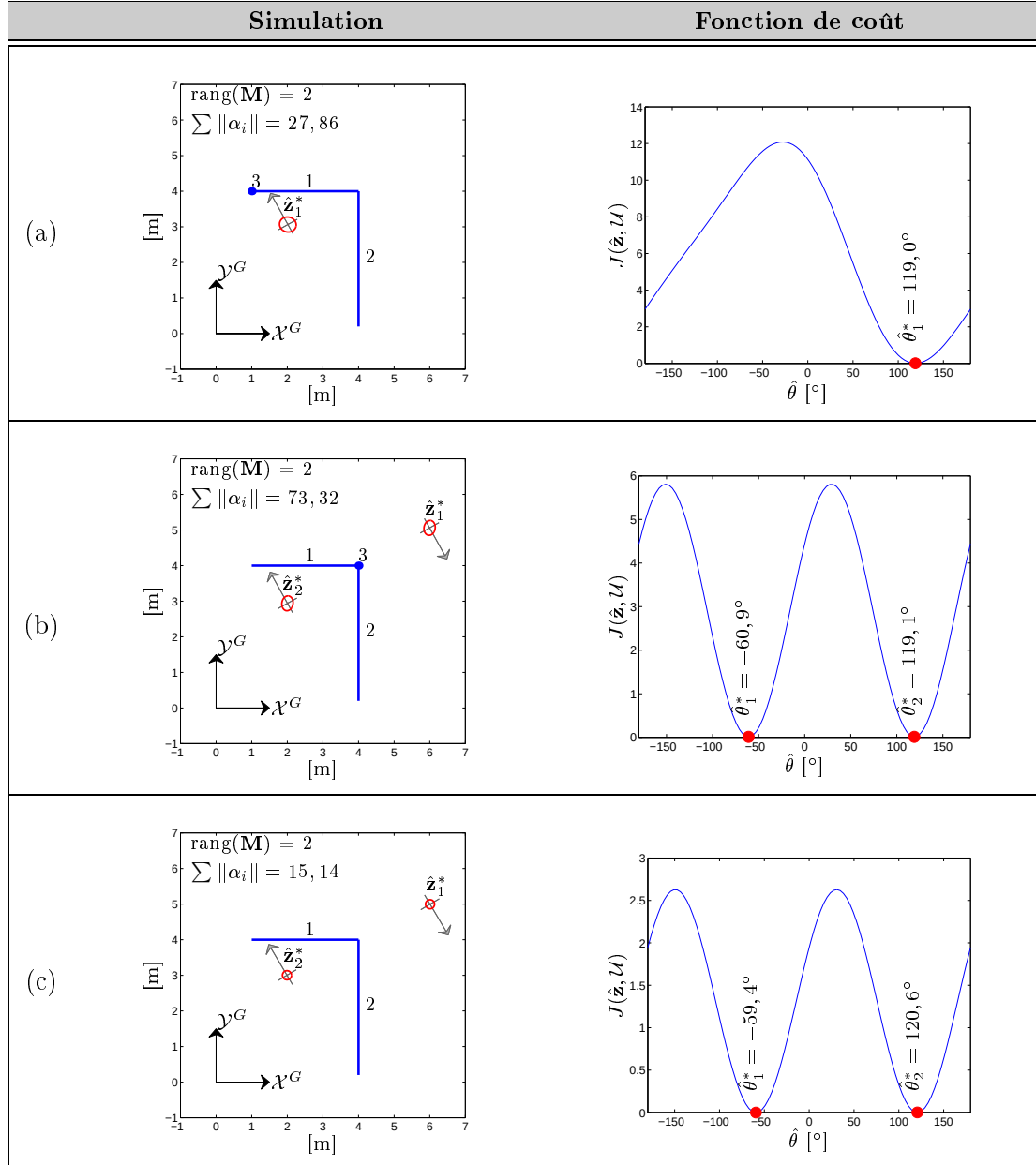
et montrent que les axes de solutions infinies ne sont pas informatifs. Remarquons qu'à la différence de l'exemple précédent, les axes non informatifs des solutions $\hat{\mathbf{z}}_1^*$ et $\hat{\mathbf{z}}_2^*$ sont inclinés par rapport aux axes $\mathcal{X}^G$ et $\mathcal{Y}^G$ du repère global. Lors de la fusion avec la prédiction $\hat{\mathbf{z}}_{k|k-1}$, les estimées sur $\hat{x}$ et sur $\hat{y}$ sont affectées en fonction de la contribution portée par les correspondances (comme avec le filtre de Kalman) ;

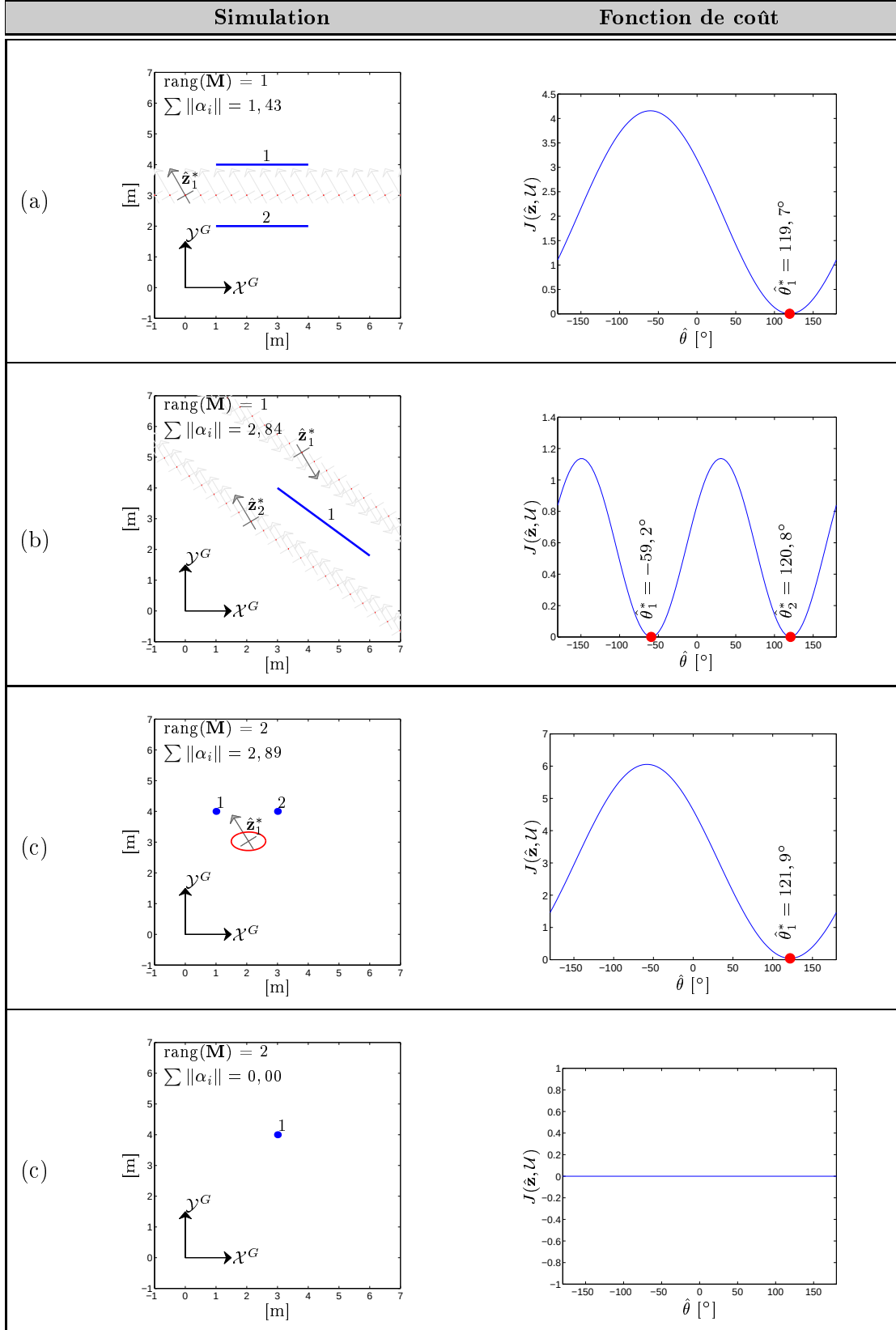
- *Deux points (Fig. 6.6(c)).* Contrairement aux cas de deux droites infinies des Figures 6.5(b) et 6.5(c), ceci est un exemple sans ambiguïté de solution ;
- *Un point (Fig. 6.6(d)).* Nous avons simulé le cas de l'exemple 6.2 pour confirmer que pour seulement une correspondance de type point, il n'existe pas de solution en utilisant le *MCP* incomplet. Ici la somme des valeurs absolues des coefficients du polynôme (6.62) est nulle.

6.5 Résultats expérimentaux

Nous proposons dans cette section une comparaison hors ligne des techniques de localisation en utilisant des données réelles acquises avec le robot Omni. Ces données sont les mesures codeurs, les images télémétriques et les images vidéo, correspondant aux expériences *Premier étage extension - B* et *Couloir robotique - B*, présentées respectivement dans les annexes G.7 et G.5. Nous avons choisi ces environnements pour leurs particularités qui seront analysées prochainement. Les cartes globales correspondantes ont été construites en utilisant la méthode sous contraintes stochastiques développée dans le chapitre 7. Cette méthode requiert un estimateur de position pour recalibrer les cartes locales le repère global. Pour éviter toute corrélation entre la carte globale et les techniques évaluées ici, nous avons utilisé la gyrodométrie comme estimateur de position pour la cartographie. Etant donné la durée de ces expériences, la gyrodométrie peut être considérée comme référence de positionnement. La preuve est donnée par la superposition cohérente des images télémétriques dans l'annexe G.

Nous avons évalué la performance des méthodes de localisation par filtrage stochastique (*FKE* parallèle, *FKE* séquentiel et *FKEI* séquentiel), et celle des méthodes *MCP* (*MCP* incomplet (μ -Mahalanobis), *MCP* incomplet (μ -Médiane), M-estimateur (μ -Tukey) et M-estimateur (μ -Huber)). On remarque que les approches par filtrage stochastique utilisent la représentation polaire pour les correspondances de type droite, et la représentation cartésienne pour les points (cf. section 6.3.2). Pour vérifier l'influence de la représentation des indices dans la performance de ces approches, nous avons inclus des implémentations utilisant la représentation unifiée (section 6.3.4.1) : *FKE* parallèle (rep. unifiée), *FKE* séquentiel (rep. unifiée), *FKEI* séquentiel (rep. unifiée).

FIG. 6.5 – Etude de cas pour *MCP* Incomplet (partie I).

FIG. 6.6 – Etude de cas pour *MCP* Incomplet (partie II).

Pour chaque environnement, nous faisons trois évaluations différentes, en fonction des paramètres utilisés pour calculer l'incertitude de l'estime. L'estime est donnée par l'odométrie calibrée, utilisant les paramètres $\hat{\lambda}$ estimés par identification. Nous avons vu que l'estime utilise un modèle de propagation de l'erreur qui dépend des composantes de la matrice diagonale $\mathbf{\Gamma} = \mathbf{Diag}(\gamma_x, \gamma_y, \gamma_\theta)$ (cf. section 6.2.5). Ces paramètres ont été ajustés de façon empirique pour obtenir une estimation consistante de l'incertitude. Rappelons ici ces valeurs :

$$\gamma_x = 10, \quad \gamma_y = 10, \quad \gamma_\theta = 5. \quad (6.81)$$

Nous effectuons des exécutions avec d'autres valeurs pour $\mathbf{\Gamma}$. Cela peut nous aider à vérifier l'influence d'une estime plus pessimiste ou plus optimiste sur la performance des approches de localisation par cartes géométriques. Par ailleurs, il est commun de considérer que pour la mise au point des approches par filtrage de Kalman, il suffit de bien déterminer l'incertitude de la prédiction, en occurrence l'estime. Des évaluations avec une estime plus pessimiste ou une estime plus optimiste permettent de vérifier ce principe de base. Ainsi, dans l'estime pessimiste, nous avons une incertitude qui croît deux fois plus vite en utilisant

$$\gamma_x = 20, \quad \gamma_y = 20, \quad \gamma_\theta = 10, \quad (6.82)$$

Pour l'estime optimiste, les gains sont deux fois moins importants :

$$\gamma_x = 5, \quad \gamma_y = 5, \quad \gamma_\theta = 2, 5. \quad (6.83)$$

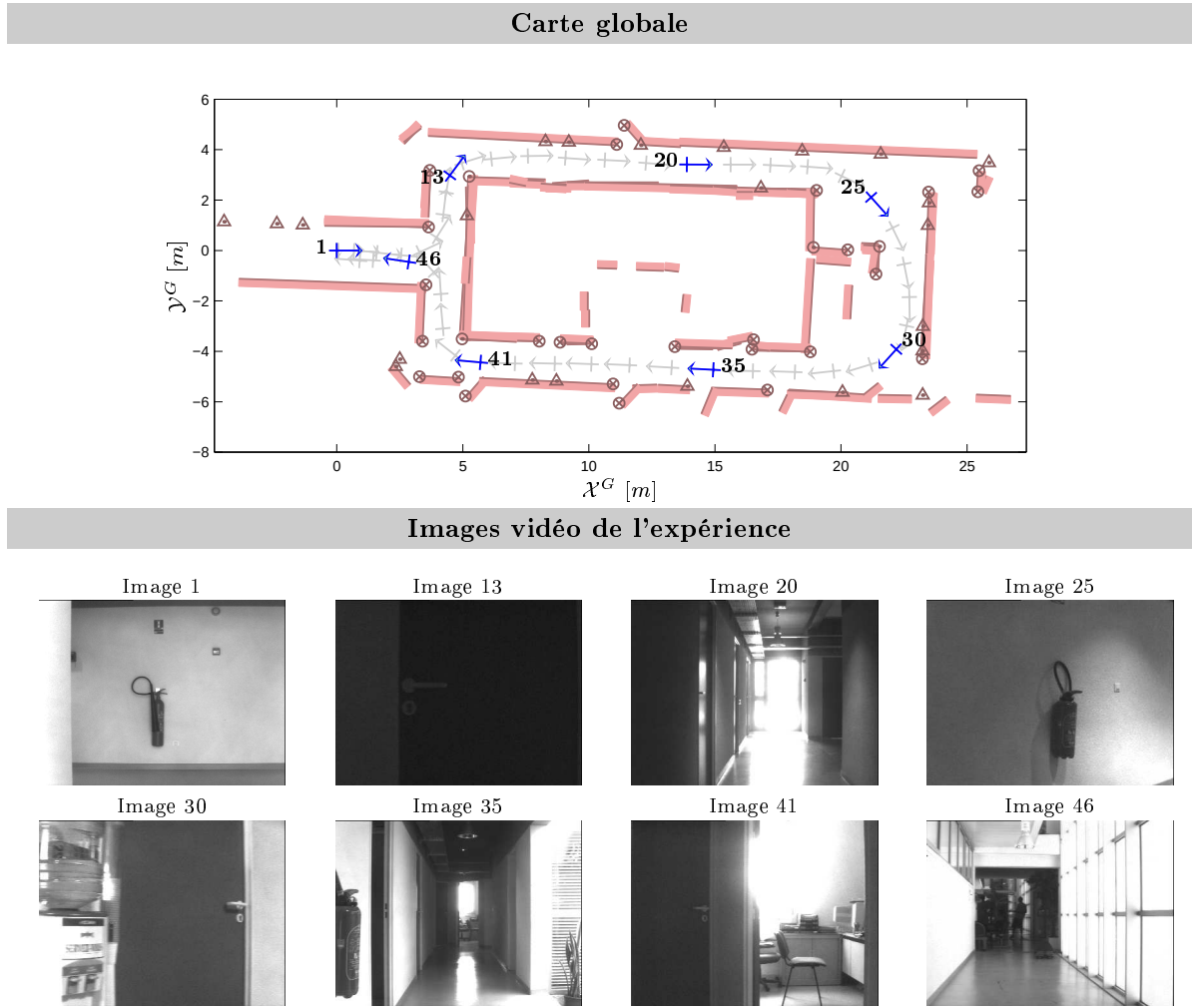
Quelle que soit la méthode d'estimation utilisée, le choix de $\mathbf{\Gamma}$ est important. L'estime doit être suffisamment pessimiste pour permettre de retrouver les bonnes correspondances entre la carte globale et la carte locale. Cependant, une prédiction trop pessimiste peut incorporer des fausses correspondances. Par ailleurs, avec une prédiction optimiste on risque de n'en trouver aucune.

Nous comparons les différentes approches de localisation à partir de mesures de l'écart de l'estimation par rapport à la gyrodométrie. Pour une exécution donnée, on enregistre les séquences $\varepsilon_{\hat{x}}$, $\varepsilon_{\hat{y}}$ et $\varepsilon_{\hat{\theta}}$ résultantes pour chaque méthode de localisation. Ces séquences sont respectivement les écarts d'estimation des composantes x , y et θ de position absolue par rapport à la gyrodométrie. Les observations que nous faisons pour chaque expérience sont basées sur les mesures statistiques sur les séquences $\varepsilon_{\hat{x}}$, $\varepsilon_{\hat{y}}$ et $\varepsilon_{\hat{\theta}}$. L'annexe H définit ces mesures qui nous permettent d'analyser le biais d'estimation, l'écart absolu moyen, et la consistance des estimées. L'annexe H présente également sous forme de tableaux comparatifs les mesures statistiques pour les expériences menées.

Dans la suite, nous présentons les résultats obtenus pour les deux environnements d'évaluation.

6.5.1 Expérience dans le site *Premier étage extension - B*

Nous avons sur la Figure 6.7 la carte globale du site *Premier étage extension - B*, avec les positions de référence du robot (certaines sont mises en évidence avec leur index dans la séquence). Les images vidéo correspondant aux points de référence sont présentées sur cette figure. Au total, nous avons 48 cycles de recalage effectués à des intervalles de 2 secondes. Comme nous pouvons le

FIG. 6.7 – Conditions d'expérimentation dans le site *Premier étage extension - B*.

voir dans l'annexe G.7, dans cette expérience le robot a effectué une trajectoire fermée à des vitesses de translation et de rotation allant jusqu'à $85,94 \text{ cm/s}$ et $24,61^\circ/\text{s}$. A ces vitesses importantes, se rajoute comme difficulté le fait que ce robot se déplace dans un couloir étroit (largeur inférieure à deux mètres), ce qui augmente l'incertitude des indices visuels étant donné l'angle de vue réduit de la caméra.

6.5.1.1 Analyse des résultats

Nous présentons sur la Figure 6.8 les écarts $\varepsilon_{\hat{x}}$, $\varepsilon_{\hat{y}}$ et $\varepsilon_{\hat{\theta}}$ obtenus pour certaines approches de localisation en utilisant une estime avec incertitude normale. Nous présentons uniquement les résultats obtenus avec le *FKE* séquentiel, le *FKEI* séquentiel, les méthodes *MCP* et les M-estimateurs. Les intervalles $\pm 3\sigma$ sont représentés par une zone grise. Nous remarquons que la partie du trajet qui induit le plus d'incertitude pour ces méthodes est celle située entre les index 25 et 30. Cet intervalle correspond à une grande courbe en forme d'arc de cercle. Parmi ces approches, seul le

FKEI séquentiel a présenté une divergence temporaire autour de la position d'index 25. Comme nous pouvons voir sur la Figure 6.8(b), cette approche ne réussit pas à corriger des erreurs à partir de la position 20, et cela se répercute sur les prochains cycles de localisation. Le *FKE* séquentiel semble être bien réglé, de sorte que ses résultats sont satisfaisants et consistants avec les intervalles de confiance estimés par le filtre. On remarque aussi que toutes les approches *MCP* ont présenté des bons résultats, voire même moins sensibles aux perturbations dues à la courbure que nous venons de mentionner.

Les mesures sont présentées en annexe sur le Tableau H.1. Dans ce tableau nous regroupons les résultats de toutes les approches évaluées, où les trois meilleures performances pour les mesures de biais et d'écartement sont écrites en gras. Nous avons observé que le *FKE* parallèle a divergé pendant cette expérience. Néanmoins, le *FKE* parallèle en utilisant la représentation unifiée a bien convergé et a présenté des bons résultats. Les meilleures performances sont celles des approches *MCP*, mais elles ne diffèrent pas trop de celles des approches *FKE* séquentiel (avec ou sans représentation unifiée). On remarque que le *FKEI* séquentiel a présenté des résultats moins satisfaisants, indépendamment de la représentation des indices en correspondance. Cela est peut être dû à sa caractéristique principale, qui est de converger vers le mode de la distribution conditionnelle $f(\mathbf{z}|\mathbf{y}_k)$ le plus proche de la prédiction $\hat{\mathbf{z}}_{k|k-1}$ (voir discussion dans la section 6.3.3). Cela se vérifie par le fait que, non seulement pour cette expérience mais aussi pour toutes les autres qui suivent, les approches *FKEI* séquentiel présentent des performances semblables indépendamment de la représentation des indices. Ainsi, nous concluons déjà que le *FKEI* séquentiel n'est pas adapté au traitement des cas de correspondances multiples, qui donnent des distributions conditionnelles $f(\mathbf{z}|\mathbf{y}_k)$ multimodales. On note aussi que, en ce qui concerne l'inconsistance des estimées, les approches *FKE* séquentiel ont présenté de plus mauvais résultats pour l'estimation du cap (*i.e.*, θ). En effet, nous avons déjà vérifié cela dans des travaux précédents [36].

Commentons maintenant les résultats obtenus avec les autres valeurs pour Γ . Les Tableaux H.2 et H.3 présentent respectivement les mesures de performance avec une odométrie calibrée plus pessimiste et plus optimiste. Avec une odométrie plus pessimiste, les approches qui ont présenté une détérioration plus importante de leur performance sont effectivement celles dépendantes de la distance de Mahalanobis : les filtres stochastiques et le *MCP* incomplet (μ -Mahalanobis). En fait, avec une odométrie pessimiste on risque d'incorporer plus de fausses correspondances, étant donné l'incertitude exagérée de la prédiction. Et cela affecte directement les méthodes qui intègrent ces mesures sans incorporer dans leurs structures une sélection plus efficace. Étant donné l'incertitude importante de la prédiction, les approches par filtrage stochastique font plus confiance aux mesures, bonnes ou mauvaises, ce qui les rend moins performantes. Avec une odométrie plus optimiste (cf. Tableau H.3), toutes les méthodes ont présenté des résultats meilleurs que ceux obtenus avec l'odométrie normale. Cela peut indiquer que les gains optimistes de l'équation (6.83) sont plus appropriés pour calculer l'incertitude de cette technique d'estime.

6.5.1.2 Conclusions de l'expérience

Pour conclure cette expérience, on remarque que les approches *MCP* incomplet et M-estimateur ont toujours présenté d'excellents résultats, qui bien que pessimistes ont été plus robustes vis à vis

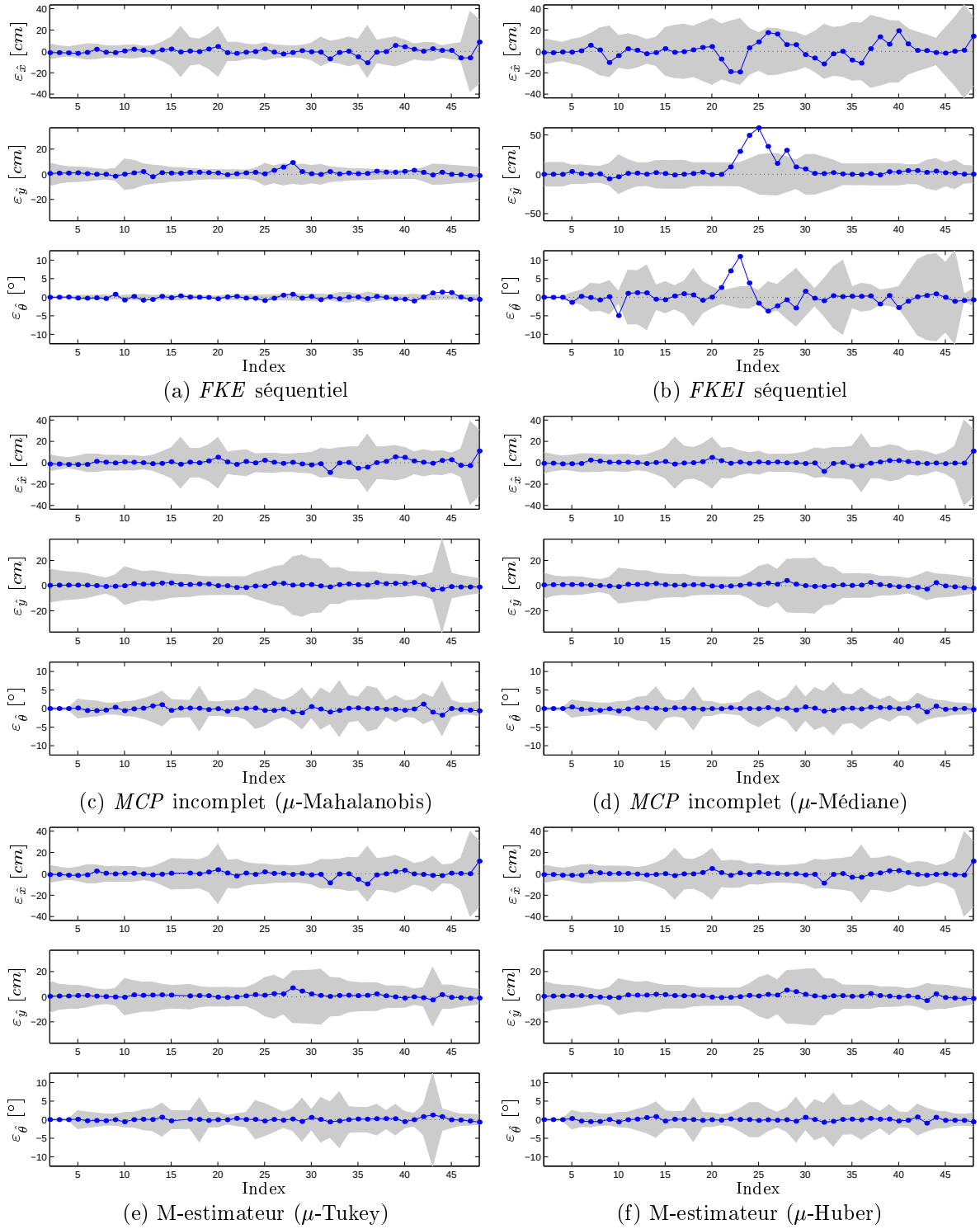


FIG. 6.8 – Ecarts et intervalles de confiance $\pm 3\sigma$ de localisation dans le site *Premier étage extension* - *B* (odométrie calibrée).

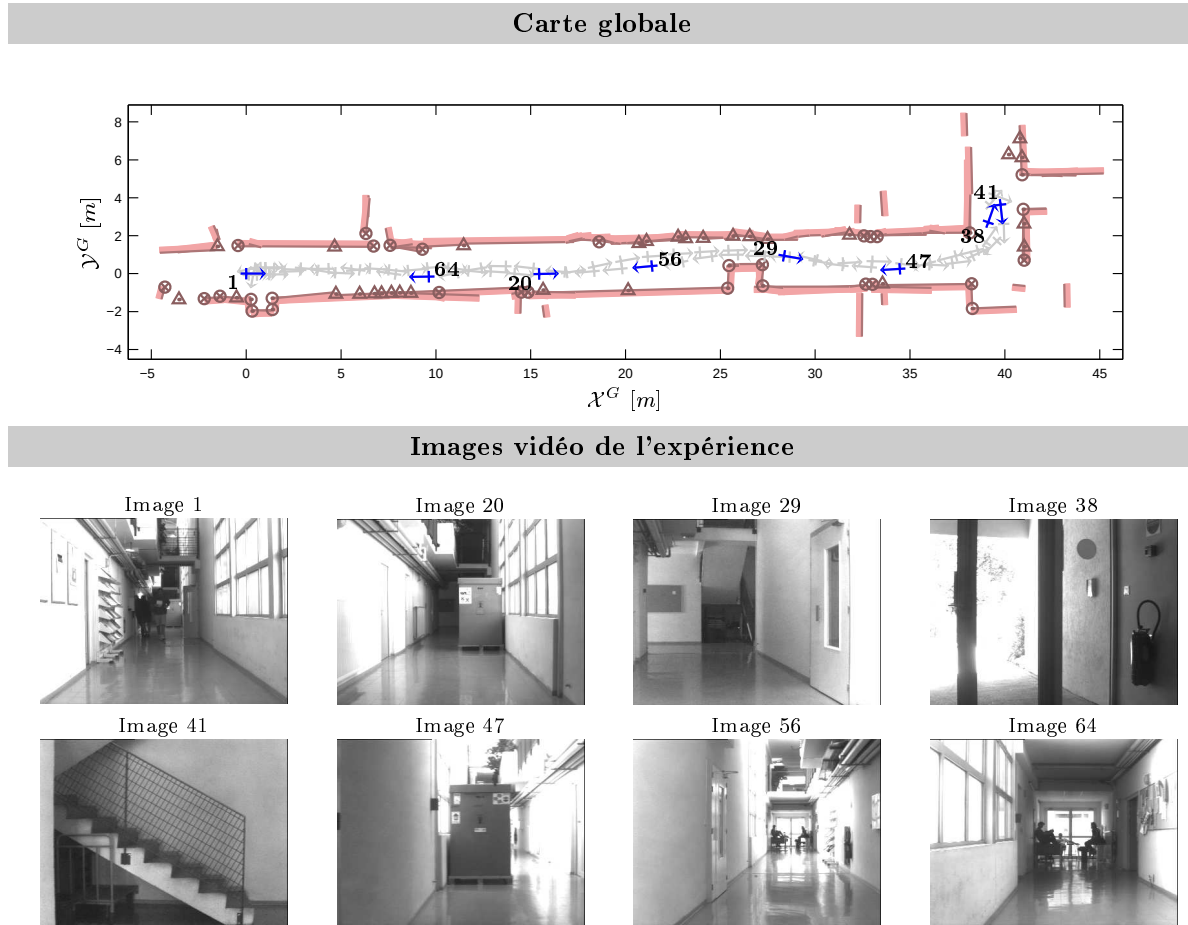


FIG. 6.9 – Conditions d'expérimentation dans le site *Couloir robotique - B*.

de l'incertitude de l'odométrie.

Le *FKE* parallèle a convergé seulement lorsque la représentation unifiée a été utilisée pour les indices géométriques. Il semble que le processus parallèle d'estimation de ce filtre soit moins efficace pour des mesures de nature différente. Le filtre itératif *FKEI* séquentiel a été très perturbé par des correspondances multiples, ce qui induit des modes multiples dans la vraisemblance conditionnelle de la position (*i.e.*, multi-hypothèses). L'approche *FKE* séquentiel, bien que moins précise, a présenté elle aussi des résultats satisfaisants. Cela justifie le fait que cette approche, une fois bien "ajustée", est la plus utilisée pour la localisation sur cartes géométriques [7][8][83][108][170][178].

6.5.2 Expérience dans le site *Couloir robotique - B*

Cette deuxième série d'expériences a été réalisée avec des données du site *Premier étage extension - B*, qui présente un niveau de difficulté supérieur par rapport à celui de l'expérience précédente. La Figure 6.9 présente la carte globale de ce site, avec les positions de référence du robot. Les images vidéo correspondant aux points de référence sont aussi présentées sur cette figure.

Au total, nous avons 75 cycles de recalage effectués à des intervalles de 2 secondes. Comme nous pouvons le voir dans l'annexe G.5, ce site est constitué d'un long couloir (à peu près 40 m) où le robot a effectué un trajet aller-retour à des vitesses de translation et de rotation allant jusqu'à 91,83 cm/s et 39,4 °/s, respectivement (plus importantes que celles de l'expérience précédente). Ce site réel est constitué d'un certain nombre d'artefacts qui se traduisent par des arêtes dans la carte globale. Dans certains endroits, comme nous pouvons les voir sur la carte de la Figure 6.9, ces arêtes sont assez proches les unes des autres. En plus, étant donné la présence de baies vitrées, le changement de lumière important dans cet environnement se traduit par du biais dans les mesures télémétriques. Celui-ci a été atténué lors du processus de cartographie. Nous observons aussi de grandes rotations entre les positions 38 et 41, ce qui peut ne pas être bien mesuré par odométrie.

6.5.2.1 Analyse des résultats

La Figure 6.10 présente les écarts $\varepsilon_{\hat{x}}$, $\varepsilon_{\hat{y}}$ et $\varepsilon_{\hat{\theta}}$ obtenus par la localisation avec le *FKE* séquentiel, le *FKEI* séquentiel, les méthodes *MCP* et les M-estimateurs en considérant l'odométrie calibrée comme technique d'estime. On remarque que les approches séquentielles *FKE* et *FKEI* ont présenté des divergences dans des endroits différents. Le *FKE* séquentiel a divergé dans la position 56, ce qui a engendré une erreur d'estimation sur x non corrigée dans les positions suivantes et qui a enfin affecté l'estimation sur y (voir Figure 6.10(a)). Les approches *MCP* et les M-estimateurs ont tous convergé, présentant moins de sensibilité aux perturbations. Les mesures sont présentées en annexe sur le Tableau H.4. En fait, aucune méthode par filtrage stochastique n'a convergé dans cette expérience, indépendamment de la représentation des indices géométriques.

Pour une vérification plus complète, nous présentons respectivement sur les Tableaux H.5 et H.6 les résultats obtenus avec une odométrie calibrée plus pessimiste et une odométrie calibrée plus optimiste. Nous observons que toutes les méthodes par filtrage stochastique ont divergé, sauf le *FKE* séquentiel. Cependant, notre critère de divergence est fondé sur une simple évaluation des moyennes des écarts absolus (cf. voir définitions au début de l'annexe H). Si nous considérons ces valeurs pour le *FKE* séquentiel, on remarque qu'elles sont encore trop importantes, autour de 9 cm. Nous observons aussi que, malgré les différences importantes sur l'incertitude des deux techniques d'estime, les approches proposées dans le cadre de notre étude (le *MCP* et les M-estimateurs) ont toutes convergé et présenté des bons résultats qui ne diffèrent pas trop les uns des autres.

6.5.2.2 Commentaires

Les résultats obtenus avec l'expérience précédente dans la section 6.5.1, nous ont laissé penser que les approches par filtrage stochastique étaient bien "ajustées". En effet, de bons résultats quantitatifs ont été obtenus, malgré une meilleure performance des approches *MCP* et des M-estimateurs. Cependant, quand le niveau de difficulté a augmenté avec les expériences menées dans cette dernière section, nous avons eu du mal à obtenir la convergence de ces méthodes. Et cela, malgré l'utilisation du principe de base pour la mise au point des filtres de Kalman, qui consiste à les essayer avec différents niveaux d'incertitude de la prédiction.

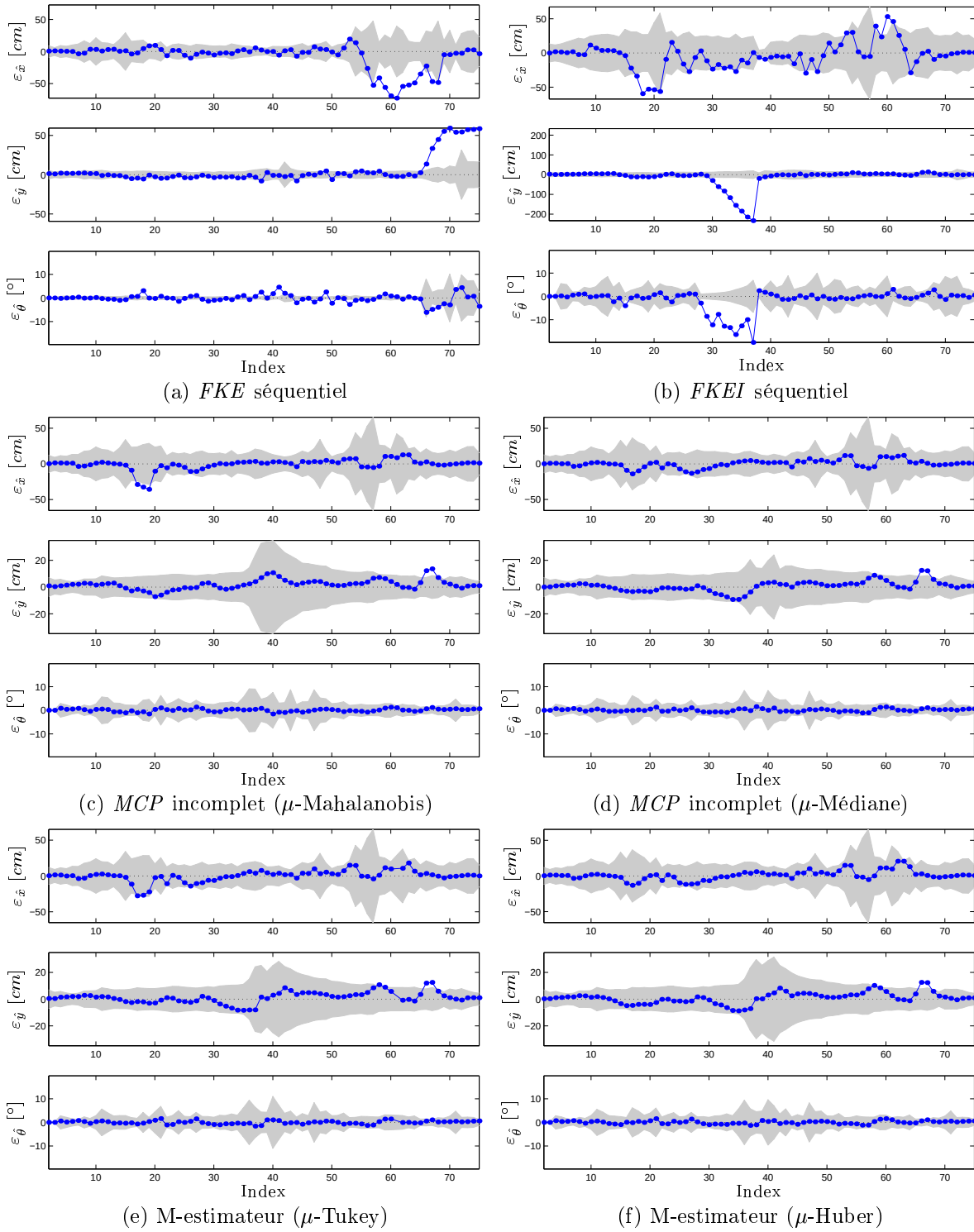


FIG. 6.10 – Ecart et intervalles de confiance $\pm 3\sigma$ de localisation dans le site *Couloir robotique - B* (odométrie calibrée).

Cette expérience nous a permis de vérifier la robustesse des approches que nous proposons pour la localisation, notamment le *MCP* incomplet (μ -Médiane) et les M-estimateurs. Bien que les temps maximum d'exécution figurent sur les tableaux de l'annexe H, nous n'y avons pas fait référence dans notre analyse des résultats car nous considérons que ce facteur est moins important que la robustesse de l'approche évaluée.

6.6 Conclusion

Nous avons présenté dans ce chapitre des nouvelles approches robustes pour la localisation de robots mobiles dans des environnements d'intérieur. Ces approches ont été intégrées et évaluées dans une architecture de type prédiction-correction en utilisant des cartes géométriques d'environnement. Des résultats expérimentaux avec le robot Omni ont permis de les comparer à certaines approches classiques de localisation utilisant le filtrage stochastique. Ces comparaisons ont été menées en utilisant des mesures de référence fournies par la gyrométrie, technique de localisation par estime que nous avons implantée avec un gyromètre laser de grande précision. Les cartes géométriques d'environnement ont été construites avec des données réelles, en appliquant la technique de cartographie qui nous décrirons dans le chapitre 7.

La robustesse des approches *MCP* et des M-estimateurs se matérialise de deux façons : l'estimation optimale non-linéaire de la position à partir de correspondances, même pour des cas de données incomplètes, et le rejet de correspondances aberrantes par des fonctions de pondération robustes. Néanmoins, il faut prendre en considération les résidus aberrants acceptables par l'estimateur. Notamment, si le nombre d'aberrations est suffisamment important et dépasse le *breakdown point*⁹ des estimateurs, on peut avoir des résultats incohérents d'estimation. Enfin, l'estimateur ne pourra pas distinguer mesures aberrantes et bruit de mesure. Cela est déjà arrivé dans certaines expériences avec nos méthodes, mais dans un nombre de cas bien moins important qu'avec les approches non-robustes.

⁹Ce terme très connu des statistiques robustes correspond au pourcentage maximum de mesures aberrantes qui sont tolérées par un estimateur.

Cartographie globale découplée

7.1 Introduction

Nous proposons dans ce chapitre une approche découplée pour la cartographie d'environnements et la localisation concurrentes (*CLC*) [34]. Le terme découplage vient du fait que les processus de localisation et de mise à jour de la carte globale sont réalisés de manière complètement indépendante l'une de l'autre, et qu'aucune mesure d'inter-corrélation n'est maintenue. Le *CLC* est un problème intrinsèquement mal-conditionné, où des erreurs de localisation ou d'estimation des structures cartographiques peuvent faire diverger le processus de cartographie.

Les approches couplées [52][61] [71][141], cherchent à maintenir des estimées des corrélations entre la position du robot et les structures de la carte. Comme l'ont remarqué Uhlmann et Julier [116], pour ces approches dont le processus d'estimation stochastique utilise le filtre de Kalman étendu, la maintenance des corrélations est une des grandes difficultés. En plus, de la même façon que le filtre propage les covariances des estimées, un biais de mesure peut aussi affecter le processus d'estimation et provoquer la divergence du filtre. A cela on rajoute la complexité de calcul et les limitations déjà assez bien connues de ce filtre sous-optimal (voir annexe E.4).

Dans cette étude, nous proposons une approche découplée [34] qui permet, au moins du point de vue théorique, de prendre en compte deux problèmes posés par la cartographie globale :

- Le maintien de la consistance stochastique des structures de la carte. En effet, la présence de structures inconsistantes dans la carte globale peut conduire facilement à des erreurs de localisation, ou à un échec lors du processus de cartographie et localisation simultanées ;

- Le maintien des relations géométriques entre des structures de types différents (arêtes de types bord et coin, et demi-plans associés). Le respect de ces relations au niveau de la cartographie locale est un pré-requis pour assurer la cohérence géométrique entre la carte construite et l'environnement. Le processus de cartographie globale doit être capable de propager les contraintes de couplage entre les structures.

7.2 Aspects théoriques de la cartographie et localisation découplés

L'approche que nous proposons traite la localisation et la cartographie globale comme des problèmes séparés. Le problème de la localisation a été abordé dans le chapitre 6. Pour la méthode de localisation proposée ici, un pré-requis important est que l'estimée de localisation soit consistante, ou même pessimiste. Nous avons vu dans les résultats expérimentaux du chapitre 6 que les techniques de localisation fournissent des estimées qui satisfont ces conditions. Ces résultats ont été obtenus avec des cartes d'environnement construites avec la méthode proposée dans ce chapitre. Ainsi, si l'estimation des structures des cartes d'environnement n'était pas consistante ou pessimiste, les approches de localisation (notamment celles basées sur le filtrage stochastique) auraient du mal à présenter des estimées consistantes. Ainsi, lors de la cartographie globale, nous mettons essentiellement l'accent sur la consistance des structures.

Notre approche s'appuie sur des travaux récents en estimation stochastique :

- I. L'échantillonnage sélectif [114][113] est employé pour garantir la propagation consistante des moyennes et des covariances de variables incertaines lors de transformations non-linéaires;
- II. Des filtres linéaires optimaux sont mis en oeuvre dans toutes procédures d'estimation (fusion et satisfaction de contraintes stochastiques). Appliqués à l'estimation d'état de modèles linéaires consistants ou pessimistes, ces filtres garantissent une estimation consistante ou pessimiste (cf. théorèmes E.2 et E.1);

Ainsi, nous évitons d'utiliser des filtres sous-optimaux dans le processus de cartographie car ils peuvent injecter un biais d'estimation. Néanmoins, si un tel biais est présent dans des estimées trop pessimistes des structures cartographiques, la cartographie peut diverger. Pour minimiser ce risque, nous appliquons des contraintes stochastiques lors de la construction incrémentale de la carte.

7.3 Fusion de structures cartographiques

La plupart des travaux en cartographie réalisent la mise à jour des structures cartographiques dans l'espace d'observation local. Par exemple, si un demi-plan global a une correspondance établie dans la carte locale, la correspondance locale est utilisée comme une mesure dans son propre repère. Ainsi, pour la mise à jour de la structure globale à partir de la correspondance locale, on applique généralement une approche du genre filtre de Kalman étendu en utilisant un modèle de mesure non-linéaire.

Pour éviter cette solution sous-optimale, nous avons choisi de réaliser la fusion de toutes les structures géométriques dans un même repère, le repère global. Ceci est le concept *converted measurement* qui a été utilisé avec succès dans des applications de suivi de cibles par radar [145], et qui permet l'utilisation de filtres linéaires optimaux. Néanmoins, comme le font remarquer Tam *et al.* [199], des problèmes d'inconsistance peuvent survenir si les incertitudes en jeu sont trop importantes. Cela est dû en partie aux limitations connues de la propagation de covariances par linéarisation (cf. Exemple B.2). Ainsi, pour permettre une propagation consistante des incertitudes lors du changement de repère des indices géométriques, nous utilisons l'échantillonnage sélectif [114][113], présenté dans l'annexe B.4.2.

Le procédé de fusion reste néanmoins spécifique au type de structure.

7.3.1 Fusion de demi-plans

Soit deux demi-plans $D_i = \{S_i, \mathbf{v}_i\}$ et $D_j = \{S_j, \mathbf{v}_j\}$, tels que $S_i = \{L_i, \mathbf{p}_{a,i}, \mathbf{p}_{b,i}\}$ et $S_j = \{L_j, \mathbf{p}_{a,j}, \mathbf{p}_{b,j}\}$. La fusion de ces deux entités produit un demi-plan $D = \{S, \mathbf{v}\}$, tel que $S = \{L, \mathbf{p}_a, \mathbf{p}_b\}$, obtenu en trois étapes :

- I. Fusion stochastique optimale entre les primitives géométriques $L_i = \{\mathbf{l}_i, \Lambda_i\}$ et $L_j = \{\mathbf{l}_j, \Lambda_j\}$, qui donne $L = \{\mathbf{l}, \Lambda\}$. Ceci est fait à l'aide d'un des deux filtres stochastiques linéaires suivants (cf. Annexe E.3) :

Filtre de Kalman :

$$\begin{cases} \mathbf{l} = \mathbf{l}_i + \Lambda_i \cdot (\Lambda_i + \Lambda_j)^{-1} \cdot (\mathbf{l}_j - \mathbf{l}_i) \\ \Lambda = \Lambda_i - \Lambda_i \cdot (\Lambda_i + \Lambda_j)^{-1} \cdot \Lambda_i \end{cases} \quad (7.1)$$

Intersection de covariances modifiée :

$$\begin{cases} \mathbf{l} = \mathbf{l}_i + (1 - \omega) \cdot \Lambda_i \cdot ((1 - \omega) \cdot \Lambda_i + \omega \cdot \Lambda_j)^{-1} \cdot (\mathbf{l}_j - \mathbf{l}_i) \\ \Lambda = \frac{1}{\omega} \cdot \Lambda_i - \frac{(1-\omega)}{\omega} \cdot \Lambda_i \cdot ((1 - \omega) \cdot \Lambda_i + \Lambda_j)^{-1} \cdot \Lambda_i \end{cases} \quad (7.2)$$

tel que $\omega \in [\gamma, 1 - \gamma]$ et $\omega = \arg \min \det(\Lambda)$.

- II. Calcul géométrique des extrémités : les extrémités du demi-plan fusionné D sont choisies parmi les projections des extrémités de D_i et D_j sur la droite $\mathbf{l}$ de façon que l'on ait la plus grande longueur possible. Cette étape est illustrée sur la Figure 7.1(a).
- III. Calcul géométrique des paramètres de visibilité : on détermine un point de contrôle $\mathbf{p}_c$, qui se trouve à une distance orthogonale $l = 5m$ par rapport au centre de gravité du segment fusionné S . On détermine le vecteur $\mathbf{v}$ de paramètres de visibilité tel que $\mathbf{p}_c$ ait la même visibilité $\mathcal{V}$ par rapport à D , D_i et D_j . Cette étape est illustrée sur la Figure 7.1(b), et est accomplie par la résolution de :

$$\mathcal{V}(\mathbf{p}_c, \mathbf{v}) = \mathcal{V}(\mathbf{p}_c, \mathbf{v}_i) = \mathcal{V}(\mathbf{p}_c, \mathbf{v}_j). \quad (7.3)$$

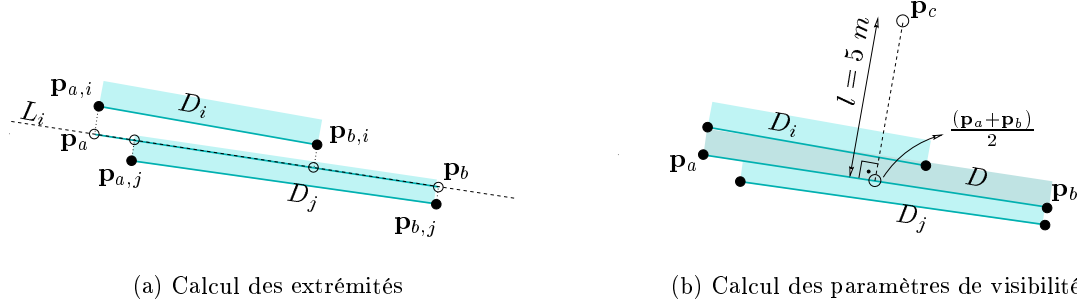


FIG. 7.1 – Détermination des paramètres géométriques pour la fusion de demi-plans.

7.3.2 Fusion d'arêtes

Soit deux arêtes $A_i = \{P_i, \kappa_i^A, n_{1,i}^A, n_{2,i}^A\}$ et $A_j = \{P_j, \kappa_j^A, n_{1,j}^A, n_{2,j}^A\}$ tels que $P_i = \{\mathbf{p}_i, \Lambda_{\mathbf{p}_i}\}$ et $P_j = \{\mathbf{p}_j, \Lambda_{\mathbf{p}_j}\}$. La fusion entre ces deux entités donne une arête $A = \{P, \kappa^A, n_1^A, n_2^A\}$, telle que $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$, obtenue en une seule étape :

- I. Fusion stochastique optimale entre les primitives géométriques P_i et P_j . Ceci est réalisé à l'aide d'un des deux filtres stochastiques linéaires suivantes :

Filtre de Kalman :

$$\begin{cases} \mathbf{p} = \mathbf{p}_i + \Lambda_{\mathbf{p}_i} \cdot (\Lambda_{\mathbf{p}_i} + \Lambda_{\mathbf{p}_j})^{-1} \cdot (\mathbf{p}_j - \mathbf{p}_i) \\ \Lambda_{\mathbf{p}} = \Lambda_{\mathbf{p}_i} - \Lambda_{\mathbf{p}_i} \cdot (\Lambda_{\mathbf{p}_i} + \Lambda_{\mathbf{p}_j})^{-1} \cdot \Lambda_{\mathbf{p}_i} \end{cases} \quad (7.4)$$

Intersection de covariances modifiée :

$$\begin{cases} \mathbf{p} = \mathbf{p}_i + (1 - \omega) \cdot \Lambda_{\mathbf{p}_i} \cdot ((1 - \omega) \cdot \Lambda_{\mathbf{p}_i} + \omega \cdot \Lambda_{\mathbf{p}_j})^{-1} \cdot (\mathbf{p}_j - \mathbf{p}_i) \\ \Lambda_{\mathbf{p}} = \frac{1}{\omega} \cdot \Lambda_{\mathbf{p}_i} - \frac{(1 - \omega)}{\omega} \cdot \Lambda_{\mathbf{p}_i} \cdot ((1 - \omega) \cdot \Lambda_{\mathbf{p}_i} + \Lambda_{\mathbf{p}_j})^{-1} \cdot \Lambda_{\mathbf{p}_i} \end{cases} \quad (7.5)$$

tel que $\omega \in [\gamma, 1 - \gamma]$ et $\omega = \arg \min \det(\Lambda_{\mathbf{p}})$.

La fusion d'arêtes ne permet de déterminer que la position P de l'arête résultante. Les attributs κ^A, n_1^A et n_2^A seront déterminés au cours de la mise à jour de la carte globale (cf. section 7.5.3).

7.3.3 Exemple simulé

L'exemple suivant illustre l'application des techniques de fusion de structures en simulation.

Exemple 7.1: Considérons le cas d'un robot qui construit une carte locale contenant seulement deux structures : un demi-plan D_1^R et un bord A_1^R , tels que A_1^R est associé à D_1^R . Supposons que ces structures ont respectivement comme correspondantes D_1^G et A_1^G dans la carte globale, telles que A_1^G est un bord associé à D_1^G . Ces structures sont virtuellement couplées dès lors que nous considérons qu'un bord doit se trouver sur la droite du demi-plan associé.

Ce cas est illustré sur la Figure 7.2(a), où les structures de la carte globale sont présentées en gris foncé et où la position du robot est présentée sous forme iconique en bas de la figure.

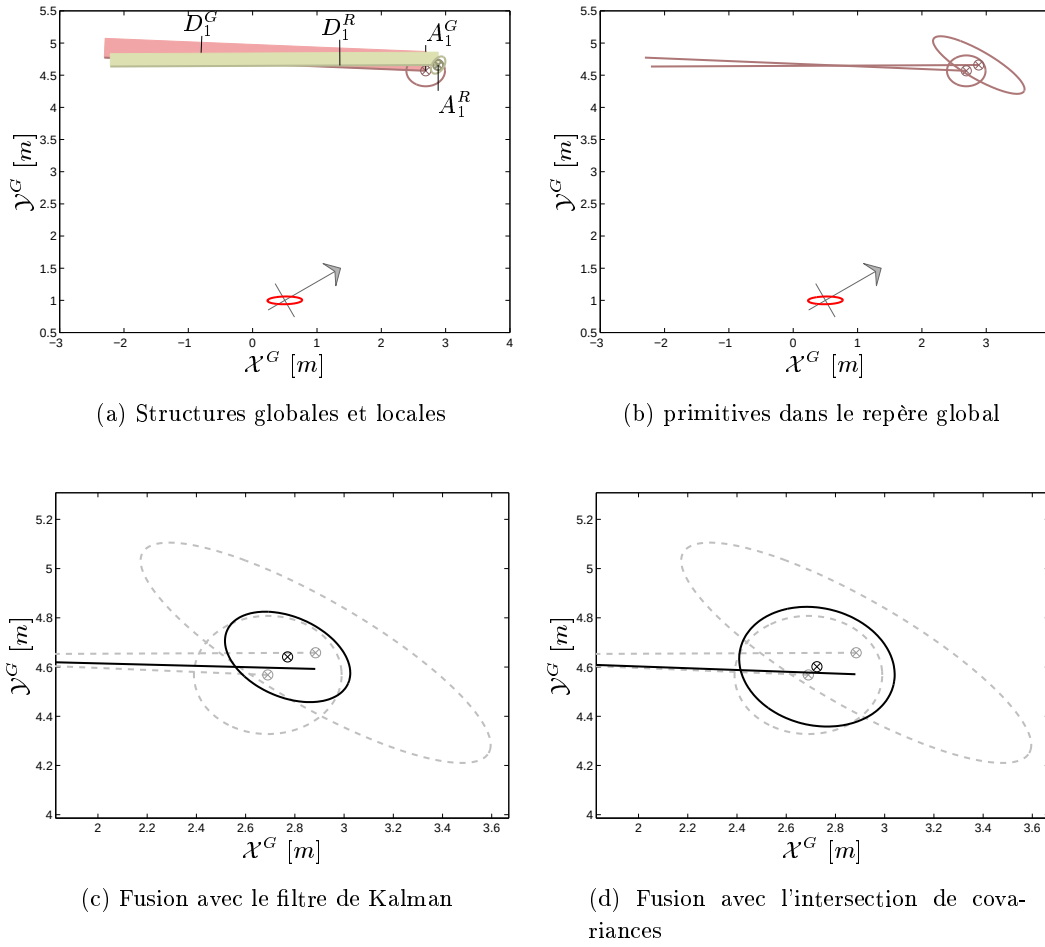


FIG. 7.2 – Exemple de la fusion de structures.

Les ellipses d'incertitude 3σ sont également dessinées, tant pour la position du robot, que pour les arêtes. L'incertitude du bord local est représentée dans ce même repère, ce qui explique la petite ellipse associée à A_1^R .

Etant donné les correspondances $D_1^G \xleftrightarrow{Z} D_1^R$ et $A_1^G \xleftrightarrow{Z} A_1^R$, nous nous intéressons à la fusion des primitives de ces structures. La fusion est réalisée dans le repère global, après avoir appliqué le changement de repères (utilisant Z) pour amener les structures locales dans le repère global. Nous présentons sur la Figure 7.2(b) les primitives géométriques et les incertitudes des arêtes dans ce repère. L'incertitude de ces structures a augmenté après changement de repères, étant donné l'incertitude sur la position du robot. Les Figures 7.2(c) et 7.2(d) présentent le segment de droite et le point résultant de la fusion en utilisant respectivement le filtre de Kalman et l'intersection de covariances ($\gamma = 0, 2$). Les primitives d'origine sont illustrées en gris. $\diamond$

Les résultats présentés dans l'exemple 7.1 confirment que la fusion réalisée avec l'intersection de covariances donne des matrices de covariance plus importantes que celles obtenues avec le filtre de Kalman. On peut donc se poser la question de l'intérêt d'utiliser une telle approche dans la

fusion d'entités géométriques.

7.4 Fusion de structures sous contraintes

Dans l'exemple 7.1, nous remarquons que les primitives résultant de la fusion n'ont pas conservé la relation géométrique qu'avaient les primitives d'origine. Ainsi, l'arête A_1^G se trouvait sur la droite de D_1^G (de même pour A_1^R et D_1^R). Cette relation n'est plus vérifiée après la fusion. En effet, nous n'avons pas pris en compte de contrainte liée au couplage entre les structures.

Cette section est dédiée à la propagation de contraintes liées au couplage entre les structures cartographiques. Nous présentons dans la section 7.4.1 un bref survol de quelques approches destinées à la prise en compte de contraintes dans des processus d'estimation stochastique. À partir de considérations portées sur ces approches, nous présentons notre solution dans la section 7.4.2.

Nous commençons par définir les contraintes déterministes et les contraintes stochastiques.

Définition 7.1 (Contraintes déterministes): *Considérons un ensemble de N contraintes déterministes d'égalité associées à une estimée $\mathbf{x}$ qui s'écrivent sous la forme générale*

$$\mathbf{h}(\mathbf{x}, \psi) = \mathbf{c}, \quad (7.6)$$

où ψ représente les variables qui décrivent la fonction $\mathbf{h}$ et $\mathbf{c} \in \mathbb{R}^N$. La satisfaction de ces contraintes est équivalente à $\mathbf{h}(\mathbf{x}, \psi) - \mathbf{c} = \mathbf{0}$. $\diamond$

Définition 7.2 (Contraintes stochastiques): *Les contraintes stochastiques sont une extension des contraintes déterministes au cas de variables incertaines, où nous considérons que $\mathbf{x} \sim \mathcal{N}(\mathcal{E}\{\mathbf{x}\}, \Lambda_{\mathbf{x}})$, $\psi \sim \mathcal{N}(\mathcal{E}\{\psi\}, \Lambda_{\psi})$ et $\mathbf{c} \sim \mathcal{N}(\mathcal{E}\{\mathbf{c}\}, \Lambda_{\mathbf{c}})$ dans l'équation (7.6). Ainsi, la satisfaction de ces contraintes implique que le résidu $\mathbf{r} = \mathbf{h}(\mathbf{x}, \psi) - \mathbf{c}$ doit être compatible avec l'incertitude présente dans le modèle. $\diamond$*

7.4.1 Propagation de contraintes stochastiques

La prise en compte de contraintes dans un processus d'estimation est un sujet de recherche très vaste, où il existe différentes techniques basées sur l'optimisation sous contraintes [150]. Cependant, la prise en compte de l'incertitude dans les contraintes n'est considérée que dans certaines approches. Par exemple, dans [137], les auteurs proposent une méthodologie de fusion optimale de données sous contraintes en utilisant la programmation quadratique. Dans ce cas, comme les estimées correspondent au minimum d'une fonction de coût, la propagation d'incertitude est obtenue de façon directe.

Par ailleurs, le formalisme du filtre de Kalman a été souvent utilisé pour la prise en compte de contraintes stochastiques [188]. Cependant, les approches proposées diffèrent par la façon dont l'incertitude est considérée. En général, on considère le cas où l'estimée finale est une variable stochastique, mais soumise à une relation exacte de type (7.6) (*i.e.*, $\Lambda_{\psi} = \mathbf{0}$ et $\Lambda_{\mathbf{c}} = \mathbf{0}$). Dans ce cas, la solution la plus simple consiste à appliquer la phase de mise à jour du filtre en considérant

la pseudo-mesure $\mathbf{y} = \mathbf{c}$ de variance $\Lambda_{\mathbf{y}} = \mathbf{0}$. Ainsi, l'estimée obtenue est projetée le plus près du sous-espace des solutions plausibles (*i.e.*, qui satisfont la relation (7.6)). Cette solution permet de corriger la matrice de covariance $\Lambda_{\mathbf{x}}$ de $\mathbf{x}$ de sorte que l'incertitude soit réduite suivant la surface de la contrainte. Elle a été utilisée dans [96] pour la localisation d'objets articulés en utilisant des mesures vidéo. Cependant, comme le font remarquer De Geeter *et al.* [81], appliquer des contraintes avec une variance $\Lambda_{\mathbf{y}} = \mathbf{0}$ peut être dangereux si l'équation (7.6) est fortement non-linéaire ou si la prédiction de $\mathbf{x}$ est loin de sa valeur nominale. Dans ces cas, le filtre peut facilement diverger, si des erreurs importantes se produisent dans la projection. Pour remédier à ce problème, De Geeter *et al.* [81] introduisent une version du filtre de Kalman étendu qui prend en compte l'incertitude des contraintes sous forme d'un modèle de mesure non-linéaire, et rajoutent un facteur d'affaiblissement des contraintes de façon à minimiser le risque de divergence lié à la linéarisation des équations de mesure. Cela peut rendre moins rapide la convergence du filtre, mais elle est plus sûre.

Considérons maintenant les contraintes stochastiques au sens large, cela veut dire avec $\Lambda_{\psi} \neq \mathbf{0}$ et $\Lambda_{\mathbf{c}} \neq \mathbf{0}$. Dans ce cas, la relation $\mathbf{h}$ elle même a une incertitude associée, due à une matrice de covariance Λ_{ψ} non nulle pour ces paramètres. Pour que l'estimée $\mathbf{x}$ respecte ce type de contrainte, nous pouvons appliquer le concept de pseudo-mesure $\mathbf{y} = \mathbf{c}$ lors de la mise à jour de $\mathbf{x}$. Dans ce cas, $\Lambda_{\mathbf{y}}$ est donné par

$$\Lambda_{\mathbf{y}} = \Lambda_{\mathbf{c}} + \nabla_{\psi} \mathbf{h}(\mathbf{x}, \psi) \cdot \Lambda_{\psi} \cdot \nabla_{\psi}^T \mathbf{h}(\mathbf{x}, \psi). \quad (7.7)$$

7.4.2 Une nouvelle approche pour la fusion de structures cartographiques

Nous remarquons que les contraintes présentes dans la cartographie locale sont d'ordre géométrique :

- Le centre $\mathbf{p}$ d'une arête de type bord se trouve sur la droite du demi-plan support ;
- Le centre $\mathbf{p}$ d'une arête de type coin se trouve à l'intersection entre les droites des demi-plans.

Donc, nous pouvons exprimer l'ensemble de contraintes sur le centre $\mathbf{p}$ d'une arête par une relation telle que

$$\mathbf{h}(\mathbf{p}, \psi) = \mathbf{c}, \quad (7.8)$$

où les composantes de ψ seraient données par les paramètres des droites supports. Pour toutes ces structures, dans la carte locale, la relation $\mathbf{h}(\mathbf{p}, \psi) - \mathbf{c} = \mathbf{0}$ est respectée indépendamment de l'incertitude de $\mathbf{p}$, ψ ou $\mathbf{c}$.

Nous présentons dans la section 7.5 des opérations pour la mise à jour de la carte globale à partir de correspondances dans la carte locale. Ces procédures effectuent des modifications sur les structures globales sans prendre en compte des contraintes existant entre elles. Ainsi, à la fin de ces procédures et comme cela a été illustré par l'exemple 7.1, nous n'avons pas la garantie que ces contraintes soient toujours respectées. La propagation des contraintes lors de la mise à jour des structures est nécessaire. Pour réaliser cela, nous avons dû répondre à deux questions primordiales :

“Quelle stratégie choisir pour faire respecter les contraintes ?”

“Quel modèle de contrainte utiliser ?”

Nos réponses à ces questions ont été élaborées dans un compromis entre la facilité d'implantation et la diminution du risque d'inconsistance stochastique des structures. Nous présentons une stratégie d'application des contraintes dans la section suivante.

7.4.2.1 Stratégie d'application de contraintes

Lors des différentes étapes de mise à jour de la carte globale, le nombre de contraintes pour chaque arête n'est pas connu d'avance. Cela est en partie dû aux correspondances multiples, car chaque correspondance contribue à la mise à jour des structures. Une solution serait d'utiliser un graphe de recherche pour déterminer toutes les fusions à réaliser et les contraintes à vérifier, et ainsi de les effectuer en une seule fois dans un processus de filtrage. Les inconvénients de cette approche sont (i) la difficulté d'implantation d'un arbre et d'un filtre de taille variable en fonction du nombre de contraintes, et (ii) une augmentation de la complexité et du temps calcul.

La solution que nous avons choisie découpe le problème en deux phases :

- I. *La phase de fusion sans contraintes*, où l'on réalise d'abord toutes les fusions sans considérer les contraintes ;
- II. *La phase d'application des contraintes*, où l'on applique le modèle des contraintes à toutes les arêtes de type bord ou de type coin qui satisfont au moins une des conditions suivantes :
 - leur centre P a été modifié à cause d'une fusion ;
 - au moins une de leurs droites supports a été l'objet d'une fusion ;

La phase de fusion utilise les techniques présentées dans la section 7.3. Après cette phase, les contraintes à appliquer sont identifiées facilement à partir des structures modifiées.

7.4.2.2 Modèle des contraintes

La garantie de consistance des structures assurée par l'application de filtres linéaires de fusion (*i.e.*, filtre de Kalman ou intersection de covariances) ne servira à rien si une même attention n'est pas apportée à la phase d'application des contraintes. Cette réflexion nous a amené à considérer les points suivants pour la définition du modèle de contraintes :

- Si nous cherchons la satisfaction des contraintes et en même temps nous voulons préserver la consistance des estimées, il faut considérer les incertitudes présentes dans toutes les variables qui constituent les contraintes. En effet, comme nous allons voir dans cette section, tant ψ comme $\mathbf{c}$ sont des variables incertaines, c'est à dire, nous avons $\Lambda_\psi \neq \mathbf{0}$ et $\Lambda_{\mathbf{c}} \neq \mathbf{0}$. Si nous négligeons ses incertitudes, nous prenons le risque d'arriver à un état d'inconsistance stochastique des structures de la carte globale ;
- Une forme linéaire de $\mathbf{h}(\mathbf{p}, \psi)$ permet d'utiliser des filtres linéaires optimaux pour la prise en compte de contraintes stochastiques. Comme illustré par les théorèmes E.2 et E.1, ces filtres ont des conditions à satisfaire pour garantir la consistance lors de leur application : que toutes les mesures d'incertitude du processus soient consistantes ou pessimistes. Or, nous verrons que $\mathbf{p}$ et ψ sont obtenus des structures de la carte globale. Ainsi, si les différents processus

de mise à jour de la carte globale utilisent des techniques performantes de propagation consistante de l'incertitude, $\mathbf{p}$ et ψ sont théoriquement des variables consistantes.

Ainsi, si nous considérons ces points clés, nous avons à faire à des contraintes stochastiques dans la cartographie globale. De plus, nous devons les représenter par un modèle linéaire de sorte que des filtres linéaires optimaux puissent être utilisés pour leur application. Heureusement ce modèle existe et nous le présentons ici sous une forme plus générale.

Considérons que nous avons un point $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$, où $\mathbf{p} = (x, y)^T$, soumis à N contraintes stochastiques, une pour chaque droite contraignante $L_i = \{\mathbf{l}_i, \Lambda_{\mathbf{l}_i}\}$ où $\mathbf{l}_i = (\rho_i, \alpha_i)^T$ pour $i = 1, \dots, N$. Ces contraintes sont utilisées sous la forme de pseudo-mesures par un filtre linéaire optimal qui les intègre simultanément en utilisant le modèle de mesure suivant :

$$\mathbf{y} = \mathbf{H}(\mathbf{l}_1, \dots, \mathbf{l}_N) \cdot \mathbf{p}. \quad (7.9)$$

Après l'application du filtre, nous avons une nouvelle estimée pour le point P qui s'approche plus des droites contraignantes de forme optimale étant donné l'incertitude de ces primitives. Pour déterminer les coefficients de (7.9), nous utilisons la relation suivante, qui relie les coordonnées $\mathbf{p}$ d'un point appartenant à une droite de paramètres $\mathbf{l} = (\rho, \alpha)^T$:

$$\rho = x \cdot \cos(\alpha) + y \cdot \sin(\alpha). \quad (7.10)$$

Ainsi, une forme directe de représentation de N contraintes de ce type est :

$$\mathbf{y} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_N \end{pmatrix} \quad \text{et} \quad \mathbf{H}(\mathbf{l}_1, \dots, \mathbf{l}_N) = \begin{pmatrix} \cos(\alpha_1) & \sin(\alpha_1) \\ \vdots & \vdots \\ \cos(\alpha_N) & \sin(\alpha_N) \end{pmatrix}. \quad (7.11)$$

Lorsque ces N contraintes sont imposées au point P , nous obtenons le point $P_c = \{\mathbf{p}_c, \Lambda_{\mathbf{p}_c}\}$. Les pseudo-mesures sont supposées non-corrélées, *i.e.*,

$$\Lambda_{\mathbf{y}} = \mathbf{Diag}(\sigma_{y_1}^2, \dots, \sigma_{y_N}^2)$$

où $\sigma_{y_i}^2$, $i = 1, \dots, N$, est la variance de l'*i*-ième pseudo-mesure, qui doit incorporer non seulement l'incertitude par rapport à ρ_i , mais aussi celle due à α_i . Cela agit de façon indirecte sur les mesures, et sur les inter-covariances. Ainsi, si nous modélisons l'erreur ε_{y_i} sur y_i par

$$\varepsilon_{y_i} = \rho_i - x \cdot \cos(\alpha_i) - y \cdot \sin(\alpha_i),$$

nous obtenons par propagation de covariances

$$\sigma_{y_i}^2 = \begin{pmatrix} 1 & \sin(\alpha_i) - \cos(\alpha_i) \end{pmatrix} \cdot \Lambda_{\mathbf{l}_i} \cdot \begin{pmatrix} 1 & \sin(\alpha_i) - \cos(\alpha_i) \end{pmatrix}^T. \quad (7.12)$$

Exemple 7.2: Considérons les résultats de la fusion entre demi-plans et arêtes de l'exemple 7.1, où les contraintes entre les arêtes de type bord et les demi-plans n'ont pas été considérés (Figures 7.3(a) et 7.3(b)). Si nous appliquons les contraintes, les résultats respectifs obtenus le filtre de

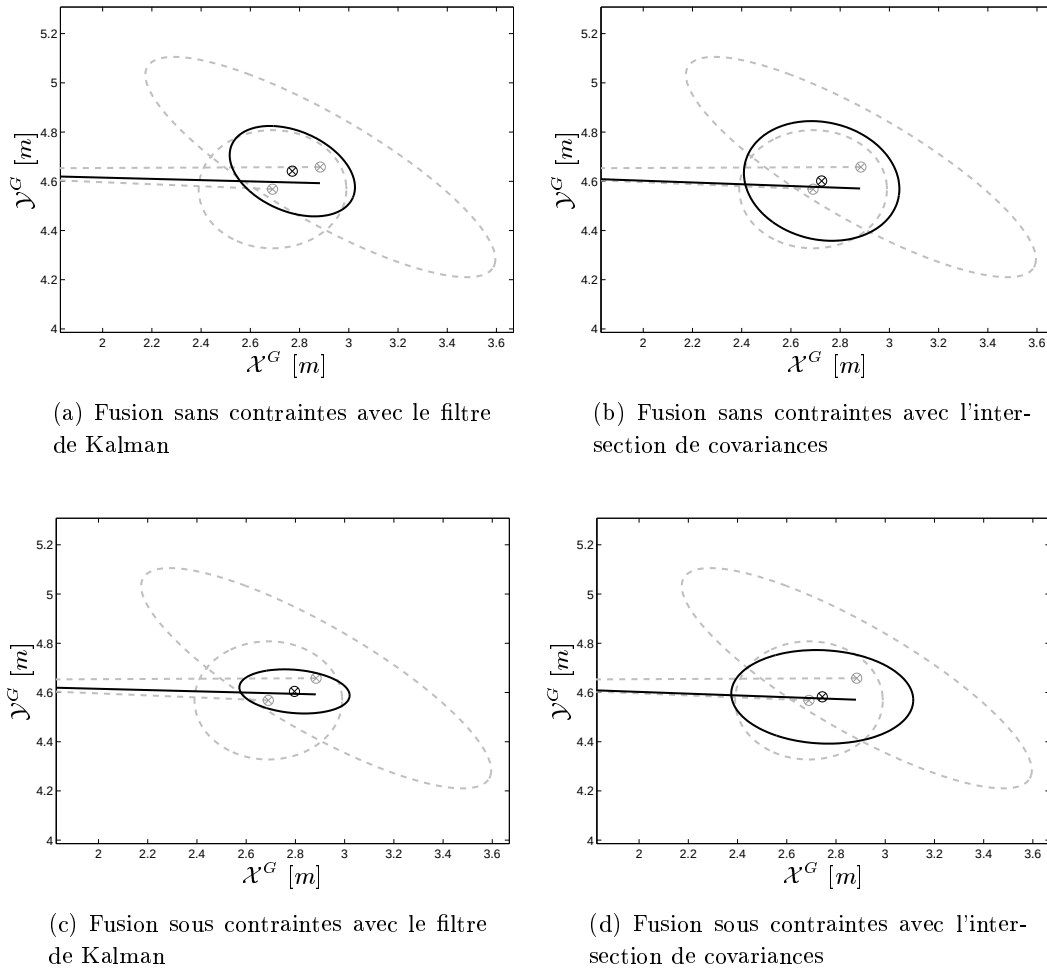


FIG. 7.3 – Exemple de la prise en compte de contraintes dans la fusion de structures.

Kalman et par intersection de covariances sont présentés sur les Figures 7.3(c) et 7.3(d). Nous pouvons vérifier que les bords se sont bien rapprochés des droites supports. Nous remarquons que le filtre de Kalman a réduit l'incertitude de l'arête. Par ailleurs, le filtre intersection de covariances a modifié la forme de l'ellipse d'incertitude, en conservant à peu près son aire. Cela est dû à la façon dont le paramètre ω est calculé dans ce filtre. Il semble que ce paramètre ait été proche d'une des limites de l'intervalle $[\gamma, 1 - \gamma]$ lors de son optimisation. $\diamond$

7.5 Cartographie globale découplée

Les sections précédentes ont présenté des outils de base pour la fusion de structures en utilisant des filtres linéaires optimaux et une stratégie d'application de contraintes stochastiques. Ces outils sont largement utilisés dans les procédures de cartographie découplée présentées dans cette section.

TAB. 7.1 – Listes de correspondances

Liste	Description
$\Omega_{D_i^G} = (\{index_k\}, k = 1, \dots, N_{D_i^G})$	Liste d'informations sur les correspondances locales de D_i^G : l'index $index_k$ de la k-ième correspondance dans la carte locale.
$\Omega_{A_i^G} = (\{index_k, \kappa_k\}, k = 1, \dots, N_{A_i^G})$	Liste d'informations sur les correspondances locales de A_i^G : l'index $index_k$ et le sous-type κ_k de la k-ième correspondance dans la carte locale.
$\Omega_{D_j^R} = (\{index_k\}, k = 1, \dots, N_{D_j^R})$	Liste d'informations sur les correspondances globales de D_j^R : l'index $index_k$ de la k-ième correspondance dans la carte globale.
$\Omega_{A_j^R} = (\{index_k, \kappa_k\}, k = 1, \dots, N_{A_j^R})$	Liste d'informations sur les correspondances globales de A_j^R : l'index $index_k$ et le sous-type κ_k de la k-ième correspondance dans la carte globale.

7.5.1 Mise en correspondance et classification des structures

Dans cette phase, on vérifie la correspondance entre chaque structure de la carte globale et chaque structure de la carte locale. La technique locale de mise en correspondance utilise l'estimée $\hat{Z} = \{\hat{\mathbf{z}}, \mathbf{\Lambda}_{\hat{\mathbf{z}}}\}$ de la position pour relier le repère local et le repère global. Le résultat de la mise en correspondance est une liste établie pour chaque structure et contenant des informations sur ses correspondances (cf. Tableau 7.1).

Une structure de la carte globale qui a au moins une correspondance dans la carte locale est dite “*observée*”. De la même façon, une structure de la carte locale qui a au moins une correspondance dans la carte globale est dite “*supportée*”. Cette classification suit celle adoptée par Leonard et Durrant-Whyte [141].

7.5.2 Inclusion de structures non supportées

7.5.2.1 Inclusion de demi-plans

Les demi-plans locaux D_j^R non supportés sont systématiquement inclus sur la carte globale. Supposons que la carte globale ait déjà N_D^G demi-plans. Alors, un nouveau demi-plan est créé sous l'index $i = N_D^G + 1$ tel que

$$D_i^G = \mathcal{T}^{-1}(D_j^R, \hat{Z}).$$

Des listes de correspondance $\Omega_{D_i^G}$ et $\Omega_{D_j^R}$ sont créées pour garder une trace de cette correspondance :

$$\Omega_{D_i^G} = (\{index_1 = j\}) \quad \text{et} \quad \Omega_{D_j^R} = (\{index_1 = i\}). \quad (7.13)$$

La conséquence directe de cette action est que tout demi-plan de la carte locale aura au moins une correspondance dans la carte globale. Cela est en fait un artifice qui facilite la détermination

des demi-plans supports des arêtes lors de leur inclusion ou de leur mise à jour dans la carte.

7.5.2.2 Inclusion d'arêtes

De la même façon que pour les demi-plans, toute arête locale $A_j^R = \{P_j^R, \kappa_j^R, n_{1,j}^R, n_{2,j}^R\}$ non supportée est incluse dans la carte globale comme une nouvelle arête. Ainsi, supposons que la carte globale ait déjà N_A^G arêtes. Alors, une nouvelle arête $A_i^G = \{P_i^G, \kappa_i^G, n_{1,i}^G, n_{2,i}^G\}$ est créée sous l'index $i = N_A^G + 1$ tel que

$$A_i^G = \mathcal{T}^{-1}(A_j^R, \hat{Z}).$$

Le type de la nouvelle arête A_i^G est le même que celui de l'arête locale $A_j^R : \kappa_i^G := \kappa_j^R$. Les demi-plans supports de cette nouvelle arête dans la carte globale sont ceux qui sont en correspondance avec des demi-plans de supports de l'arête locale A_j^R . On peut donc se poser la question suivante : qu'arrive-t-il si un demi-plan support de l'arête A_j^R n'a pas de correspondance dans la carte globale ? En fait, cela ne peut pas se produire. En effet, comme tout demi-plan local non-supporté est déjà inclus dans la carte globale à l'étape précédente, lors de cette procédure des listes de correspondance ont été automatiquement créées. Comme nous l'avons mentionné, c'est à partir de ces listes que nous pouvons retrouver les index des demi-plans globaux de support de la nouvelle arête. Dans le cas d'un demi-plan support local ayant des correspondants multiples dans la carte globale, nous affectons comme demi-plan support de A_i^G celui ayant l'extrémité la plus proche de cette arête. Bien entendu, ce problème ne se pose pas pour les arêtes photométriques, car elles n'ont pas de demi-plan support.

7.5.3 Fusion des observations

Dans cette étape, toutes les structures de la carte globale observées localement sont mises à jour à partir des correspondances locales. Nous appliquons la stratégie de fusion sous contraintes de la section 7.4.2, qui est divisée en deux phases : *la phase de fusion sans contraintes* et *la phase d'application des contraintes*.

Lors de la phase de fusion sans contraintes, les structures dites *observées* de la carte globale sont fusionnées avec leurs correspondances de la carte locale représentées dans le repère global. Les procédures de fusion sont appliquées de façon à mettre à jour les primitives géométriques des structures globales et les paramètres de visibilité des demi-plans (fusionnés en premier). Pour ces structures, les correspondances multiples sont traitées individuellement comme des mesures séparées. Cela est dû au fait qu'un demi-plan de l'environnement peut être divisé en plusieurs parties (par exemple, à cause de zones d'ombres créées par des objets occultants). La phase d'application des contraintes ne s'applique pas aux demi-plans.

Ensuite, les observations locales sont utilisées pour mettre à jour les paramètres des arêtes. Les correspondances multiples sont traitées de la même façon que pour les demi-plans. Les filtres optimaux de fusion permettent de mettre à jour seulement la position des arêtes. L'actualisation de leurs attributs dépend fortement de l'observation locale, comme cela est illustré par l'exemple suivant :

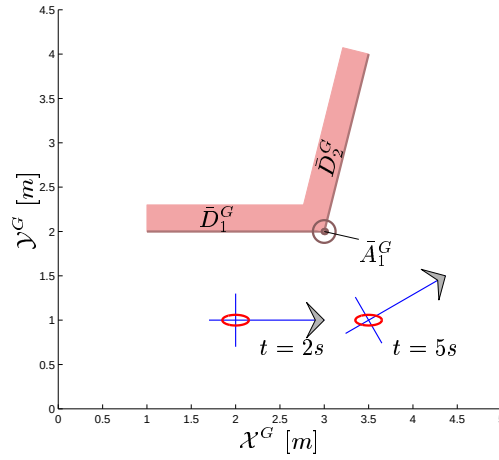


FIG. 7.4 – Environnement de simulation de l'Exemple 7.3

Exemple 7.3: Soit l'environnement de simulation de la Figure 7.4, dans lequel un robot mobile évolue et construit de cartes locales aux instants $t = 2s$ et $t = 5s$. Cet environnement est composé de deux demi-plans $\bar{D}_1^G$ et $\bar{D}_2^G$ et d'une arête $\bar{A}_1^G$ de type coin. Initialement, à $t = 0s$, le robot ne dispose d'aucune représentation de cet environnement, et la carte globale est vide. Nous nous intéressons à une progression du sous-type de l'arête globale correspondante à $\bar{A}_1^G$ lors de la construction incrémentale de la carte. La Figure 7.5 illustre l'état de la carte globale avant et après la fusion avec les structures locales à $t = 2s$ et à $t = 5s$. Nous pouvons voir à $t = 2s$ que toutes les structures locales sont systématiquement ajoutées à la carte globale. Nous remarquons qu'étant donné la position du robot, le coin $\bar{A}_1^G$ de l'environnement est observé comme une arête A_1^R de type bord parce que seulement un de ses demi-plans supports est détecté par le système sensoriel. Ainsi, la carte globale mise à jour contient un demi-plan D_1^G et une arête A_1^G de type bord. Ce n'est qu'à l'instant $t = 5s$ que la carte locale a capturé l'arête de l'environnement sous la forme d'un coin A_1^R . La carte globale représente cette même structure physique sous forme d'un bord A_1^G . Ces deux arêtes étant en correspondance, elles sont fusionnées en utilisant un filtre linéaire optimal, ce qui met à jour seulement la position de l'arête globale A_1^G . Cependant, il nous reste encore à définir un mécanisme de décision sur le sous-type de l'arête A_1^G . Un tel mécanisme est indispensable pour la modélisation de structures physiques qui se traduisent sous forme d'arêtes dans la cartographie. En ne prenant pas en compte les contraintes imposées par des observations plus récentes, le risque d'inconsistance géométrique de la carte globale peut augmenter. Par exemple, supposons que l'arête globale A_1^G présente une éventuelle erreur sur sa position le long du demi-plan support D_1^G . Suivant notre stratégie de fusion sous contraintes, si cette arête reste de type bord associé à D_1^G lors de sa mise à jour, une seule contrainte sera appliquée pour corriger sa position. Dans ce cas, il est évident que cette contrainte ne va pas contribuer davantage à réduire l'erreur de position de A_1^G dans la direction de son demi-plan support. Une solution à ce problème consiste à faire plus confiance aux mesures locales, en l'occurrence A_1^R , et à associer le type coin à l'arête A_1^G lors de sa mise à jour. En même temps, le demi-plan support D_2^R est ajouté à la carte globale comme une

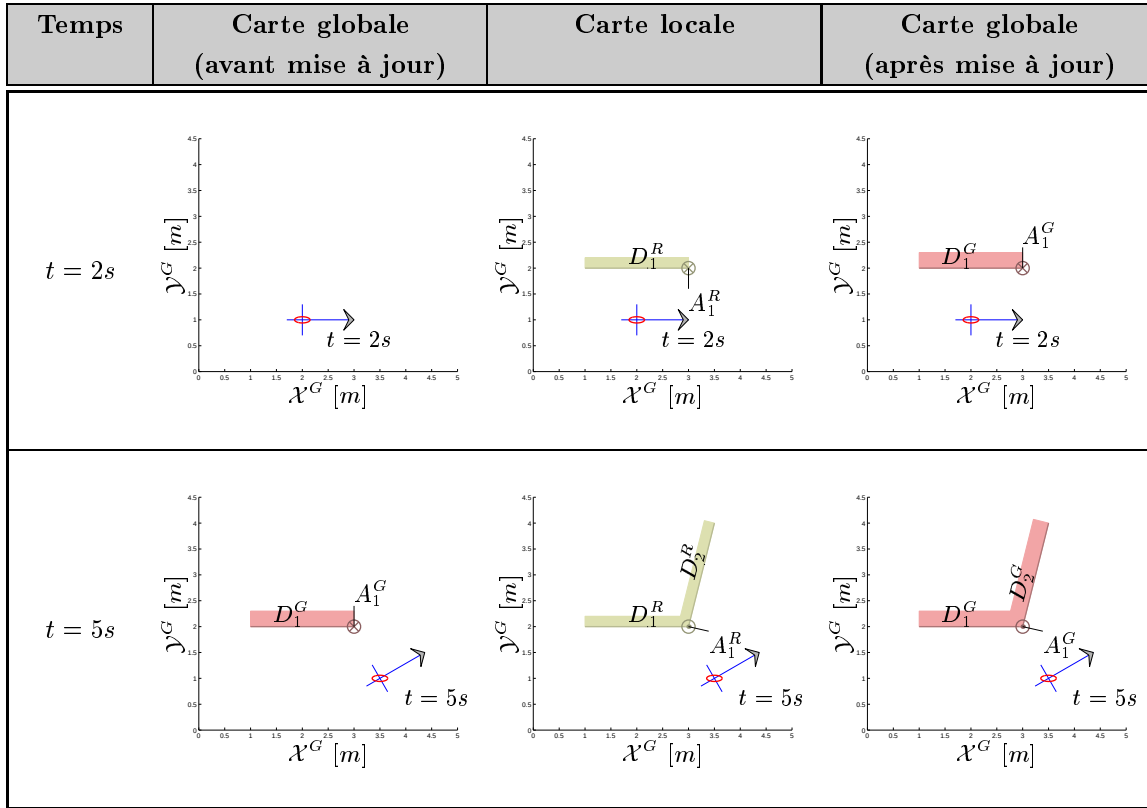


FIG. 7.5 – Exemple de l'évolution des arêtes en fonction de leur observation locale.

nouvelle structure D_2^G . Ainsi, D_2^G et D_1^G sont les demi-plans supports de A_1^G . En conséquence, l'erreur sur A_1^G est réduite dans les directions orthogonales à D_1^G et D_2^G lors de la phase d'application de contraintes. $\diamond$

Cet exemple illustre bien le fait qu'une gestion appropriée du sous-type des arêtes lors de la fusion des observations est indispensable. En plus, il est nécessaire privilégier la prise en compte de contraintes correspondant à des observations plus récentes. C'est dans ce sens que nous proposons un mécanisme de gestion relativement simple. Ce mécanisme établit que le changement du sous-type d'une arête globale en fonction de son observation locale doit suivre le diagramme de transition de la Figure 7.6. Dans ce diagramme, les cercles correspondent aux sous-types possibles d'une arête globale : **P** représente une arête photométrique, **B** un bord et **C** un coin. Les arcs orientés représentent le changement de type d'une arête globale, représentée par le cercle d'origine, dans un nouveau type représenté par le cercle de destination. Les étiquettes associées aux arcs correspondent au type de la correspondance dans la carte locale. Ce diagramme montre que nous n'effectuons pas la fusion d'arêtes globales de type coin avec des arêtes locales de type photométrique. En réalité, nous appliquons la démarche adoptée pour la cartographie locale, où les mesures vidéo ne sont pas utilisées pour la mise à jour des coins de la carte globale. Pour le même motif, la position de ces structures est déjà assez bien définie par ses deux demi-plans supports.

Après fusion d'une arête globale avec une correspondance locale, l'arête globale peut changer de

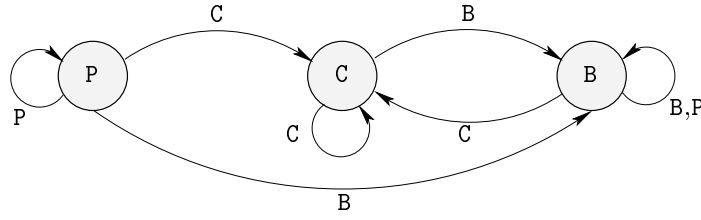


FIG. 7.6 – Diagramme de transition du type d'une arête globale (cercles) lors de la fusion avec des observations locales (arcs).

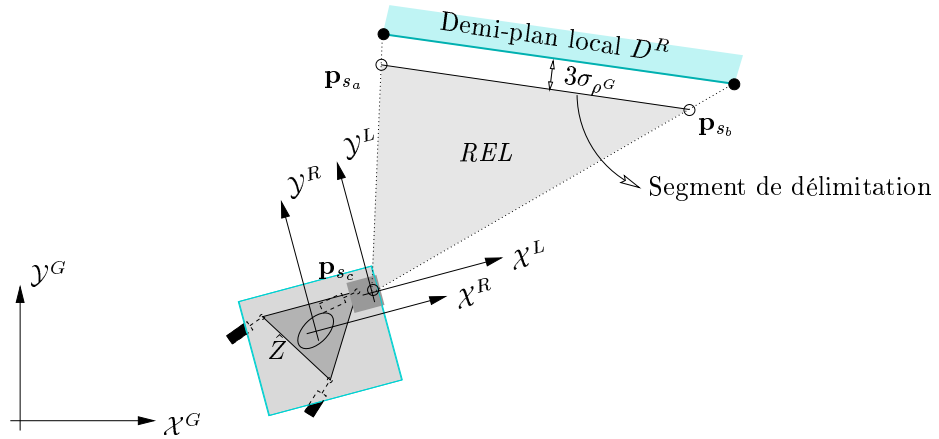


FIG. 7.7 – Détermination de la région d'espace libre (REL) associée à un demi-plan local.

demi-plans supports. En fait, les nouveaux demi-plans supports de l'arête globale sont déterminés de façon indirecte, parmi les correspondances globales des demi-plans locaux supportant l'arête locale (voir la procédure d'attribution des demi-plans supports des nouvelles arêtes dans la section 7.5.2.2).

7.5.4 Conservation de l'espace libre

Dans notre approche, l'effacement des structures de la carte globale se fait de sorte que l'espace libre correspondant aux demi-plans locaux soit respecté. Il s'appuie sur le fait que, vu du système multicauteur, aucun demi-plan de la carte locale n'est occulté par un autre de la même carte. Pour chaque demi-plan local D^R , nous déterminons une région d'espace libre (REL) correspondante. La REL correspond à un triangle, à l'intérieur duquel aucune structure globale n'est observable (en considérant les incertitudes de mesure). Ainsi, il suffit d'éliminer de la carte globale toutes les structures se trouvant à l'intérieur de chaque REL en utilisant des règles de maintenance de l'espace libre.

Commençons par la détermination de la REL pour un demi-plan local D^R (Figure 7.7). Cette région est délimitée par trois points $\mathbf{p}_{sa}$, $\mathbf{p}_{sb}$ et $\mathbf{p}_{sc}$, déterminés de façon à prendre en compte les incertitudes du demi-plan local D^R et de l'estimée $\hat{Z}$ de la position du robot. $\mathbf{p}_{sc}$ correspond au

centre d'observation du système multicapteur. Les sommets $\mathbf{p}_{sa}$ et $\mathbf{p}_{sb}$ sont les projections des extrémités du demi-plan local sur un *segment de délimitation*, suivant leur direction d'observation. Le *segment de délimitation* est déterminé en fonction de l'intervalle 3σ du demi-plan D^G , qui correspond à la représentation de D^R dans le repère global, *i.e.*, $D^G = \mathcal{T}^{-1}(D^R, \hat{Z})$. Il correspond à un cas extrême du positionnement réel de D^G dans le repère global. Soit $\mathbf{l}^G = (\rho^G, \alpha^G)^T$ les paramètres polaires de la droite du demi-plan D^G , et σ_{ρ^G} l'écart-type de ρ^G . En considérant l'intervalle de confiance 3σ sur le paramètre ρ^G , *i.e.*, $[\rho^G - 3 \cdot \sigma_{\rho^G}, \rho^G + 3 \cdot \sigma_{\rho^G}]$, nous choisissons une des limites de cet intervalle pour délimiter la zone d'espace libre. En effet, nous déterminons le *segment de délimitation* comme celui parallèle au segment de D^G , mais qui se trouve à une distance $3 \cdot \sigma_{\rho^G}$ plus proche du robot que D^G . Les points $\mathbf{p}_{sa}$ et $\mathbf{p}_{sb}$ sont les intersections entre ce segment de délimitation et les lignes d'observation des extrémités de D^G . Ainsi, nous pouvons affirmer que, au moins dans le sens des probabilités, aucune structure cartographique n'est observable à l'intérieur du triangle délimité par les trois points $\mathbf{p}_{sa}$, $\mathbf{p}_{sb}$ et $\mathbf{p}_{sc}$ (voir Figure 7.7).

L'exemple suivant illustre l'application de cette procédure de maintenance de l'espace libre dans la cartographie.

Exemple 7.4: *Considérons trois cas de configurations distinctes (a), (b) et (c), illustrés sur la Figure 7.8. Nous avons représenté à gauche les structures globales et l'espace libre déterminé à partir du demi-plan D_1^R de la carte locale. La carte globale est initialement composée d'un demi-plan D_1^G et des deux bords associés A_1^G et A_2^G . Les demi-plans global et local ne sont pas en correspondance. Les figures de droite montrent l'état des structures de la carte globale après application de la procédure.* $\diamond$

Remarque 7.1 (Elimination de structures): *L'élimination d'arêtes de la carte globale se fait sans altération des demi-plans supports qui leur sont associés. Néanmoins, lorsqu'un demi-plan est éliminé, cela implique l'élimination ou le changement de type des arêtes associées, c'est à dire :*

- *Toutes les arêtes de type bord associées au demi-plan sont aussi éliminées ;*
- *Toutes les arêtes de type coin associées au demi-plan deviennent arêtes de type bord, tout en conservant l'autre demi-plan support ;* $\diamond$

7.5.5 Fusion de ressemblances

La mise à jour de la carte globale décrite dans le paragraphe précédent conduit parfois à faire apparaître dans cette carte plusieurs structures géométriques qui représentent le même élément physique de l'environnement. L'objectif de cette dernière étape est de réaliser la fusion des structures semblables, afin de simplifier la carte et d'alléger les traitements suivants (par exemple d'éviter des mises en correspondance multiples).

La fusion de ressemblances est réalisée de façon itérative, en commençant par celles des demi-plans. La détection des ressemblances est équivalente à une mise en correspondance carte globale / carte globale, sachant qu'une structure ne peut pas être en correspondance avec elle-même. Une fois les correspondances déterminées conformément aux critères de l'annexe D.3, nous considérons que deux demi-plans D_i^G et D_j^G tels que $i \neq j$ sont ressemblants si :

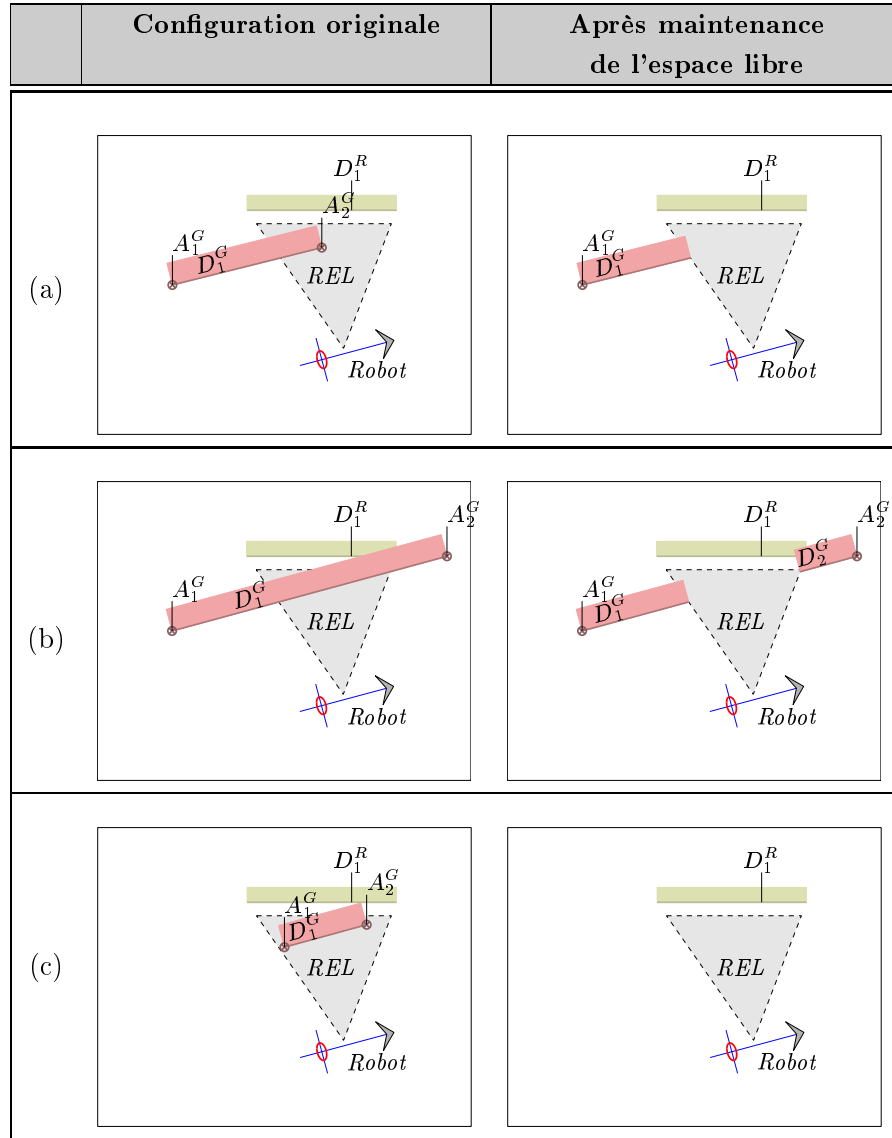


FIG. 7.8 – Exemples d'application de l'étape de maintenance de l'espace libre.

- I. $D_{\chi^2}(D_i^G, D_j^G) < D_{\chi^2}(D_i^G, D_k^G)$, pour $k \neq j$;
- II. $D_{\chi^2}(D_i^G, D_j^G) < D_{\chi^2}(D_l^G, D_j^G)$, pour $l \neq i$.

Deux arêtes A_i^G et A_j^G telles que $i \neq j$ sont ressemblantes si :

- I. $D_{\chi^2}(A_i^G, A_j^G) < D_{\chi^2}(A_i^G, A_k^G)$, pour $k \neq j$;
- II. $D_{\chi^2}(A_i^G, A_j^G) < D_{\chi^2}(A_l^G, A_j^G)$, pour $l \neq i$;
- III. A_i^G et A_j^G sont du même sous-type et ont les mêmes demi-plans supports.

Une fois les ressemblances détectées pour l'itération courante, elles sont fusionnées et le résultat de la fusion remplace les structures d'origine dans la carte globale. Ce processus est réitéré jusqu'à

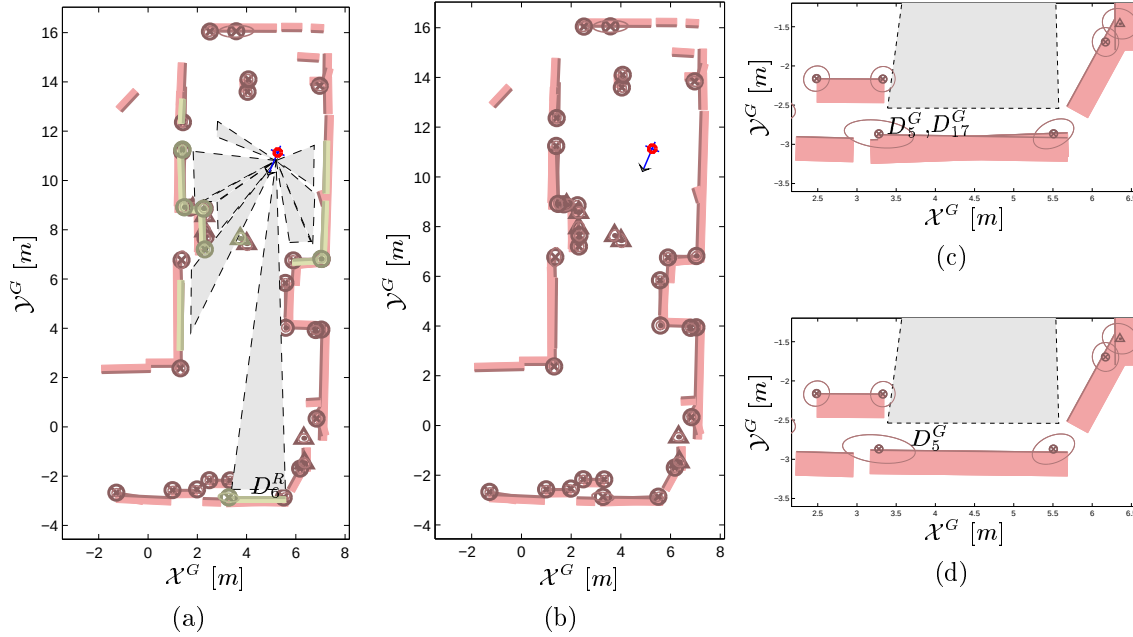


FIG. 7.9 – Exemple de la fusion de ressemblances : (a) configuration originale, (b) carte globale mise à jour sans fusion de ressemblances, (c) structures ressemblantes et (d) leur fusion

n'avoir plus aucune ressemblance. Ensuite, les contraintes sont appliquées aux arêtes qui ont été changées ou dont un demi-plan support a été mis à jour.

Exemple 7.5: *Considérons la situation réelle de la Figure 7.9(a), où sont illustrées la carte globale avant sa mise à jour, une carte locale (en superposition), et les régions d'espace libre. Portons notre attention sur le demi-plan local D_6^R , situé au bas de cette figure, et qui a deux correspondances globales. La carte globale mise à jour sans fusion des ressemblances est montrée sur la Figure 7.9(b). A première vue, nous ne remarquons rien d'anormal sur cette carte. Cependant, en la regardant de plus près sur la Figure 7.9(c), nous voyons que les deux correspondances globales D_5^G et D_{17}^G de D_6^R se sont rapprochées. Ce rapprochement qui est une conséquence de leur mise à jour indépendante par la mesure locale D_6^R , est suffisant pour qualifier ces demi-plans comme ressemblants. Ainsi, dans l'étape de fusion de ressemblances, D_5^G et D_{17}^G sont fusionnés. Le résultat remplace D_5^G , et D_{17}^G est effacé de la carte globale (voir Figure 7.9(d)).* $\diamond$

7.6 Etudes en simulation

A travers les exemples 7.1 et 7.2 nous avons illustré l'utilisation d'approches de fusion de structures cartographiques avec et sans contraintes. Les résultats obtenus semblent qualitativement satisfaisants, malgré la grande incertitude présentée par le filtre intersection de covariances. Cependant, ces résultats ne nous apportent pas suffisamment d'éléments permettant de choisir la

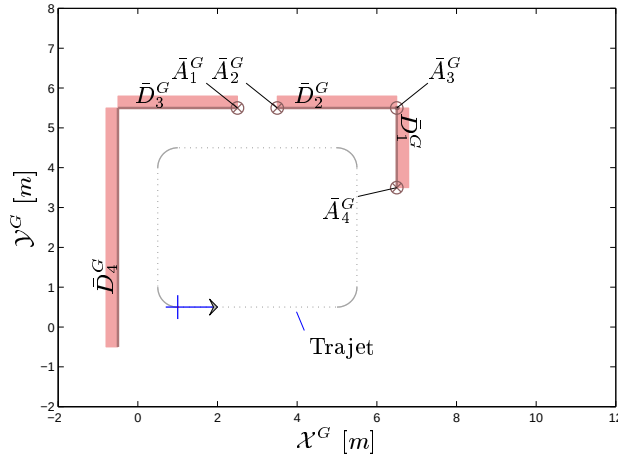


FIG. 7.10 – Environnement de simulation

meilleure méthode de filtrage pour la cartographie globale découplée (filtre de Kalman ou intersection de covariances).

Afin d'évaluer de manière plus générale l'apport de la propagation des contraintes dans la cohérence et la stabilité de la carte, nous proposons, dans les sections qui suivent, une comparaison en simulation des deux méthodes de filtrage optimal pour la cartographie découplée, avec puis sans propagation des contraintes.

7.6.1 Environnement de simulation

Considérons un robot mobile navigant suivant un trajet fermé dans l'environnement de la Figure 7.10. La carte de l'environnement est composée de quatre demi-plans $\bar{D}_1^G$, $\bar{D}_2^G$, $\bar{D}_3^G$ et $\bar{D}_4^G$, et de quatre arêtes $\bar{A}_1^G$, $\bar{A}_2^G$, $\bar{A}_3^G$ et $\bar{A}_4^G$. Durant son trajet, le robot effectue 130 cycles de construction et mise à jour de la carte globale. Nous simulons l'estimée $\hat{Z}_k = \{\hat{\mathbf{z}}_k, \mathbf{\Lambda}_{\hat{\mathbf{z}}_k}\}$ de la position du robot lors du k -ième cycle de localisation en utilisant le modèle de perturbation suivant :

$$\hat{\mathbf{z}}_k = \mathbf{z}_k + \mathbf{w}_k \quad \mathbf{\Lambda}_{\hat{\mathbf{z}}_k} = \mathbf{\Lambda}_{\mathbf{w}_k} \quad (7.14)$$

tel que $\mathbf{z}_k$ est la position réelle du robot, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_{\mathbf{w}_k})$, et

$$\mathbf{\Lambda}_{\mathbf{w}_k} = \mathbf{Diag}((5 \text{ cm})^2, (5 \text{ cm})^2, (0, 5^\circ)^2) .$$

La carte locale utilisée à chaque cycle est constituée de toutes les structures de l'environnement réel, représentées dans le repère robot à la position $\mathbf{z}_k$. Les primitives géométriques de cette carte ont été perturbées par des bruits gaussiens $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_{\mathbf{v}_k})$, avec $\mathbf{\Lambda}_{\mathbf{v}_k} = \mathbf{Diag}((4 \text{ cm})^2, (0, 2^\circ)^2)$ pour les droites et $\mathbf{\Lambda}_{\mathbf{v}_k} = \mathbf{Diag}((4 \text{ cm})^2, (4 \text{ cm})^2)$ pour les points. Les matrices de covariance des primitives sont les mêmes que celles de la perturbation. Ainsi, nous pouvons affirmer que l'estimée de position et les structures de la carte locale sont consistantes et sans biais.

7.6.2 Résultats de la cartographie découplée sous contraintes

La Figure 7.11 présente les résultats de simulation de la cartographie globale sous contraintes. Nous avons sur 7.11(a) et 7.11(b) les cartes globales obtenues à la fin de la simulation, en utilisant respectivement l'intersection de covariances et le filtre de Kalman dans la fusion de structures. Les deux cartes présentent le même nombre de structures, et à cette échelle nous ne remarquons aucune différence par rapport à l'environnement de simulation (cf. Figure 7.10). Pour analyser qualitativement ces résultats, nous avons réalisé, à chaque cycle de cartographie, une vérification statistique de la consistance entre les structures de l'environnement et leurs ressemblances dans la carte globale, de la façon suivante :

Pour chaque structure de la carte, nous cherchons les correspondances dans la carte globale en considérant un seuil très élevé pour la distance de Mahalanobis. Ce seuil étant fixé à 20, il nous permet aussi de capturer les inconsistances résultant soit d'un biais d'estimation, soit d'une sous-estimation de l'incertitude des structures. Nous considérons comme structure ressemblante la structure correspondante ayant la plus petite distance de Mahalanobis.

Les Figures 7.11(c) et 7.11(e) montrent l'évolution de la consistance d'estimation des différentes structures de la carte lors de sa mise à jour, en utilisant le filtre intersection de covariances. Cette consistance est mesurée par la distance de Mahalanobis entre chaque structure de la carte et sa correspondante dans la carte globale en construction. Comme ces mesures suivent une distribution χ^2_2 , nous considérons comme inconsistantes les valeurs en dehors de l'intervalle de confiance à 95 %. Ainsi, on voit que, durant toute la simulation, les cas d'inconsistance sont rares. Par contre, le filtre de Kalman présente un plus grand nombre de cas d'inconsistance, comme nous pouvons le voir le voir sur les Figures 7.11(d) et 7.11(f). Et pourtant, la carte globale obtenue ressemble bien à la carte d'environnement de la Figure 7.10. Cela nous amène à penser que la source d'inconsistance est une sous-estimation des matrices de covariance des structures, notamment pour les arêtes (cf. Figure 7.11(f)). En effet, en examinant de plus près une de ces arêtes, disons $\bar{A}_2^G$ de centre $\bar{\mathbf{p}} = \begin{pmatrix} 3,50 \text{ m} & 5,50 \text{ m} \end{pmatrix}^T$ et son estimée A_2^G , nous obtenons les mesures suivantes pour chaque approche :

Grandeur	Intersection de covariances	Filtre de Kalman
$\ \bar{\mathbf{p}} - \mathbf{p}\ $	6,41 cm	0,27 cm
$D_{\chi^2}(\bar{A}_2^G, A_2^G)$	1,34	14,80

Nous observons que la précision sur l'estimation du centre de l'arête $\bar{A}_2^G$ obtenue par le filtre de Kalman a été plus importante que celle obtenue par intersection de covariances. Cependant, comme nous l'avons vu, la distance de Mahalanobis indique que le filtre de Kalman a présenté des résultats inconsistants.

Nous avons utilisé comme argument en faveur des filtres linéaires optimaux le fait qu'ils sont capables de fournir des estimées consistantes lorsque la prédiction et l'observation sont consistantes. Ces conditions semblent avoir été réunies dans cette simulation, tant pour la position du robot,

que pour les structures de la carte locale. Nous considérons que les inconsistances observées sont dues à d'autres facteurs, qui font l'objet des remarques suivantes :

Remarque 7.2: *Le modèle d'application de contraintes n'incorpore pas une modélisation du bruit de processus associé à la prédiction de l'arête. En effet, si nous avons un tel modèle, le filtre de Kalman présenterait une estimation moins optimiste lors de la fusion sous contraintes dans l'Exemple 7.2.* $\diamond$

Remarque 7.3: *Comme le fait remarquer [105], en réalité des inter-corrélations existent entre les estimées des structures cartographiques. Au fur et à mesure que le robot se déplace les structures de la carte se rendent encore plus corrélées. Notre approche découplée ne préserve pas des estimées des inter-covariances entre les structures de la carte. Nous savons que le filtre intersection de covariances présente une estimation consistante quand la prédiction et l'observation le sont aussi, et cela même en présence d'inter-covariances non modélisées. Hélas, cela n'est pas le cas du Filtre de Kalman. Ainsi, nous pensons que les résultats d'inconsistance présentés par ce filtre sur la Figure 7.11(f) sont dus à la présence d'inter-covariances inconnues entre les arêtes et les demi-plans supports.* $\diamond$

Cette dernière remarque nous mené à choisir l'intersection de covariance comme filtre optimal pour la fusion et la maintenance de contraintes dans la processus de cartographie. Ses performances en présence d'inter-covariances non-modélisées en font un outil approprié à la cartographie découplée.

7.7 Résultats expérimentaux

Dans cette section, nous présentons deux expériences ayant comme objectif d'évaluer l'approche de cartographie proposée dans des sites réels différents. Ces sites sont *Premier étage extension - B* et *Halle mécatronique*, dont les données capteur sont présentées respectivement dans les annexes G.7 et G.10.

7.7.1 Cartographie du site *Premier étage extension - B*

Cette expérience a pour but de vérifier les caractéristiques de convergence à court terme de notre approche de cartographie en utilisant les mesures du site *Premier étage extension - B*. Ces mêmes données ont été déjà utilisées pour évaluer les approches de localisation dans la section 6.5.1. Nous souhaitons évaluer différentes associations de méthodes de localisation et de cartographie, en comparant les cartes construites à une carte de référence représentant l'environnement réel. En fait, comme nous ne disposons pas de modèle CAO du site cartographié, notre carte de référence (Figure 7.12) a été construite en utilisant la gyrodométrie comme technique de localisation. Elle permet une comparaison au moins visuelle des résultats.

Dans cette carte, l'espace libre parcouru par le robot représente un circuit fermé sur lequel deux positions particulières sont indiquées : la position 1 de départ, et la position 40, juste avant

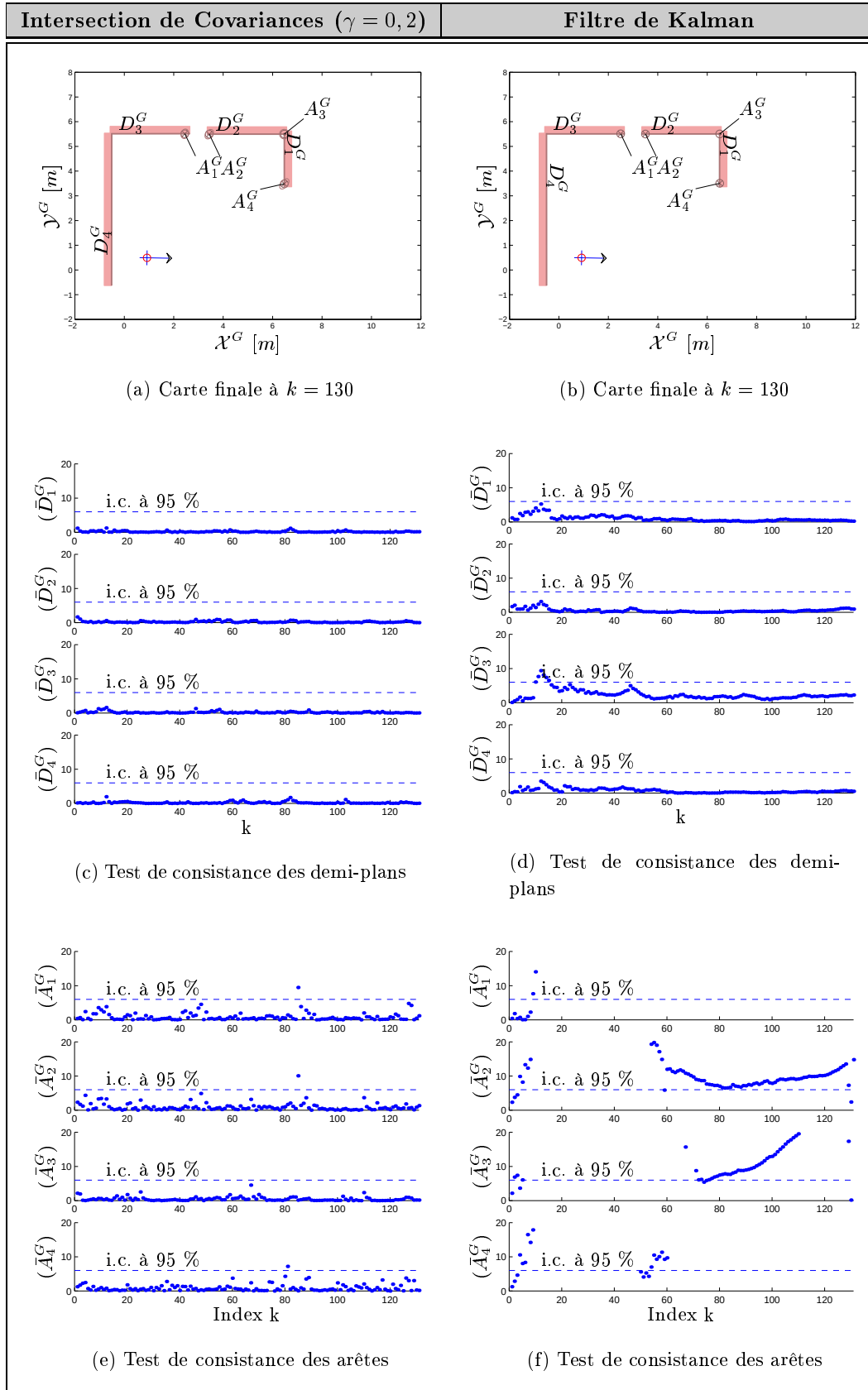
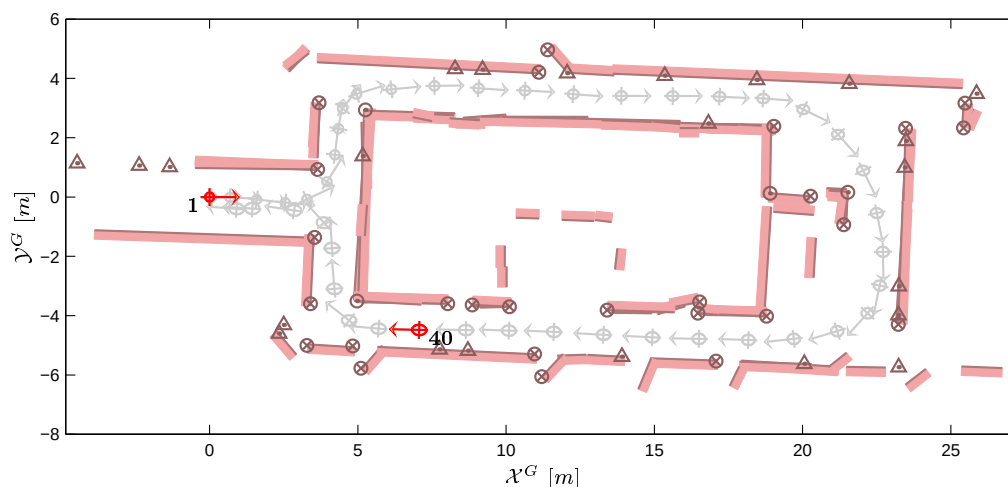


FIG. 7.11 – Résultats de simulation de la construction de cartes globales sous contraintes

FIG. 7.12 – Carte globale du site *Premier étage extension - B*.

la fermeture du circuit. Notre évaluation consiste à vérifier l'état de la carte globale, c'est à dire sa convergence, lorsque le robot arrive à cette position.

Nous présentons les résultats obtenus avec les associations de différentes méthodes de localisation et de cartographie. En ce qui concerne la localisation, nous combinons des approches différentes d'estime et de recalage. Pour la cartographie, nous évaluons les apports de l'application de contraintes stochastiques et ceux de la propagation d'incertitude par échantillonnage sélectif.

Les résultats obtenus sont présentés sur la Figure 7.13. Nous avons à gauche les cartes globales obtenues sans prise en compte de contraintes stochastiques, et avec propagation d'incertitude par linéarisation. A droite, les résultats produits par l'application de contraintes stochastiques et la propagation d'incertitude par échantillonnage sélectif semblent être meilleurs que les précédents. Cependant, le succès de la cartographie dépend beaucoup de la performance de la localisation. Nous avons vu, dans la section 7.2, qu'une des sources d'erreurs pour la cartographie est la propagation de biais provenant de la localisation. Avec une approche de localisation peu performante, ce biais d'estimation peut être important. C'est le cas sur la Figure 7.13(a) où l'odométrie est utilisée comme estime et où un *FKE* séquentiel réalise le recalage. C'est dans des cas comme celui-ci que nous remarquons l'importance des contraintes et de l'échantillonnage sélectif.

On obtient de meilleurs résultats avec une localisation utilisant un M-estimateur (μ -Huber) pour le recalage et l'odométrie pour l'estime (Figure 7.13(b)). Cependant, un biais réduit a produit une légère courbure dans la carte. On remarque aussi un cas d'incohérence pour un demi-plan, indiqué par une flèche dans la carte à gauche de la Figure 7.13(b). Etant donné l'incertitude importante de la localisation lors du passage du robot par cette région, nous pensons que cette incohérence résulte d'une propagation incorrecte de cette incertitude par linéarisation (cf. exemple B.2). Cette incohérence ne s'est pas produite avec la propagation d'incertitude par échantillonnage sélectif, comme nous pouvons le voir sur la carte à droite de cette figure.

Enfin, les Figures 7.13(c)-(d) présentent les cartes globales obtenues en utilisant la gyrodométrie comme technique d'estime. Les résultats obtenus sont satisfaisants, que ce soit en utilisant le *FKE*

séquentiel, ou le M-estimateur comme méthode de recalage.

7.7.2 Cartographie du site *Halle mécatronique*

Les expériences précédentes ont permis de tester les performances de nos algorithmes de cartographie avec des trajectoires de courte durée. Dans l'expérience qui suit, nous souhaitons évaluer la qualité des résultats obtenus avec une trajectoire plus longue, lorsque l'estime est guidée par gyrodométrie.

Les données acquises correspondent à un trajet aller-retour de 133,18 m effectué dans le site *Halle mécatronique* constitué d'une halle et d'un long couloir. La durée de la trajectoire (323,5 s) est suffisante pour que l'incertitude de la gyrodométrie soit importante à la fin du parcours. La Figure 7.14 présente la carte globale du site obtenue en utilisant la gyrodométrie pour l'estime et le M-estimateur (μ -Huber) pour le recalage. Cette carte est apparemment conforme au site.

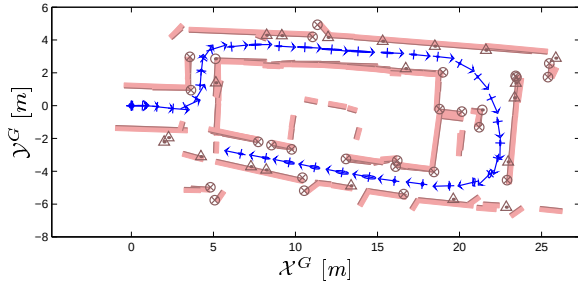
Cependant, la convergence obtenue sur cette carte n'est pas seulement due à la gyrodométrie. En effet, des problèmes d'incohérence ont été observés avec les autres méthodes de recalage, notamment dans la zone d'intérêt délimitée sur la Figure 7.14. Cette zone est amplifiée sur la Figure 7.15, où nous présentons des résultats obtenus avec les différents algorithmes de recalage évalués dans la section précédente. En utilisant le *FKE* séquentiel, nous observons les incohérences indiquées par les flèches sur les cartes de la Figure 7.15(a). Sans contraintes et en propageant l'incertitude par linéarisation (figure de gauche), nous obtenons un mur représenté par un nombre trop important de demi-plans suffisamment espacés pour ne pas être fusionnés. Nous remarquons qu'en prenant en compte des contraintes et avec l'échantillonnage sélectif, cet écartement est beaucoup moins important (carte globale à droite de la Figure 7.15(a)). Néanmoins, la flèche sur cette carte indique encore que la convergence n'est pas encore complète, car il persiste un certain nombre d'arêtes sensées correspondre à un seul coin de l'environnement. Cependant, nous remarquons sur la Figure 7.15(b) qu'en utilisant le M-estimateur pour le recalage, ces problèmes n'ont pas lieu. Ce résultat illustre bien le besoin d'une localisation performante pour assurer la convergence de la cartographie (au moins à moyen terme). Enfin, un autre avantage de cette approche découplée est son temps de calcul réduit. En effet, pour cette expérience, il a atteint au maximum 135 ms par cycle de mise à jour (mise en correspondance comprise).

7.8 Conclusion

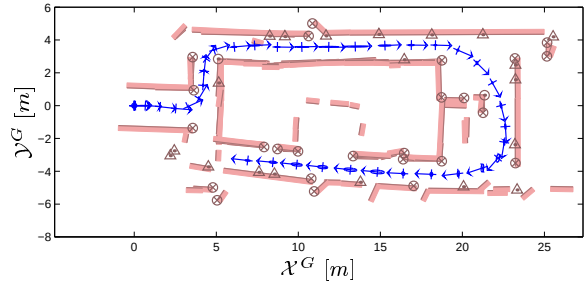
Nous avons présenté dans ce chapitre une approche découplée pour la cartographie d'environnements et la localisation concurrentes. La localisation et la cartographie sont traitées séparément. Cette approche est fondée sur des techniques récentes qui sont très performantes dans certaines conditions d'utilisation. Nous avons aussi incorporé des contraintes stochastiques qui réduisent l'effet du biais provoqué par une localisation peu performante. Des résultats expérimentaux ont illustré la bonne qualité des cartes géométriques obtenues pour des environnements de niveaux de difficulté différents. Ces résultats associés à ceux du chapitre 6 montrent bien la robustesse de notre approche de localisation et cartographie découplées.

Sans contraintes / Propagation d'incertitude par linéarisation

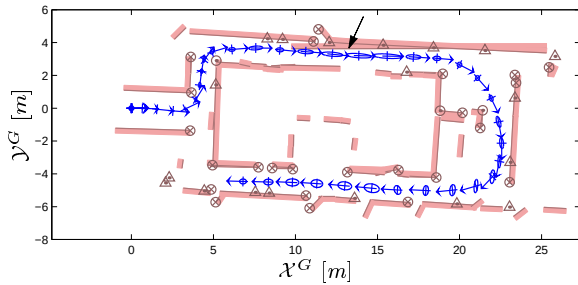
Avec contraintes / Propagation d'incertitude par échantillonnage sélectif



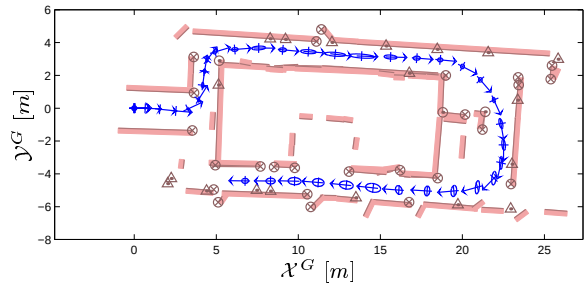
(a) Estime par odométrie et recalage par *FKE* séquentiel



(b) Estime par odométrie et recalage par M-estimateur (μ -Huber)

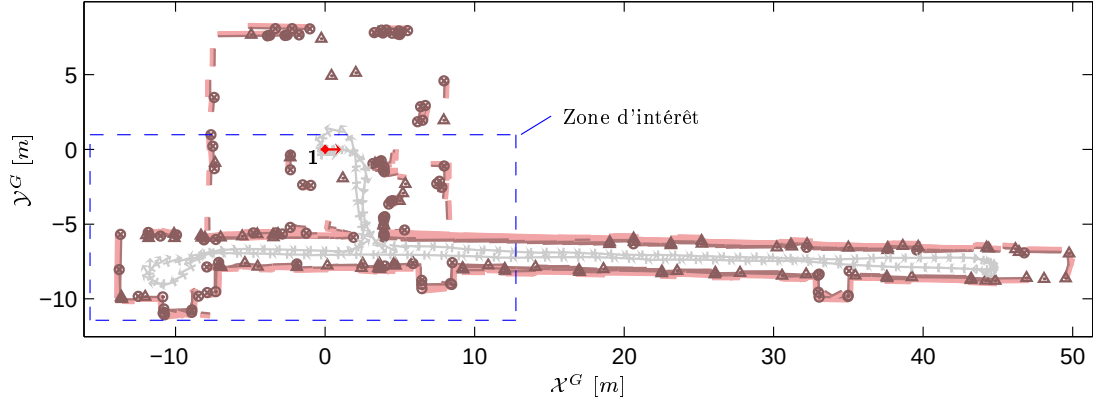


(c) Estime par gyrodométrie et recalage par *FKE* séquentiel



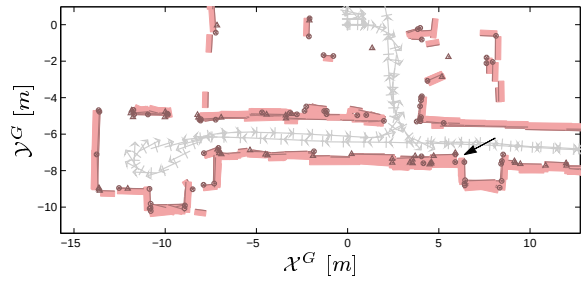
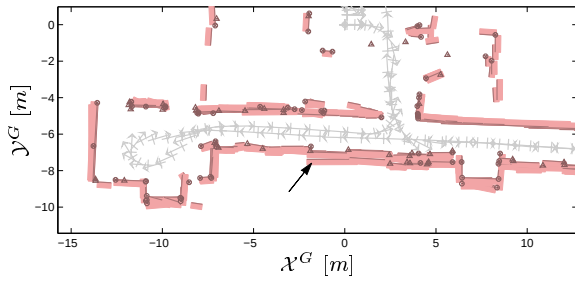
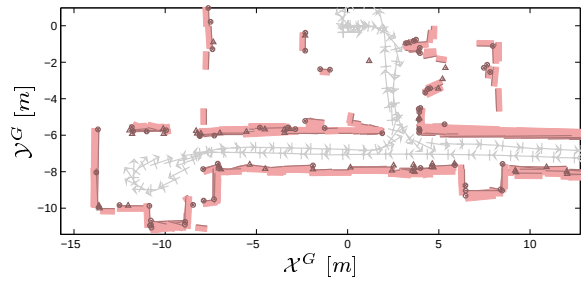
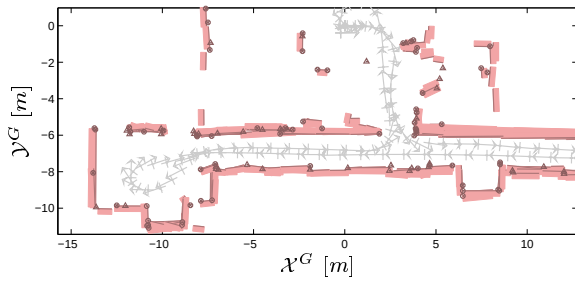
(d) Estime par gyrodométrie et recalage par M-estimateur (μ -Huber)

FIG. 7.13 – Résultats expérimentaux obtenus pour la cartographie du site *Premier étage extension* - B.

FIG. 7.14 – Carte globale du site *Halle mécatronique*.

Sans contraintes / Propagation d'incertitude par linéarisation

Avec contraintes / Propagation d'incertitude par échantillonnage sélectif

(c) Estime par gyrométrie et recalage par *FKE* séquentiel(d) Estime par gyrométrie et recalage par M-estimateur (μ -Huber)FIG. 7.15 – Résultats expérimentaux obtenus pour la cartographie du site *Halle mécatronique*.

Conclusion générale et perspectives

Au cours de cette thèse, nous avons étudié le problème théorique de la cartographie d'environnements et de la localisation de robot mobiles dans des milieux structurés. Du point de vue expérimental, nous avons implanté nos méthodes de construction de cartes et de localisation sur une plate-forme réelle, le robot Omni. L'architecture proposée est illustrée sur la Figure 1.3. Elle est aujourd'hui totalement opérationnelle sur le robot Omni [30], ce qui permet à ce robot de se localiser et de construire une carte de l'environnement en même temps. Le grand nombre d'expérimentations effectuées nous a permis de valider nos méthodes en simulation, puis sur site réel, et de les comparer avec d'autres approches classiques. Rappelons les différents points étudiés.

Afin d'obtenir un modèle cohérent de l'environnement mesuré par le système multicauteur (caméra vidéo et télémètre laser), nous avons défini et mis au point les prétraitements nécessaires pour synchroniser et corriger les données acquises. Tout d'abord, nous avons cherché à minimiser au niveau logiciel l'écart entre les instants de fin d'acquisition des images $\mathcal{L}$ et $\mathcal{I}$ du télémètre et de la caméra. Nous avons ensuite développé des procédures de caractérisation et de correction du biais de mesure du télémètre laser, de compensation de la déformation des images télémétriques due au mouvement du robot et de pseudo-synchronisation des images multicauteur. Nous avons illustré les apports de ces procédures sur des cas réels qui montrent l'intérêt de leur utilisation pour la localisation dynamique.

Ensuite, les données multicauteur sont segmentées pour extraire des primitives géométriques pouvant correspondre à des artefacts fiables pour la localisation. Dans l'image vidéo $\mathcal{I}$, nous cherchons des droites verticales correspondant à des contours significatifs de la scène observée. Leur extraction se fait en deux étapes : calcul de l'image gradient par filtrage et détection de droites

verticales par une transformée de Hough spécialisée. Dans l'image télémétrique qui représente le profil de la scène dans un plan horizontal, nous recherchons des points de forte discontinuité (points de cassure) et des segments de droite. Nous avons étudié deux approches de détection des points, une par seuillage adaptatif et l'autre par filtrage de Kalman. Notre contribution majeure a porté sur l'extraction des segments de droite pour laquelle nous avons proposé un nouvel algorithme basé sur les C-moyennes floues, le *SMF* [32] qui permet d'identifier ces segments et leur incertitude. Cet algorithme a été comparé en simulation à deux autres approches classiques. Une analyse quantitative des résultats obtenus nous a montré qu'avec un capteur précis les résultats du *SMF* sont comparables à ceux des approches classiques. Cependant, quand le niveau de bruit des mesures augmente, la robustesse de notre algorithme est supérieure. Les performances obtenues avec des données réelles confirment les résultats de la simulation.

La représentation 2-D de l'environnement observé par le système multicapteur pour une position donnée du robot constitue ce que nous avons appelé la carte locale. Elle est construite par fusion de données capteur représentées sous forme de primitives géométriques extraites par segmentation. Avec cette carte, nous introduisons des structures de niveau d'abstraction plus élevé : les demi-plans et les arêtes. Un demi-plan est composé d'une primitive segment de droite associée à sa zone de visibilité. Une arête représentée par un point peut être de trois sous-types : bord, coin et arête photométrique. Les deux premiers sont toujours associés à des demi-plans, ce qui établit une contrainte qui est toujours satisfaite au niveau de la carte locale. Une arête photométrique est détectée par la caméra vidéo, et ses paramètres 2-D sont estimées par *bootstrap* en utilisant des mesures télémétriques associées. Pour les procédures de fusion, nous avons eu recours à la transformation qui relie le repère image au repère du télémètre laser. Cette transformation a été estimée par étalonnage du système multicapteur. Une expérimentation nous a permis d'évaluer visuellement la qualité des structures obtenues, mais aussi de mettre en évidence le phénomène des observations multiples.

La localisation dynamique [33] a été implantée suivant le cycle classique prédiction-correction. Des techniques d'estime sont employées pour maintenir, à un taux d'échantillonnage important, une prédiction de la position du robot. Nous avons présenté deux techniques localisation à l'estime : l'odométrie, qui intègre les mesures des codeurs associés aux axes, et la gyrodométrie qui intègre en plus une mesure de changement du cap fournie par le gyromètre laser du robot. Nous avons identifié les paramètres du modèle géométrique du robot Omni, de façon à minimiser les erreurs d'odométrie. La gyrodométrie s'est avérée être une technique très fiable de localisation pour des expériences de courte durée. En conséquence, elle a été utilisée comme référence pour comparer nos méthodes robustes de recalage avec des approches classiques basées sur le filtrage stochastique.

Le correction de position s'effectue par mise en correspondance de cartes d'environnement. La mise en correspondance permet d'associer les structures de la carte locale $\mathcal{M}^R$ à des structures de la carte globale $\mathcal{M}^G$ pouvant correspondre au même artefact de l'environnement. Ce procédé rapide n'est pas néanmoins sans faille. Etant donné les incertitudes qui entourent les variables du processus, il peut en effet conduire à de fausses ou multiples correspondances. On rajoute à cela le fait que la cartographie locale n'est pas sans défaut. Les estimateurs de position n'utilisent que les primitives géométriques des structures de $\mathcal{M}^R$ et $\mathcal{M}^G$ en correspondance. Nous avons

présenté brièvement les méthodes basées sur le filtrage stochastique, ainsi que leurs limitations. Nous nous sommes intéressés notamment aux correspondances multiples qui peuvent se traduire par des distributions multimodales de vraisemblance de la position, étant donné les mesures. Pour illustrer leur inefficacité dans des cas de vraisemblance multimodale, nous avons utilisé le fait que ces filtres cherchent à estimer le mode de la vraisemblance [18][104]. En fait, l'estimation optimale est donnée par la moyenne, et non par les modes [104]. Dans les cas multimodaux, l'utilisation de la moyenne biaise l'estimation. Cela a conduit certains chercheurs à proposer des solutions multi-hypothèses plus adaptées [4][205]. De plus, nous avons vu que les approches classiques par filtrage stochastique non-linéaire souffrent de certaines limitations (cf. annexe E.4).

Notre deuxième contribution (après l'algorithme *SMF*) a porté sur des estimateurs robustes de position. Ils ont été développés à partir de nos études initiales sur l'estimation optimale de paramètres de position au sens des moindres carrés pondérés [31][35][36][37]. Le procédé d'estimation est non-itératif, et requiert une représentation unifiée des primitives géométriques en correspondance. La limitation principale de ce procédé est son incapacité à donner une solution dans les cas où ce nombre de primitives est insuffisant. Nous avons donc proposé une variante de cet algorithme, le *MCP* incomplet, qui utilise la décomposition en valeurs singulières et la fusion linéaire. Cette approche donne une solution garantie optimale, et sa robustesse provient essentiellement de l'utilisation de fonctions de pondération robustes pour la prise en compte des correspondances. À partir de cela, nous avons formulé le *MCP* incomplet comme un M-estimateur itératif. Nous avons vérifié ses performances des expériences réelles, et dans un cas où les approches par filtrage stochastique ont divergé. Cependant, nous n'avons pas eu le temps de tester des approches plus récentes, telles que la localisation Monte Carlo [205], les filtres multi-hypothèse [4][12], ou le filtrage adaptatif [111].

Ensuite, nos études se sont naturellement orientées vers le problème de la cartographie d'environnements. En fait, c'est avec le développement d'une méthode de cartographie que nous avons pu évaluer, dans des conditions plus réalistes, les différentes approches de localisation. Nous avons alors proposé une approche découplée pour la cartographie [34]. Nous nous sommes intéressés à des outils récents de filtrage linéaire optimal consistant en de la propagation d'incertitude. Dans cette approche, comme dans les approches couplées, des erreurs de localisation peuvent biaiser le processus de cartographie. En conséquence, nous avons intégré dans notre algorithme une stratégie de propagation de contraintes stochastiques, de façon à limiter les dégâts d'une localisation peu performante. Ces contraintes qui existent déjà au niveau des structures de la carte locale, sont maintenant propagées dans la carte globale. Nous avons vérifié les apports de ces différents outils mathématiques dans le cadre de la cartographie découplée. Cela a d'abord été fait avec des données acquises dans un site réel, lors d'une expérience de courte durée. Avec une expérience de plus longue durée, nous avons observé l'influence de la localisation dans le processus de cartographie. Dans ce cas, des résultats très satisfaisants ont été obtenus avec un M-estimateur pour le recalage, la gyrodométrie pour l'estime et notre approche découplée pour la mise à jour de la carte. En effet, malgré la bonne précision de la gyrodométrie, le recalage par filtrage stochastique n'avait pas donné de bons résultats dans cette expérience de longue durée.

Nous avons, à travers ce manuscrit, essayé de montrer les diverses sources de difficulté qui

interviennent dans le problème de la localisation autonome et de la cartographie d'environnement. Nous avons proposé des solutions qui ont fait preuve de robustesse dans les cas où les approches classiques peuvent échouer. Cependant, même que si des bons résultats ont été obtenus, le monde réel est non-déterministe et imprévisible, et il serait prétentieux de notre part d'affirmer que nos solutions sont universelles.

Bien que les approches de localisation proposées ici soient relativement performantes, elles ne sont pas suffisantes pour réaliser un robot mobile complètement autonome. En effet, son système de localisation doit intégrer un certain nombre de fonctionnalités dont nous ne parlons pas dans le cadre de cette étude, comme par exemple : l'auto-localisation, la planification de trajectoires, ou l'évitement d'obstacles.

Par ailleurs, plusieurs pistes sont possibles pour améliorer ou généraliser les solutions proposées :

- Malgré la complexité réduite de l'approche découplée pour la cartographie, on peut imaginer que, pour des environnements de grande taille, les algorithmes seront plus coûteux en temps de calcul. Une solution élégante pour réduire cette complexité pourrait être celle proposée par Leonard *et al.* [143]. Elle consiste à décomposer l'environnement en sous-cartes ayant un certain recouvrement. Les auteurs présentent une procédure pour préserver la consistance de l'estimation de position lors du changement de sous-cartes. Une idée intéressante serait d'appliquer ces concepts à notre système de cartographie afin d'optimiser son implantation dans de grands environnements ;
- Des approches récentes [68][95] utilisent le *bundle adjustment*, pour reconstruire un environnement 3D à partir d'une séquence d'images vidéo acquises par une caméra en mouvement [94]. Cette méthode, basée sur la mise en correspondance des images, estime les positions successives du capteur par minimisation d'une fonction de coût étendue des résidus de correspondance de toutes les images de la séquence. La minimisation se fait via un calcul hors-ligne (il requiert que toutes les données soient déjà disponibles) en utilisant des techniques itératives. Appliquée à la cartographie, cette technique permettrait d'estimer les positions consécutives du capteur, $Z_1, \dots, Z_N$. En effet, si pour deux mesures capteur consécutives aux instants n et $n + 1$, on dispose d'un ensemble de primitives géométriques communes, cela permet d'établir des correspondances entre Z_n et Z_{n+1} .

On peut envisager d'appliquer un MCP incomplet dans le noyau du bundle adjustment, pour résoudre le problème d'estimation du déplacement relatif ΔZ_n entre deux positions consécutives Z_n et Z_{n+1} , initialisées par odométrie. L'idée serait d'utiliser des cycles d'estimation $\Delta Z_1, \dots, \Delta Z_{N-1}$, où l'on actualiserait les mesures de correspondance des indices observés en Z_n et Z_{n+1} . Cela pourrait être une solution à la cartographie d'environnements plus larges en utilisant l'odométrie comme technique d'estime ;

- L'étude sur la cartographie s'est davantage concentrée sur des problèmes d'estimation, que sur la gestion de structures. Notre méthode de cartographie/localisation tolère la présence d'une proportion raisonnable d'obstacles mobiles dans le champ des capteurs. Cependant, dans le cas d'environnements très dynamiques (*i.e.*, avec beaucoup d'obstacles mobiles), il faut prévoir un module décisionnel capable de sélectionner des structures fiables pouvant être incorporées à la carte. Ce problème non trivial commence à être étudié [187]. Une

solution consiste à associer aux nouvelles structures une mesure de fiabilité qui augmente en fonction du nombre de fois où elles sont observées [71][226]. Il faudrait ainsi prendre en compte également la stabilité de ces structures au cours du temps dans le repère de la carte globale ;

- L'étude développée dans cette thèse s'est limitée au cas de robots mobiles navigant sur sol plan, dans des environnements relativement structurés. La suite à plus long terme de ce travail consistera à étudier l'applicabilité des outils mathématiques proposés au cas plus général de la navigation 3D dans des environnements naturels. En effet, les solutions développées pour la navigation dans un espace 2D peuvent également s'appliquer lorsque les primitives disponibles sont essentiellement des points de l'espace 3D.

Bibliographie

- [1] M.-J. ALDON. « Capteurs et méthodes pour la localisation des robots mobiles ». *Techniques de l'ingénieur*, S (article S7852), 2001.
- [2] M.-J. ALDON et M. BENOIT. « Localization system of the hospital transport robot FIRST ». Dans *2nd IFAC Conference on Intelligent Autonomous Vehicles*, pages 244–249, Helsinki, Finland, June 12 - 14 1995.
- [3] P. ALLEN et R. BAJCSY. « Two sensors are better than one : example of vision and touch ». Dans *Proceedings of 3rd International Symposium on Robotics Research*, pages 48–55, 1986.
- [4] F. M. ANTONIALI et G. ORIOLO. « Robot localization in nonsmooth environments : experiments with a new filtering technique ». Dans *IEEE International Conference on Robotics and Automation*, pages 1591–1596, 2001.
- [5] J. ARAGONES et A. FOURNIER. « Commande dynamique d'un robot omni-directionnel redondant ». Dans *Conférence Internationale Francophone d'Automatique (CIFA)*, 2002.
- [6] E. G. ARAUJO et R. A. GRUPEN. « Feature detection and Identification using a Sonar-Array ». Dans *IEEE International Conference on Robotics and Automation*, pages 1584–1589, 1998.
- [7] K. O. ARRAS et N. TOMATIS. « Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision ». Dans *Third European Workshop on Advanced Mobile Robots (Eurobot 99)*, pages 42–53, 1999.
- [8] K. O. ARRAS, N. TOMATIS, B. T. JENSEN, et R. SIEGWART. « Multisensor on-the-fly localization : Precision and reliability for applications ». *Robotics and Autonomous Systems*, 34 :131–143, 2001.
- [9] A. ASRÉNIO et M. I. RIBEIRO. « Active Range Sensing for Mobile Robot Localization ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1066–1071, 1997.
- [10] S. ATIYA et G. HAGER. « Real-time vision-based localization ». Dans *IEEE International Conference on Robotics and Automation*, pages 639–644, 1991.
- [11] S. ATIYA et G. HAGER. « Real-time vision-based localization ». *IEEE Transactions on Robotics and Automation*, 9(6) :785–800, December 1993.
- [12] D. J. AUSTIN et P. JENSEFELT. « Using Multiple Gaussian Hypotheses to represent probability Distributions for Mobile Robot Localization ». Dans *IEEE International Conference on Robotics and Automation*, pages 1036–1041, 2000.
- [13] A. BAB-HADIASHAR et D. SUTER. *Data Segmentation and Model Selection for Computer Vision*. Springer, 2000.

- [14] P. BALLARD et F. VACHERAND. « Simulation and Understanding of Range Images acquired in Fast Motion ». Dans *IEEE International Conference on Robotics and Automation*, pages 2242–2247, 1994.
- [15] Y. BAR-SHALOM et X.-R. LI. *Estimation and Tracking : Principles, Techniques and Software*. Norwood : Artech House, 1987.
- [16] M. BARNI, V. CAPPELLINI, A. PAOLI, et A. MECOCCHI. « Unsupervised Detection Of Straight Lines Through Possibilistic Clustering ». Dans *IEEE International Conference on Image Processing*, pages 963–966, 1996.
- [17] B. BARSHAN et H. F. DURRANT-WHYTE. « Inertial navigation systems for mobile robots ». *IEEE Transactions on Robotics and Automation*, 11(3) :328–342, June 1995.
- [18] B. M. BELL et F. W. CATHEY. « The Iterated Kalman Filter Update as a Gauss-Newton Method ». *IEEE Transactions on Automatic Control*, 38(2) :294–297, February 1993.
- [19] J. C. BEZDEK. *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York, USA, 1981.
- [20] M. J. BLACK et A. RANGARAJAN. « On the unification of line processes, outlier rejection, and robust statistics with applications in early vision ». *International Journal of Computer Vision*, 19(1) :57–92, July 1996.
- [21] S. S. BLACKMAN et R. POPOLI. *Design and analysis of modern tracking systems*. Artech House, 1999.
- [22] D. L. BOLEY et K. T. SUTHERLAND. « A Rapidly Converging Recursive Method for Mobile Robot Localization ». *The International Journal of Robotics Research*, 17(10) :1027–1039, October 1998.
- [23] E. BØLVIKEN, P. J. ACKLAM, N. CHRISTOPHERSEN, et J.-M. STØRDAL. « Monte Carlo Filters for non-linear state estimation ». *Automatica*, 37 :177–183, 2001.
- [24] P. BONNIFAIT et G. GARCIA. « Design and Experimental Validation of an Odometric and Goniometric Localization System for Outdoor Robot Vehicles ». *IEEE Transactions on Robotics and Automation*, 14(4) :541–548, August 1998.
- [25] J. BORENSTEIN et J. EVANS. « The OmniMate Mobile Robot - Design, Implementation, and Experimental Results ». Dans *IEEE International Conference on Robotics and Automation*, pages 3505–3510, 1997.
- [26] J. BORENSTEIN, H. R. EVERETT, et L. FENG. *Navigating Mobile Robots : Systems and Techniques*. A K Peters, Wellesley, Massachusetts, 1996.
- [27] J. BORENSTEIN et L. FENG. « Gyrodometry : a new method for combining data from gyros and odometry in mobile robots ». Dans *IEEE International Conference on Robotics and Automation*, pages 423–428, 1996.
- [28] J. BORENSTEIN et L. FENG. « Measurement and Correction of Systematic Odometry Errors in Mobile Robots ». *IEEE Transactions on Robotics and Automation*, 6(12) :869–880, December 1996.
- [29] G. A. BORGES. « Acquisition et datation de données en robotique mobile : une étude de cas utilisant RTX ». GDR ISIS OT 6.3 : Journées d'exposés sur la perception et fusion de données pour véhicules intelligents, ENST, Paris, Octobre 2000.
- [30] G. A. BORGES. « Architectures matérielle et logicielle du robot Omni ». Rapport Technique 02082, LIRMM UMR CNRS/UMI n°. 55060, 161, rue ADA - 34392 - Montpellier - France, 2002.
- [31] G. A. BORGES et M.-J. ALDON. « Motion Estimation by Iterative 2-D Features Matching in Range Images ». Dans *IEEE International Conference on Robotics and Automation*, April 2000.
- [32] G. A. BORGES et M.-J. ALDON. « A Split-and-Merge segmentation algorithm for line extractions in 2-D range images ». Dans *15th International Conference on Pattern Recognition*, September 2000.
- [33] G. A. BORGES et M.-J. ALDON. « Design of a Robust Real-Time Dynamic Localization System for Mobile Robots ». Dans *9th International Symposium on Intelligent Robotic Systems*, 2001.
- [34] G. A. BORGES et M.-J. ALDON. « A Decoupled Approach for Simultaneous Stochastic Mapping and Mobile Robot Localization ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- [35] G. A. BORGES et M.-J. ALDON. « Optimal Mobile Robot Pose Estimation Using Geometrical Maps ». *IEEE Transactions on Robotics and Automation*, 18(1) :87–94, February 2002.
- [36] G. A. BORGES, M.-J. ALDON, et T. GIL. « An Optimal Pose Estimator for Map-based Mobile Robot Dynamic Localization : Experimental Comparison with the EKF ». Dans *IEEE International Conference on Robotics and Automation*, 2001.

- [37] G. A. BORGES, M.-J. ALDON, et O. STRAUSS. « Estimation de mouvement par appariement de droites 2-D dans des images de distance ». Dans *12^{ème} Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle*, Février 2000.
- [38] G. A. BORGES, A. M. N. LIMA, et G. S. DEEP. « Characterization of a Trajectory Recognition Optical Sensor for an Automated Guided Vehicle ». *IEEE Transactions on Instrumentation and Measurement*, 49(4) :813–819, August 2000.
- [39] S. BORTHWICK et H. DURRANT-WHYTE. « Simultaneous localization and Map Building for autonomous guided vehicles ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 761–768, 1994.
- [40] S. BORTHWICK, M. STEVENS, et H. DURRANT-WHYTE. « Position Estimation and Tracking using Optical Range data ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2172–2177, 1993.
- [41] H. BULATA et M. DEVY. « Incremental Construction of a Landmark-based and Topological Model of Indoor Environments by a Mobile Robot ». Dans *IEEE International Conference on Robotics and Automation*, pages 1054–1060, 1996.
- [42] W. BURGARD, A.B. CREMERS, D. FOX, D. HÄHNEL, G. LAKEMEYER, D. SCHULZ, W. STEINER, et S. THRUN. « Experiences with an interactive museum tour-guide robot ». *Artificial Intelligence*, 114 :3–55, 1999.
- [43] W. BURGARD, D. FOX, et D. HENNIG. « Fast Grid-Based Position Tracking for Mobile Robots ». Dans G. BREWKA, C. HABEL, et B. NEBEL, éditeurs, *21th German Conference on Artificial Intelligence*, volume 1303 de *Lecture Notes in Computer Science*. Springer, 1997.
- [44] W. BURGARD, D. FOX, et S. THRUN. « Active Mobile Robot Localization ». Dans *Fourteenth International Joint Conference on Artificial Intelligence*, pages 1346–1352, 1997.
- [45] J. B. BURNS, A. R. HANSON, et E. M. RISEMAN. « Extracting Straight Lines ». *PAMI*, 8(4) :425–455, July 1986.
- [46] F. CALISKAN et C. M. HAJIYEV. « Innovation sequence application to aircraft sensor fault detection : comparison of checking covariance matrix algorithms ». *International Society for Measurement and Control (ISA) Transactions*, 39(1) :47–56, February 2000.
- [47] G. CAMPION, G. BASTIN, et B. D'ANDRÉA-NOVEL. « Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots ». *IEEE Transactions on Robotics and Automation*, 12(1) :47–62, February 1996.
- [48] J. CANNY. « A Computational Approach to Edge Detection ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986.
- [49] J. A. CASTELLANOS, M. DEVY, et J. D. TARDÓS. « Simultaneous localization and map building for mobile robots : a landmark based approach ». Dans *IEEE International Conference on Robotics and Automation : Workshop on mobile robot navigation and mapping*, April 2000.
- [50] J. A. CASTELLANOS, J. NEIRA, O. STRAUSS, et J. D. TARDÓS. « Detecting High Level Features for Mobile Robot Localization ». Dans *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration*, pages 2917–2923, 1996.
- [51] J. A. CASTELLANOS et J. D. TARDÓS. Laser-based segmentation and localization for a mobile robot. Dans F. Pin M. JAMSHIDI et P. DAUCHEZ, éditeurs, *Robotics and Manufacturing : Recent Trends in Research and Applications*, volume 6, pages 101–109. ASME Press, 1996.
- [52] J. A. CASTELLANOS, J. D. TARDÓS, et G. SCHMIDT. « Building a global map of the environment of a mobile robot : the importance of correlations ». Dans *IEEE International Conference on Robotics and Automation*, pages 1053–1059, 1997.
- [53] C.-T. CHANG et J.-I. HWANG. « Simplification techniques for EKF computations in fault diagnosis - suboptimal gains ». *Chemical Engineering Science*, 53(22) :3771–3911, November 1998.
- [54] R. CHEN et J. S. LIU. « Mixture Kalman filters ». *Journal of the Royal Statistical Society : Series B (Methodological)*, 62(3) :493–508, 2000. Disponible en ligne dans <http://www.people.fas.harvard.edu/~junliu/TechRept/00.html>.

- [55] F. CHENAVIER, I. L. TAIBI, et J. L. CROWLEY. « Estimation de la Position d'un Robot par Odométrie et Vision Monoculaire ». *Traitement du Signal*, 12(3) :255–266, 1995.
- [56] K. S. CHONG et L. KLEEMAN. « Accurate Odometry and Error Modelling for a Mobile Robot ». Dans *IEEE International Conference on Robotics and Automation*, pages 2783–2788, 1997.
- [57] K. S. CHONG et L. KLEEMAN. « Mobile-robot map building from an advanced sonar array and accurate odometry ». *International Journal of Robotics Research*, 18(1) :20–36, 1999.
- [58] H. CHOSET et K. NAGATANI. « Topological Simultaneous Localization and Mapping (SLAM) : Toward Exact Localization Without Explicit Localization ». *IEEE Transactions on Robotics and Automation*, 17(2) :125–137, April 2001.
- [59] C. K. CHUI et G. CHEN. *Kalman filtering with real-time applications*. Series in Information Sciences. Springer-Verlag, 1991.
- [60] H. CHUNG, L. OJEDA, et J. BORENSTEIN. « Accurate Mobile Robot Dead-Reckoning with a Precision-Calibrated Fiber-Optic Gyroscope ». *IEEE Transactions on Robotics and Automation*, 17(1) :80–84, February 2001.
- [61] S. CLARK et G. DISSANAYAKE. « Simultaneous localisation and map building using millimeter wave radar to extract natural features ». Dans *IEEE International Conference on Robotics and Automation*, pages 1316–1321, 1999.
- [62] A. CLERENTIN, L. DELAHOCHÉ, C. PEGARD, et E. BRASSART. « A Localization method based on two omnidirectional perception systems cooperation ». Dans *IEEE International Conference on Robotics and Automation*, pages 1219–1224, 2000.
- [63] I. J. COX. « Blanche - An experiment in guidance and navigation of an autonomous robot vehicle ». *IEEE Transactions on Robotics and Automation*, 7(2) :193–204, April 1991.
- [64] G. DANUSER et M. STRICKER. « Parametric Model Fitting : from Inlier Characterisation to Outlier Detection ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2) :263–280, March 1998.
- [65] R. N. DAVÉ et R. KRISHNAPURAM. « Robust Clustering Methods : A Unified View ». *IEEE Transactions on Fuzzy Systems*, 5(2) :270–293, 1997.
- [66] A. J. DAVISON et N. KITA. « Sequential localisation and map-building for real-time computer vision and robotics ». *Robotics and Autonomous Systems*, 36(4) :171–183, September 2001.
- [67] A. J. DAVISON et D. W. MURRAY. « Mobile robot localisation using active vision ». Dans *European Conference on Computer Vision*, 1998.
- [68] M. DEANS et M. HEBERT. « Experimental comparison of techniques for localization and mapping using a bearings-only sensor ». Dans *International Symposium on Experimental Robotics*, 2000.
- [69] F. DELLAERT, D. FOX, W. BURGARD, et S. THRUN. « Monte Carlo Localization for Mobile Robots ». Dans *IEEE International Conference on Robotics and Automation*, 1999.
- [70] R. DERICHE. « Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector ». *The International Journal of Computer Vision*, 1(2) :167–187, 1987.
- [71] M. W. M. G. DISSANAYAKE, P. NEWMAN, S. CLARK, H. F. DURRANT-WHYTE, et M. CSORBA. « A Solution to the Simultaneous Localization and Map Building (SLAM) Problem ». *IEEE Transactions on Robotics and Automation*, 17(3) :229–241, June 2001.
- [72] T. DUCKETT, S. MARSLAND, et J. SHAPIRO. « Learning globally consistent maps by relaxation ». Dans *IEEE International Conference on Robotics and Automation*, pages 3841–3846, 2000.
- [73] R. O. DUDA et P. E. HART. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, USA, 1973.
- [74] B. EFRON. « Bootstrap methods : another look at the jackknife ». *Journal of Ann. Stat.*, 7 :1–26, 1979.
- [75] T. EINSELE. « Real-Time Self-Localization in Unknown Indoor Environments Using a Panorama Laser Range Finder ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 697–702, 1997.
- [76] A. ELFES. « *Occupancy grids : A probabilistic Framework for Robot Perception and Navigation* ». PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.

- [77] H. J. S. FEDER, J. J. LEONARD, et C. M. SMITH. « Adaptive mobile robot navigation and mapping ». *International Journal of Robotics Research*, 18(7) :650–668, 1999.
- [78] J. FORSBERG, U. LARSSON, et Å. WERNERSSON. « Mobile robot navigation using the range-weighted Hough transform ». *IEEE Robotics and Automation Magazine*, pages 18–26, March 1995.
- [79] D. FOX, W. BURGARD, S. THRUN, et A. B. CREMERS. « Position estimation for mobile robots in dynamic environments ». Dans *AAAI Conference on Innovative Applications of Artificial Intelligence*, pages 983–988, 1998.
- [80] H. FRIGUI et R. KRISHNAPURAM. « A Robust Competitive Clustering Algorithm With Applications in Computer Vision ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5) :450–465, May 1999.
- [81] J. De GEETER, H. Van BRUSSEL, J. De SCHUTTER, et M. DECRETON. « A Smoothly Constrained Kalman Filter ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10) :1171–1177, October 1997.
- [82] P. GOEL, G. DEDEOGLU, S. I. ROUMELIOTIS, et G. S. SUKHATME. « Fault Detection and Identification in a Mobile Robot Using Multiple Model Estimation and Neural Network ». Dans *IEEE International Conference on Robotics and Automation*, pages 2302–2309, 2000.
- [83] P. GOEL, S. I. ROUMELIOTIS, et G. S. SUKHATME. « Robust Localization Using Relative and Absolute Position Estimates ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1134–1140, 1999.
- [84] M. GOLFARELLI, D. MARIO, et S. RIZZI. « Correction of dead-reckoning errors in map building in map building for mobile robots ». *IEEE Transactions on Robotics and Automation*, 17(1) :37–47, February 2001.
- [85] Covariance Intersection Working GROUP. « A culminating advance in the theory and practice of data fusion, filtering, and decentralized estimation ». Rapport Technique, Disponible en ligne dans <http://www.ait.nrl.navy.mil/people/uhlmann/CovInt.html>, 1997.
- [86] J. E. GUIVANT et E. M. NEBOT. « Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation ». *IEEE Transactions on Robotics and Automation*, 17(3) :242–257, June 2001.
- [87] J.-S. GUTMANN, W. BURGARD, D. FOX, et K. KONOLIGE. « An Experimental Comparison of Localization Methods ». Dans *International Conference on Intelligent Robots and Systems*, 1998.
- [88] J.-S. GUTMANN et C. SCHLEGEL. « AMOS : Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments ». Dans *1st Euromicro Workshop on Advanced Mobile Robots*, 1996.
- [89] J.-S. GUTMANN, T. WEIGEL, et B. NEBEL. « Fast, Accurate, and Robust Self-Localization in Polygonal Environments ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1412–1419, 1999.
- [90] U. D. HANEBECK et K. BRIECHLE. « New Results for Stochastic Prediction and Filtering with Unknown Correlations ». Dans *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 147–152, 2001.
- [91] U. D. HANEBECK et J. HORN. « An Efficient Method for Simultaneous Map Building and Localization ». Dans *Proceedings of AeroSense Symposium*, volume 4385, pages 147–152. SPIE, 2001.
- [92] R. M. HARALICK, H. JOO, C.-N. LEE, X. ZHUANG, V. G. VAIDYA, et M. B. KIM. « Pose Estimation from Corresponding Point Data ». *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6) :1426–1446, November/December 1989.
- [93] R.M. HARALICK. « Propagating Covariances in Computer Vision ». Dans *International Conference on Pattern Recognition*, pages 493–498, 1994.
- [94] R. HARTLEY et A. ZISSERMAN. *Multiple View Geometry*. Cambridge University Press, Cambridge, UK, 2000.
- [95] M. HEBERT, M. DEANS, D. HUBER, B. NABBE, et N. VANDAPEL. « Progress in 3-D Mapping and Localization ». Dans *Symposium Intelligent Robotic Systems*, pages 145–154, 2001.
- [96] Y. HEL-OR et Werman M.. « Constraint-Fusion for Interpretation of Articulated Objects ». Dans *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 39–45, 1994.

- [97] D. HERMEY et G. A. WATSON. « Fitting Data with Errors in All Variables Using the Huber M-estimator ». *SIAM Journal on Scientific Computing*, 20(4) :1276–1298, 1999.
- [98] A. A. HOLENSTEIN, M. A. MÜLLER, et E. BADREDDIN. « Mobile Robot Localization in a Structured Environment Cluttered with Obstacles ». Dans *IEEE International Conference on Robotics and Automation*, pages 2576–2581, May 1992.
- [99] R. HORAUD et O. MONGA. *Vision par ordinateur*. Hermès, Paris, France, 1995.
- [100] P. J. HUBER. *Robust Statistics*. John Wiley & Sons, 1981.
- [101] B. JÄHNE et H. HAUßECKER. *Computer vision and Applications : a Guide for Students and Practitioners*. Academic Press, London, UK, 2000.
- [102] J. A. JANÉ, R. GUTIERREZ, T. A. CHASE, M. W. WHITE, et III J. C. SUTTON. « Autonomous mobile robot global self-localization using kohonen and region-feature neural networks ». *Journal of Robotic Systems*, 14(4) :263–282, 1997.
- [103] L. JAULIN, M. KIEFFER, O. DIDRIT, et E. WALTER. *Applied Interval Analysis*. Springer-Verlag, London, UK, 2001.
- [104] A. H. JAZWINSKI. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [105] M. JENKIN, E. MILIOS, et P. JASIOBEDZKI. « Global Navigation for ARK ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2165–2171, 1993.
- [106] P. JENSFELT, D. J. AUSTIN, O. WIJK, et M. ANDERSON. « Feature-based Condensation for Mobile Robot Localization ». Dans *IEEE International Conference on Robotics and Automation*, pages 2531–2537, 2000.
- [107] P. JENSFELT et H. I. CHRISTENSEN. « Laser based position acquisition and tracking in an indoor environment ». Dans *International Symposium on Robotics and Automation*, pages 331–338, 1998.
- [108] P. JENSFELT et H. I. CHRISTENSEN. « Pose Tracking Using Laser Scanning and Minimalistic Environment Models ». *IEEE Transactions on Robotics and Automation*, 17(2) :138–147, April 2001.
- [109] P. JENSFELT, O. WIJK, J. AUSTIN, et M. ANDERSON. « Experiments on Augmenting Condensation for Mobile Robot Localization ». Dans *IEEE International Conference on Robotics and Automation*, pages 2518–2524, 2000.
- [110] H. J. JEON et B. K. KIM. « Feature-based probabilistic map building using time and amplitude information of sonar in indoor environments ». *Robotica*, 19 :423–437, 2001.
- [111] L. JETTO, S. LONGHI, et G. VENTURINI. « Development and Experimental Validation of an Adaptive Kalman Filter for the Localization of Mobile Robots ». *IEEE Transactions on Robotics and Automation*, 15(2) :219–229, 1999.
- [112] J. M. JOLION, P. MEER, et S. BATAOUCHE. « Robust Clustering with Applications in Computer Vision ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8) :791–802, August 1991.
- [113] S. J. JULIER. « *Process Models for the Navigation of High-Speed Land Vehicles* ». PhD thesis, University of Oxford, Robotics Research Group, Department of Engineering Science., 1997.
- [114] S. J. JULIER, J. UHLMANN, et H. F. DURRANT-WHYTE. « A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators ». *IEEE Transactions on Robotics and Automation*, 45(3) :477–482, 2000.
- [115] S. J. JULIER et J. K. UHLMANN. « A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations ». Dans *American Control Conference*, 1997.
- [116] S. J. JULIER et J. K. UHLMANN. « A Counter Example to the Theory of Simultaneous Localization and Map Building ». Dans *IEEE International Conference on Robotics and Automation*, 2001.
- [117] S. J. JULIER, J. K. UHLMANN, et H. DURRANT-WHYTE. « A New approach for filtering non-linear systems ». Dans *American Control Conference*, pages 1628–1632, 1995.
- [118] R. E. KALMAN. « A New Approach to Linear Filtering and Prediction Problems ». *Transactions ASME, Ser. D : J. Basic Eng.*, 82 :35–45, 1960.
- [119] R. E. KALMAN. Algebraic aspects of the generalized inverse of a rectangular matrix. Dans M. Z. NASHED, éditeur, *Generalized Inverses and Applications*, pages 111–124. Academic Press, Inc., 1976.

- [120] B. KAMGAR-PARSI et B. KAMGAR-PARSI. « Matching Sets of 3D Line Segments with Applications to Polygonal Arc Matching ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10) :1090–1099, October 1997.
- [121] M. KIEFFER, L. JAULIN, E. WALTER, et D. MEIZEL. « Robust autonomous robot localization using interval analysis ». *Reliable Computing*, 3(6) :337–361, 2000.
- [122] L. KLEEMAN. « Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning ». Dans *IEEE International Conference on Robotics and Automation*, pages 2582–2587, 1992.
- [123] R. KRISHNAPURAM, H. FRIGUI, et O. NASRAOUI. « Fuzzy and Possibilistic Shell Clustering Algorithms and their application to boundary detection and surface approximation - Part I ». *IEEE Transactions on Fuzzy Systems*, 3(1) :29–43, February 1995.
- [124] B. J. A. KRÖSE et M. EECEN. « A self-organizing representation of sensor space for mobile robot navigation ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 9–14, 1994.
- [125] B. J. A. KRÖSE, N. VLASSIS, R. BUNSCHOTEN, et Y. MOTOMURA. « A probabilistic model for appearance-based robot localization ». *Image and Vision Computing*, 19 :381–391, 2001.
- [126] B. KUIPERS. The cognitive map : could it have been any other way ? Dans H. L. Pick JR. et L. P. ACREDOLO, éditeurs, *Spatial orientation : theory, research, and application*, pages 345–359. Preum Press, New York, 1983.
- [127] B. KUIPERS et Y. T. BYAN. « A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations ». *Journal of Robotics and Autonomous Systems*, 8 :47–63, 1991.
- [128] Y. D. KWON et J. S. LEE. « A Stochastic Environment Modelling Method for Mobile Robot by using 2-D Laser scanner ». Dans *IEEE International Conference on Robotics and Automation*, pages 1688–1693, 1997.
- [129] D. LAGUNOVSKY et S. ABLAMEYKO. « Straight-line-based primitive extraction in grey-scale object recognition ». *Pattern Recognition Letters*, 20(10) :1005–1014, October 1999.
- [130] A. LALLEMENT, A. SIADAT, M. DUFAUT, et R. HUSSON. « Laser-vision cooperation for map building and landmarks recognition ». Dans *IEEE ISIC/CIRA/ISAS Joint Conference*, pages 387–392, 1998.
- [131] U. LARSSON, J. FORSBERG, et Å. WERNERSSON. « On robot navigation using identical landmarks : integrating measurements from time-of-flight laser ». Dans *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 17–26, 1994.
- [132] U. LARSSON, J. FORSBERG, et Å. WERNERSSON. « Mobile robot localization : integrating measurements from a time-of-flight laser ». *IEEE Transactions on Industrial Electronics*, 43(3) :422–431, June 1996.
- [133] Y. le CORRE. « Conception et commande d'un robot omnidirectionnel ». PhD thesis, LIRMM / Université Montpellier II, Décembre 1998.
- [134] Y. le CORRE, A. FOURNIER, et M. BENOIT. « Performances of a 3 caster wheeled mobile robot : the advantages of over-actuation ». Dans *International Conference on Advances in Vehicle Control and Safety (ACVS)*, 1998.
- [135] J. E. LECKY. « Turbocharged Algorithms : how to optimize a machine vision application for MMX ». *Image Processing Europe*, pages 16–20, march 1999.
- [136] J.-W. LEE et I.-S. KWEON. « Extraction of Line Features in a Noisy Image ». *Pattern Recognition*, 30(10) :1651–1660, October 1997.
- [137] J.-W. LEE et S. LEE. « A New Data Fusion method and Its Application to State Estimation of Nonlinear Dynamic Systems ». Dans *IEEE International Conference on Robotics and Automation*, pages 3525–3530, 2000.
- [138] W.-H. LEE, K.-S. ROH, et I.-S. KWEON. « Self-localization of a mobile robot without camera calibration using projective invariants ». *Pattern Recognition Letters*, 21 :46–60, 2000.
- [139] J. J. LEONARD et H. F. DURRANT-WHYTE. « Mobile Robot Localization by Tracking Geometric Beacons ». *IEEE Transactions on Robotics and Automation*, 7(3) :376–382, June 1991.
- [140] J. J. LEONARD et H. F. DURRANT-WHYTE. « Simultaneous Map Building and localization for an Autonomous Mobile Robot ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1442–1447, 1991.

- [141] J. J. LEONARD, H. F. DURRANT-WHYTE, et I. J. COX. « Dynamic Map Building for an Autonomous Mobile Robot ». *The International Journal of Robotics Research*, 11(4) :286–298, August 1992.
- [142] J. J. LEONARD et H. J. S. FEDER. « A computationally efficient method for large-scale concurrent mapping and localization ». Dans *Proceedings of the Ninth International Symposium on Robotics Research*, pages 169–176, 2000.
- [143] J. J. LEONARD et H. J. S. FEDER. « Decoupled Stochastic Mapping ». *IEEE Journal of Oceanic Engineering*, 26(4) :561–571, October 2001.
- [144] J. J. LEONARD et J. D. TARDÓS. « Mobile robot navigation an mapping ». Dans *IEEE International Conference on Robotics and Automation : Workshop on mobile robot navigation an mapping*, April 2000.
- [145] D. LERRO et Y. BAR-SHALOM. « Tracking with debiased consistent converted measurements versus EKF ». *IEEE Transactions on Aerospace Electronic Systems*, 29 :1017–1022, 1993.
- [146] O. LÉVÊQUE, L. JAULIN, D. MEIZEL, et E. WALTER. « Vehicle Localization from Inacurate Telemetric Data : A Set-Inversion Approach ». Dans *Fifth IFAC Symposium on Robot Control*, pages 179–185, 1997.
- [147] J. H. LIM et J. J. LEONARD. « Mobile robot relocation from echolocation constraints ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9) :1035–1041, September 2000.
- [148] L. LJUNG. *System Identification - Theory for the User*. Prentice-Hall, 1999.
- [149] J. LOBO, C. QUEIROZ, et J. DIAZ. « Vertical world feature detection and mapping using stereo vision and accelerometers ». Dans *International Symposium on Intelligent Robotic Systems*, pages 229–238, 2001.
- [150] D. G. LUENBERGER. *Linear and nonlinear programming*. Addison-Wesley Publishing Company, second édition, 1989.
- [151] E. A. MAGUIRE, N. BURGESS, et J. O'KEEFE. « Human Spatial Navigation : cognitive maps, sexual dimorphism and Neural Substrates ». *Current Opinion in Neurobiology*, 9 :171–177, 1999.
- [152] O. L. MANGASARIAN et D. R. MUSICANT. « Robust Linear and Support Vector Regression ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9) :950–955, September 2000.
- [153] J. A. CASTELLANOS J. M. MARTÍNEZ, J. NEIRA, et J. D. TARDÓS. « Experiments in Multisensor Mobile Robot Localization and Map Building ». Dans *3rd IFAC Symposium on Intellignet Autonomous Vehicles*, 1998.
- [154] J. A. CASTELLANOS J. M. MARTÍNEZ, J. NEIRA, et J. D. TARDÓS. « Simultaneous Map Building and localization for mobile robots : a multisensor fusion approach ». Dans *IEEE International Conference on Robotics and Automation*, pages 1244–1249, 1998.
- [155] M. MATTAVELLI, V. NOEL, et E. AMALDI. « A New Approach for Fast Line Detection Based on Combinatorial Optimization ». Dans *10th International Conference on Image Analysis and Processing*, pages 168–173, 1999.
- [156] C. D. MCGILLEM et T. S. RAPPAPORT. « Infra-red location system for navigation of autonomous vehicles ». Dans *IEEE International Conference on Robotics and Automation*, pages 1236–1238, 1988.
- [157] R. MCKENDALL et M. MINTZ. Data Fusion Techniques Using Robust Statistiques. Dans M. A. ABIDI et R. C. GONZALEZ, éditeurs, *Data Fusion in Robotics and Machine Intelligence*, pages 211–244. Academic Press, Inc., 1992.
- [158] J. A. CASTELLANOS J. M. M. MONTIEL, J. NEIRA, et J. D. TARDÓS. « The SPmap : a probabilistic framework for simultaneous localization and map building ». *IEEE Transactions on Robotics and Automation*, 15(5) :948–952, October 1999.
- [159] J. A. CASTELLANOS J. M. M. MONTIEL, J. NEIRA, et J. D. TARDÓS. Sensor influence in the performance of simultaneous mobile robot localization and map building. Dans P. CORKE et J. TREVELYAN, éditeurs, *Experimental Robotics VI*, pages 287–296. Springer Verlag, Inc., 2000.
- [160] J. M. M. MONTIEL. « Structure and motion from straight line segments ». *Pattern Recognition*, 33 :1295–1307, 2000.
- [161] I. MOON, J. MIURA, et Y. SHIRAI. « Automatic Extraction of Visual Landmarks for a Mobile Robot under Uncertainty of Vision and Motion ». Dans *IEEE International Conference on Robotics and Automation*, pages 1188–1193, 2001.

- [162] H. P. MORAVEC. « Sensor fusion in certainty grids for mobile robots ». *AI Magazine*, 9(2) :61–74, 1988.
- [163] H. P. MORAVEC et A. ELFES. « High resolution maps from wide angle sonar ». Dans *IEEE International Conference on Robotics and Automation*, pages 116–121, 1985.
- [164] E. M. MOUADDIB et B. MARHIC. « Geometrical Matching for Mobile Robot Localization ». *IEEE Transactions on Robotics and Automation*, 16(5) :542–552, 2000.
- [165] P. MOUTARLIER et R. CHATILA. « Stochastic multisensor data fusion for mobile robot localization and environment modeling ». Dans *International Symposium on Robotics Research*, pages 85–94, 1989.
- [166] S. MURATA et T. HIROSE. « Onboard locating system of autonomous vehicle ». Dans *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 228–234, 1989.
- [167] J. M. PEÑA, J. A. LOZANO, et P. LARRA NAGA. « An empirical comparison of four initialization methods for the K-means algorithm ». *Pattern Recognition Letters*, 20 :1027–1040, 1999.
- [168] K. NAGATANI, S. TACHIBANA, M. SOFUE, et Y. TANAKA. « Improvement of odometry for Omnidirectional vehicle using opticalflow information ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 468–473, 2000.
- [169] J. NEIRA, M. I. RIBEIRO, et J. D. TARDÓS. « Mobile robot localization and map building using monocular vision ». Dans *International Symposium on Intelligent Robotic Systems*, pages 275–284, 1997.
- [170] J. NEIRA, J. D. TARDÓS, J. HORN, et G. SCHMIDT. « Fusing Range and Intensity Images for Mobile Robot Localization ». *IEEE Transactions on Robotics and Automation*, 15(1) :76–84, February 1999.
- [171] A. J. MUÑOZ et J. GONZALEZ. « Two-dimensional Landmark-based position estimation from single image ». Dans *IEEE International Conference on Robotics and Automation*, pages 3709–3714, 1998.
- [172] G. ORIOLO, G. ULIVI, et M. VENDITTELLI. « Fuzzy maps : a new tool for mobile robot perception and planning ». *Journal of Robotic Systems*, 14(3) :179–197, 1997.
- [173] G. ORIOLO, G. ULIVI, et M. VENDITTELLI. « Real-time map building and navigation for autonomous robots in unknown environments ». *IEEE Transactions on Systems, Man, and Cybernetics - Part B : Cybernetics*, 28(3) :316–333, June 1998.
- [174] K. PARK, H. CHUNG, et J. G. LEE. « Dead reckoning navigation for autonomous mobile robots ». Dans *IFAC Intelligent Autonomous Vehicles*, pages 775–780, 1998.
- [175] J. R. PARKER. *Algorithms for Image Processing and Computer Vision*. John Wiley and Sons, New York, USA, 1996.
- [176] N. E. PEARS. « Feature extraction and tracking for scanning range sensors ». *Robotics and Autonomous Systems*, 33 :43–58, 2000.
- [177] S.-C. PEI et L.-G. LIOU. « Vehicle-type Motion Estimation by the Fusion of Image Points And Line Features ». *Pattern Recognition*, 31(3) :333–344, 1997.
- [178] J. A. PÉREZ, J. A. CASTELLANOS, J. M. M. MONTIEL, J. NEIRA, et J. D. TARDÓS. « Continuous Mobile Robot Localization : Vision vs. Laser ». Dans *IEEE International Conference on Robotics and Automation*, pages 2917–2923, 1999.
- [179] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, et B. P. FLANNERY. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, second édition, 1992.
- [180] U. RASCHKE et J. BORENSTEIN. « A Comparisson of Grid-type Map-building Techniques by Index of Performance ». Dans *IEEE International Conference on Robotics and Automation*, pages 1828–1832, 1990.
- [181] A. REINA et J. GONZALEZ. « Determining mobile robot orientation by aligning 2D segment maps ». Dans *3rd IFAC Symposium on Intelligent Autonomous Vehicles*, pages 632–637, 1998.
- [182] W. D. RENCKEN. « Concurrent localisation and map building for mobile robots using ultrasonic sensors ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2192–2197, 1993.
- [183] J. REUTER. « Mobile robot self-Localization using PDAB ». Dans *IEEE International Conference on Robotics and Automation*, pages 3512–3518, 2000.
- [184] P. RIVES, J.-L. SEQUEIRA, et P. LOURTIE. « Localization and map building for a mobile robot ». Dans *International Symposium on Experimental Robotics*, 1999.

- [185] I. T. RUIZ, Y. PETILLOT, D. M. LANE, et C. SALSON. « Feature extraction and data association for AUV concurrent mapping and localization ». Dans *IEEE International Conference on Robotics and Automation*, pages 2785–2790, 2001.
- [186] B. SCHIELE et J. CROWLEY. « A comparison of position estimation techniques using occupancy grids ». Dans *IEEE International Conference on Robotics and Automation*, pages 1628–1634, 1994.
- [187] D. SCHULZ, W. BURGARD, D. FOX, et A. B. CREMERS. « Tracking Multiple Moving Targets With A Mobile Robot Using Particle Filters and Statistical Data Association ». Dans *IEEE International Conference on Robotics and Automation*, pages 1665–1670, 2001.
- [188] J. De SCHUTTER, J. De GEETER, T. LEFEBVRE, et H. BRUYNINCKX. « Kalman Filters : A Tutorial ». *Journal A - Benelux Quarterly Journal on Automatic Control*, 40(4) :538–546, December 1999.
- [189] S. SE, D. LOWE, et J. LITTLE. « Vision-based mobile robot localization and mapping using scale-invariant features ». Dans *IEEE International Conference on Robotics and Automation*, pages 2051–2058, 2001.
- [190] R. SHPILMAN et V. BRAILOVSKY. « Fast and robust techniques for detecting straight line segments using local models ». *Pattern Recognition Letters*, 20(8) :865–877, August 1999.
- [191] A. SIADAT, A. KARKE, S. KLAUSMANN, M. DUFAUT, et R. HUSSON. « An Optimized Segmentation Method for a 2-D Laser Scanner Applied to Mobile Robot Navigation ». Dans *Proc. of 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications*, pages 153–158, June 1997.
- [192] K. T. SIMSARIAN, T. J. OLSON, et N. NANDHAKUMAR. « View-invariant regions and mobile robot self-localization ». *IEEE Transactions on Robotics and Automation*, 12(5) :810–816, October 1996.
- [193] P. SKRZYPCZYNSKI. « Building geometrical map of environment using IR range finder data ». Dans *Intelligent Autonomous Systems*, pages 408–412, 1995.
- [194] P. SKRZYPCZYNSKI. « Environment modelling using optical scanner data ». Dans *IFAC Symposium on Robot Control*, pages 187–192, 1997.
- [195] R. SMITH, M. SELF, et P. CHEESEMAN. « A stochastic map for uncertain spatial relationships ». Dans MIT PRESS, éditeur, *4th International Symposium on Robotics Research*, 1987.
- [196] R. SMITH, M. SELF, et P. CHEESEMAN. Estimating uncertain spatial relationships in robotics. Dans I. J. COX et G. T. WILFONG, éditeurs, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.
- [197] R. SWAROOP, J. D. BROWNLOW, G. R. ASHWORTH, et W. R. WINTER. « BIVARIATE NORMAL, CONDITIONAL AND RECTANGULAR PROBABILITIES : A COMPUTER PROGRAM WITH APPLICATIONS ». Rapport Technique NASA-TM-81350, NASA Dryden Flight Research Center, Edwards, CA, 1980. Disponible en ligne sur www.dfr.nasa.gov/DTRS/1980/citation.html.
- [198] R. TALLURI et J. K. AGGARWAL. « Mobile Robot Self-Location using Model-Image Feature Correspondence ». *IEEE Transactions on Robotics and Automation*, 12(1) :63–77, February 1996.
- [199] W. I. TAM, K. N. PLATANOTIS, et D. HATZINAKOS. « An adaptive Gaussian Sum Algorithm for Radar Tracking ». *Signal Processing*, 77 :85–104, 1999.
- [200] S. THRUN, M. BEETZ, M. BENNEWITZ, W. BURGARD, A. B. CREMERS, F. DELLAERT, D. FOX, D. HAEHNEL, C. ROSENBERG, N. ROY, J. SCHULTE, et D. SCHULZ. « Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva ». *International Journal of Robotics Research*, 19(11) :972–999, 2000.
- [201] S. THRUN et A. BÜCKEN. « Integrating grid-based and topological maps for mobile robot navigation ». Dans *AAAI Thirteenth National Conference on Artificial Intelligence*, pages 944–950, 1996.
- [202] S. THRUN, D. FOX, et W. BURGARD. « A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots ». *Machine Learning*, 31 :29–53, 1998.
- [203] S. THRUN, D. FOX, et W. BURGARD. « Probabilistic mapping of an environment by a mobile robot ». Dans *IEEE International Conference on Robotics and Automation*, pages 1546–1551, 1998.
- [204] S. THRUN, D. FOX, W. BURGARD, et F. DELLAERT. « Robust Monte Carlo localization for mobile robots ». *Artificial Intelligence*, 128(1-2) :99–141, May 2001.
- [205] S. THRUN, D. FOX, W. BURGARD, et F. DELLAERT. « Robust Monte Carlo Localization for Mobile Robots ». *Artificial Intelligence Journal*, 128(1-2) :99–141, May 2001.

- [206] S. THRUN, J.-S. GUTMANN, D. FOX, W. BURGARD, et B. J. KUIPERS. « Integrating Topological and Metric Maps for Mobile Robot Navigation : A Statistical Approach ». Dans *AAAI Conference on Innovative Applications of Artificial Intelligence*, pages 989–995, 1998.
- [207] J. K. UHLMANN, S. J. JULIER, et M. CSORBA. Nondivergent simultaneous map-building and localization using covariance intersection. Dans Scott A. SPEIGLE, éditeur, *Navigation and Control Technologies for Unmanned Systems II*, pages 2–11. SPIE, 1997.
- [208] S. UMEYAMA. « Least-Squares Estimation of Transformation Parameters Between Two Point Patterns ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4) :376–380, April 1991.
- [209] R. van der MERWE, A. DOUCET, N. de FREITAS, et E. WAN. The Unscented Particle Filter. Dans T. K. LEEN, T. G. DIETTERICH, et V. TRESP, éditeurs, *Advances in Neural Information Processing Systems*. MIT Press, 2000.
- [210] J. VANDORPE, H. Van BRUSSEL, et H. XU. « Exact Dynamic Map Building for a Mobile Robot using Geometrical Primitives Produced by a 2D Range Finder ». Dans *IEEE International Conference on Robotics and Automation*, pages 901–908, April 1996.
- [211] S. J. VESTLI, N. TSCHICHOLD-GÜRMAN, M. ADAMS, et S. SULZBERGER. « Amplitude Modulated Optical Range Data Analysis in Mobile Robotics ». Dans *IEEE International Conference on Robotics and Automation*, pages 3243–3248, 1993.
- [212] A. C. VICTORINO, P. RIVES, et J.-J. BORRELLY. « Localization and Map Building Using a Sensor-Based Control Strategy ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 937–942, 2000.
- [213] A. C. VICTORINO, P. RIVES, et J.-J. BORRELLY. « Mobile robot navigation using a Sensor-Based Control Strategy ». Dans *IEEE International Conference on Robotics and Automation*, pages 2753–2758, 2001.
- [214] N. VLASSIS et B. J. A. KRÖSE. « Robot Environment modeling via principal component regression ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.
- [215] H.-J. von der HARDT, R. HUSSON, et D. WOLF. « An Automatic calibration method for a multisensor system : application to a mobile robot localization system ». Dans *IEEE International Conference on Robotics and Automation*, pages 3141–3146, 1998.
- [216] C. M. WANG. « Location estimation and uncertainty analysis for mobile robots ». Dans *IEEE International Conference on Robotics and Automation*, pages 1230–1235, 1988.
- [217] R. WASHINGTON. « On-Board Real-Time State and Fault Identification for Rovers ». Dans *IEEE International Conference on Robotics and Automation*, pages 1175–1181, 2000.
- [218] P. WECKESSER et R. DILLMANN. « Sensor-Fusion of Intensity- and Laserrange-Images ». Dans *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 501–508, 1996.
- [219] R. WEHRENS, H. PUTTER, et L. M. C. BUYDENS. « The bootstrap : a tutorial ». *Chemometrics and Intelligent Laboratory Systems*, 54 :35–52, 2000.
- [220] G. WEIß, C. WETZLER, et E. von PUTTKAMER. « Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 595–601, 1994.
- [221] J. WENG, T. S. HUANG, et N. AHUJA. « Motion and Structure from Line Correspondences : Closed-Form Solution, Uniqueness, and Optimization ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3) :318–336, March 1992.
- [222] S. WILLIAMS, G. DISSANAYAKE, et H. DURRANT-WHYTE. « Constrained Initialisation of the Simultaneous Localisation and Mapping Algorithm ». Dans *3rd International Conference and Field and Service Robotics*, 2001.
- [223] X. XU et S. NEGAHDARIPOUR. « Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwater vehicles ». Dans *IEEE International Conference on Robotics and Automation*, pages 2759–2766, 2001.

- [224] T. Y. YOUNG et K.-S. FU. *Handbook of Pattern Recognition and Image Processing*. Academic Press, London, UK, 1986.
- [225] W. P. YU, G. W. CHU, et M. J. CHUNG. « A robust line extraction method by unsupervised line clustering ». *Pattern Recognition*, 32(4) :529–546, April 1999.
- [226] H. ZHA, K. TANAKA, et T. HASEGAWA. « Detecting Changes in a Dynamic Environment for Updating its Maps by Using a Mobile Robot ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1729–1734, 1997.
- [227] L. ZHANG et B. K. GHOSH. « Line segment based map building and localization using 2D Laser rangefinder ». Dans *IEEE International Conference on Robotics and Automation*, pages 2538–2543, 2000.
- [228] Z. ZHANG. « Estimating Motion and Structure from Correspondences of Line Segments between Two Perspective Images ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12) :1129–1138, Dec. 1995.
- [229] Z. ZHANG. « A Flexible New Technique for Camera Calibration ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11) :1330–1334, November 2000.
- [230] H. ZHUANG et R. SUDHAKAR. « Simultaneous Rotation and Translation Fitting of Two 3-D Point Sets ». *IEEE Transactions on Systems, Man, and Cybernetics - Part B : Cybernetics*, 27(1) :127–131, Feb. 1997.

Annexes

Repères et transformations dans l'espace 2-D

Nous avons vu dans le Chapitre 3 que des repères 3-D ont été définis pour permettre une meilleure représentation de la configuration du système multicapteur du robot Omni. Dans cette annexe, nous définissons les repères 2-D de l'espace cartésien qui résultent de la projection des repères capteurs dans le plan de balayage du télémètre laser. En effet, c'est dans ce plan que nous estimons la position du robot.

A.1 Repères principaux du robot Omni

Avec la notation employée, un repère est représenté par $\mathcal{R}$. Pour la localisation du robot Omni, les trois repères utilisés sont (voir Figure A.1) :

- $\mathcal{R}^G$: Le repère global qui représente notre référence absolue ;
- $\mathcal{R}^R$: Le repère local ou repère robot qui est attaché au centre géométrique des axes de direction des roues ;
- $\mathcal{R}^L$: Le repère du système multicapteur qui coïncide avec les axes $\mathcal{X}^L$ et $\mathcal{Y}^L$ du repère 3-D du télémètre laser.

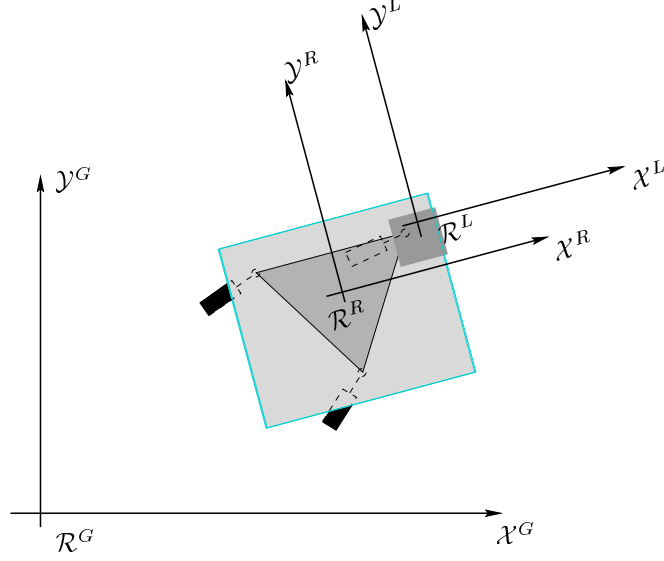


FIG. A.1 – Repères principaux du robot Omni.

A.2 Transformation de coordonnées

Deux repères $\mathcal{R}^A$ et $\mathcal{R}^B$ de l'espace euclidien 2-D sont reliés par un vecteur de paramètres $\mathbf{z} = (x, y, \theta)^T$ tel que la relation entre les coordonnées (x^A, y^A) dans $\mathcal{R}^A$ et (x^B, y^B) dans $\mathcal{R}^B$ de tout point statique de cet espace s'écrit :

$$\begin{pmatrix} x^B \\ y^B \end{pmatrix} = \mathbf{R}(\theta) \cdot \left\{ \begin{pmatrix} x^A \\ y^A \end{pmatrix} - \mathbf{t} \right\}, \quad (\text{A.1})$$

où $\mathbf{t} = \begin{pmatrix} x & y \end{pmatrix}^T$ et

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \quad (\text{A.2})$$

est une matrice orthogonale qui représente la rotation dans l'espace 2-D. (x, y) sont les coordonnées de l'origine du repère $\mathcal{R}^B$ dans le repère $\mathcal{R}^A$, et θ est l'angle entre les axes $\mathcal{X}^B$ et $\mathcal{X}^A$ de ces repères (Figure A.2). Les propriétés de la matrice orthogonale $\mathbf{R}(\theta)$ permettent d'écrire les relations suivantes :

$$\mathbf{R}^T(\theta) = \mathbf{R}^{-1}(\theta) \implies \mathbf{R}(\theta) \cdot \mathbf{R}^T(\theta) = \mathbf{I}_2, \quad (\text{A.3})$$

$$\mathbf{R}^T(\theta) = \mathbf{R}(-\theta), \quad (\text{A.4})$$

$$\mathbf{R}(\theta) \cdot \mathbf{R}(\varphi) = \mathbf{R}(\theta + \varphi), \quad (\text{A.5})$$

$$\mathbf{R}(\theta) \cdot \mathbf{R}^T(\varphi) = \mathbf{R}(\theta - \varphi), \quad (\text{A.6})$$

$$\frac{\partial \mathbf{R}(\theta)}{\partial \theta} = \begin{pmatrix} -\sin(\theta) & \cos(\theta) \\ -\cos(\theta) & -\sin(\theta) \end{pmatrix} = \mathbf{R}(\pi/2) \cdot \mathbf{R}(\theta). \quad (\text{A.7})$$

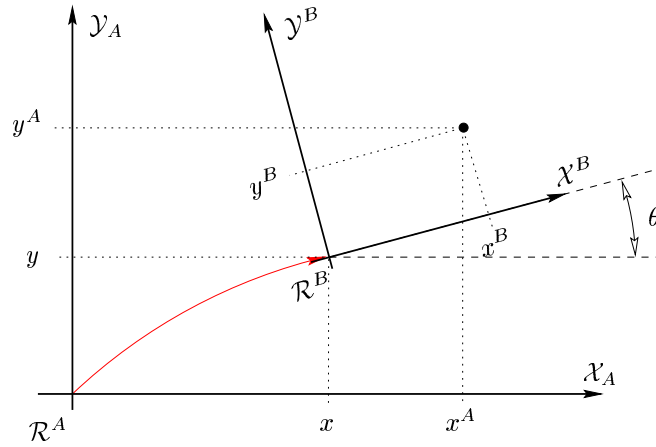


FIG. A.2 – Positionnement entre repères.

A.3 Relation entre repères

A.3.1 Position du robot dans un repère de référence

Dans la section précédente nous avons vu que deux repères quelconques appartenant au même espace euclidien 2-D sont reliés par un vecteur de paramètres $\mathbf{z} = (x, y, \theta)^T$. En robotique mobile d'intérieur, où pour la majeure partie des applications, l'hypothèse de navigation sur sol plan est acceptable, il est nécessaire d'estimer le positionnement de l'engin dans un repère de référence $\mathcal{R}^G$, c'est à dire, sa position (x, y) et son cap θ . Ainsi, nous notons $\mathbf{z} = (x, y, \theta)^T$ la position du repère robot $\mathcal{R}^R$ dans ce repère global $\mathcal{R}^G$. Nous complétons cette définition pour y incorporer l'incertitude de positionnement. Nous supposons que $\mathbf{z}$ est perturbé par un bruit aléatoire de distribution $\mathcal{N}(\mathbf{0}, \Lambda_{\mathbf{z}})$. La position du robot est alors représentée par le duplet :

$$Z = \{\mathbf{z}, \Lambda_{\mathbf{z}}\}. \quad (\text{A.8})$$

A.3.2 Position du repère capteur dans le repère robot

Le repère local $\mathcal{R}^R$ et le repère capteur $\mathcal{R}^L$ sont reliés par $Z_L^R = \{\mathbf{z}_L^R, \mathbf{0}_3\}$, où $\mathbf{z}_L^R$ définit la position de $\mathcal{R}^L$ par rapport à $\mathcal{R}^R$. La matrice de covariance $\mathbf{0}_3$ indique que nous n'associons pas d'incertitude à $\mathbf{z}_L^R$. Les composantes de $\mathbf{z}_L^R = (x_L^R, y_L^R, \theta_L^R)^T$ ont été déterminées approximativement à partir de simples mesures et en tenant en compte de la géométrie du robot. Les valeurs obtenues sont

$$x_L^R = 0,3214m \quad y_L^R = 0,0730 \text{ m} \quad \theta_L^R = 0^\circ. \quad (\text{A.9})$$

Bien entendu, on peut contester cette approche peu scientifique de détermination de la transformation entre le repère capteur et le repère robot. En effet, une détermination précise de $\mathbf{z}_L^R$ nécessiterait une procédure d'identification et l'utilisation d'instruments sophistiqués de mesure 3-D. Toutefois, nous avons estimé que le temps nécessaire pour réaliser un tel calibrage serait trop

important, et nous avons donc opté pour une solution plus simple. De toute façon, au cours des expérimentations réalisées, on n'a pas noté d'erreur importante sur $\mathbf{z}_L^R$ qui serait traduite par un biais d'estimation. Cela permet de considérer que les valeurs numériques retenues pour les composantes de $\mathbf{z}_L^R$ sont correctes. De plus, en supposant que $\mathbf{z}_L^R$ n'est pas perturbée par des erreurs de mesure, on simplifie les opérations de changement de repère entre $\mathcal{R}^R$ et $\mathcal{R}^L$.

Représentation et propagation de l'incertitude

B.1 Introduction

Comme pour la majeure partie des travaux en robotique mobile, notre choix pour la modélisation des sources d'incertitudes s'est porté sur les processus stochastiques gaussiens. Les outils disponibles dans la littérature sont nombreux, et ils s'intègrent facilement à notre approche mathématique. Notre objectif ici n'est pas seulement de les présenter, mais aussi de vérifier leurs limites d'application. Dans cette annexe, nous partons du principe que les notions de base en probabilité et en statistiques sont connues.

B.2 Processus gaussiens

Soit X une variable aléatoire (v.a.) scalaire dont la fonction densité de probabilité est donnée par

$$f_X(u) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(u - \bar{x})^2}{2\sigma^2}\right). \quad (\text{B.1})$$

Nous disons que X est une v.a. de distribution gaussienne de moyenne $\bar{x}$ et variance σ^2 . Suivant la notation adoptée, $X \sim \mathcal{N}(\bar{x}, \sigma^2)$. Soit x une réalisation de X , *i.e.*, x est tiré au hasard de la distribution de X . Lorsque la fonction de densité de probabilité est intégrée de $-\infty$ à a , nous

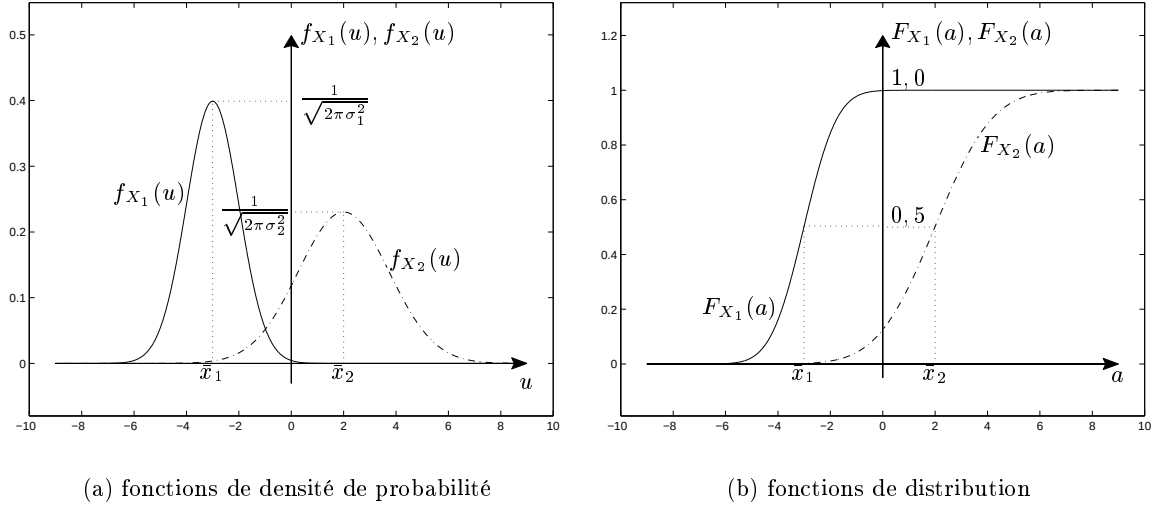


FIG. B.1 – Fonctions qui caractérisent la distribution gaussienne.

obtenons la probabilité que x soit inférieure à une valeur a , *i.e.*,

$$F_X(a) = \int_{-\infty}^a f_X(u) \cdot du = \Pr(x < a). \quad (\text{B.2})$$

L'opérateur $\mathcal{E}\{v\}$ représente l'espérance sur une réalisation v de V . Pour un processus gaussien X :

$$\mathcal{E}\{x\} = \bar{x}, \quad \mathcal{E}\{(x - \bar{x})^2\} = \sigma^2. \quad (\text{B.3})$$

La Figure B.1 illustre la forme de ces fonctions pour $X_1 \sim \mathcal{N}(-3, 1)$ et $X_2 \sim \mathcal{N}(2, 3)$.

Supposons maintenant que X soit un vecteur de variables aléatoires, *i.e.*, $X = [X_1, X_2, \dots, X_n]^T$. Ainsi, $X \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{\Lambda})$, où $\bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$ et $\mathbf{\Lambda}$ sont respectivement la moyenne et la matrice de covariance de la distribution de X . La fonction de densité de probabilité de X est donnée par [104] :

$$f_X(\mathbf{u}) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(\mathbf{\Lambda})}} \cdot \exp\left(-\frac{1}{2}(\mathbf{u} - \bar{\mathbf{x}})^T \cdot \mathbf{\Lambda}^{-1} \cdot (\mathbf{u} - \bar{\mathbf{x}})\right). \quad (\text{B.4})$$

Soit $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ une réalisation de X . Alors,

$$\mathcal{E}\{\mathbf{x}\} = \bar{\mathbf{x}}, \quad \mathcal{E}\{(\mathbf{x} - \bar{\mathbf{x}}) \cdot (\mathbf{x} - \bar{\mathbf{x}})^T\} = \mathbf{\Lambda} = \begin{pmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} & \cdots & \sigma_{x_1 x_n} \\ \sigma_{x_2 x_1} & \sigma_{x_2}^2 & \cdots & \sigma_{x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_n x_1} & \sigma_{x_n x_2} & \cdots & \sigma_{x_n}^2 \end{pmatrix}, \quad (\text{B.5})$$

où $\sigma_{x_i x_j} = \sigma_{x_j x_i} = \mathcal{E}\{(x_i - \bar{x}_i) \cdot (x_j - \bar{x}_j)\}$ pour $i \neq j$ sont les covariances entre les processus X_i et X_j , et $\sigma_{x_i}^2 = \sigma_{x_i x_i} = \mathcal{E}\{(x_i - \bar{x}_i)^2\}$. $\mathbf{\Lambda}$ est une matrice symétrique et définie positive, *i.e.*,

$\mathbf{u}^T \cdot \mathbf{\Lambda} \cdot \mathbf{u} \geq 0$ pour n'importe quel vecteur $\mathbf{u} \in \mathbb{R}^n$. Si les processus qui composent X ne sont pas corrélés, alors $\mathbf{\Lambda}$ est une matrice diagonale. $\mathbf{\Lambda}$ peut être décomposée sous la forme suivante :

$$\mathbf{\Lambda} = \sum_{i=1}^n e_i \cdot \mathbf{v}_i \cdot \mathbf{v}_i^T \quad (\text{B.6})$$

où e_i et $\mathbf{v}_i$ sont respectivement les i -ième valeur propre et vecteur propre de $\mathbf{\Lambda}$. En fait, lorsque $\mathbf{\Lambda}$ est estimé à partir d'un nuage de points, e_i est la composante de la variance de la distribution suivant l'axe $\mathbf{v}_i$. Cette propriété est normalement exploitée pour effectuer la régression linéaire d'un nuage de points. La droite qui s'ajuste le mieux au nuage de points a la même direction que le vecteur propre correspondant à la plus grande valeur propre de $\mathbf{\Lambda}$. La plus petite valeur propre correspond à la variance de la distance des points du nuage par rapport à la droite.

Considérons, en particulier, le cas d'une v.a. X de dimension $n = 2$. A maintes occasions, nous illustrons la distribution de l'incertitude autour de $\bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2]^T$ sous forme d'une *ellipse de covariance*. C'est en fait une courbe de densité de probabilité f_X constante. Si nous appliquons les transformations de normalisation $u_1 = (x_1 - \bar{x}_1)/\sigma_{x_1}$ et $u_2 = (x_2 - \bar{x}_2)/\sigma_{x_2}$, nous obtenons

$$f_U(u_1, u_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \cdot \exp\left(-\frac{1}{2}\mathbf{u}^T \cdot \mathbf{\Lambda}_u^{-1} \cdot \mathbf{u}\right) \quad (\text{B.7})$$

où $\rho = \sigma_{x_1 x_2}/(\sigma_{x_1} \cdot \sigma_{x_2}) = \sigma_{u_1 u_2}$ est le coefficient de corrélation entre X_1 et X_2 et

$$\mathbf{\Lambda}_u = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \quad (\text{B.8})$$

est la matrice de covariance du processus normalisé U . La courbe de densité de probabilité constante est donc formée par tous les points $\mathbf{u} = (u_1, u_2)^T$ qui satisfont

$$\mathbf{u}^T \cdot \mathbf{\Lambda}_u^{-1} \cdot \mathbf{u} = \frac{1}{(1-\rho^2)} (u_1^2 + u_2^2 - 2u_1 u_2 \rho) = c. \quad (\text{B.9})$$

Ces points forment une ellipse centrée à l'origine du plan $u_1 \times u_2$. Son correspondant dans le plan $x_1 \times x_2$ est donné par

$$\frac{(x_1 - \bar{x}_1)^2}{\sigma_{x_1}^2} - 2\rho \frac{(x_1 - \bar{x}_1)}{\sigma_{x_1}} \frac{(x_2 - \bar{x}_2)}{\sigma_{x_2}} + \frac{(x_2 - \bar{x}_2)^2}{\sigma_{x_2}^2} = (1 - \rho^2) \cdot c$$

qui est une ellipse centrée en $(\bar{x}_1, \bar{x}_2)$, de demi grand axe r_1 , de demi petit axe r_2 et d'angle θ avec l'axe x_1 donnés par

$$\tan(2\theta) = \frac{2\rho \cdot \sigma_{x_1} \cdot \sigma_{x_2}}{\sigma_{x_1}^2 - \sigma_{x_2}^2}, \quad (\text{B.10})$$

$$r_1^2 = c \cdot \frac{\sigma_{x_1}^2 \sigma_{x_2}^2 (1 - \rho^2)}{\sigma_{x_2}^2 \cdot \cos^2(\theta) - 2\rho \cdot \sigma_{x_1} \cdot \sigma_{x_2} \sin(\theta) \cdot \cos(\theta) + \sigma_{x_1}^2 \cdot \sin^2(\theta)}, \quad (\text{B.11})$$

$$r_2^2 = c \cdot \frac{\sigma_{x_1}^2 \sigma_{x_2}^2 (1 - \rho^2)}{\sigma_{x_2}^2 \cdot \sin^2(\theta) + 2\rho \cdot \sigma_{x_1} \cdot \sigma_{x_2} \sin(\theta) \cdot \cos(\theta) + \sigma_{x_1}^2 \cdot \cos^2(\theta)}. \quad (\text{B.12})$$

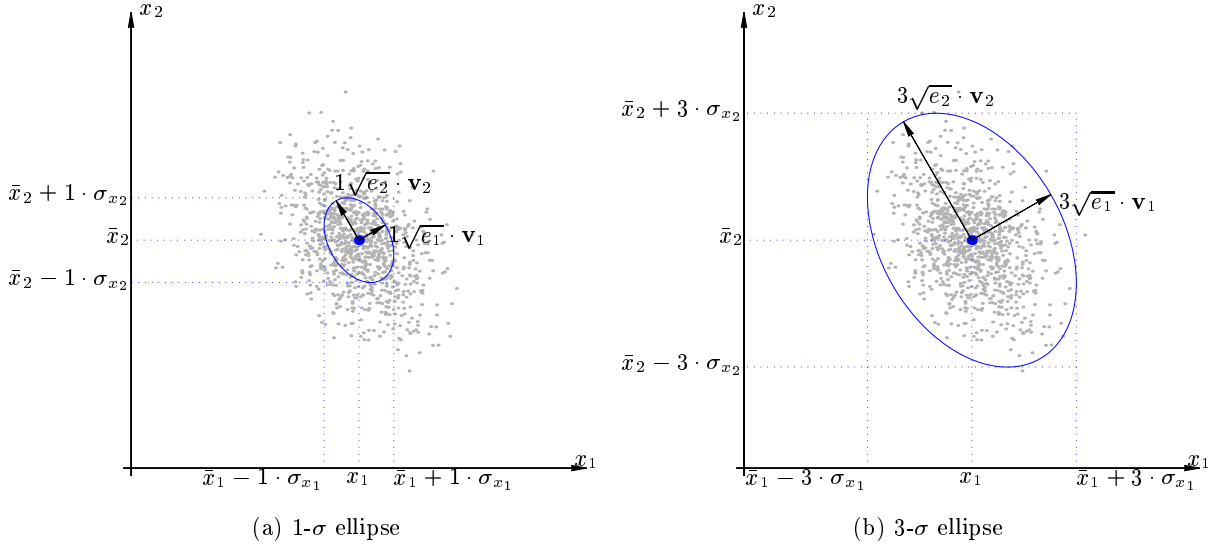


FIG. B.2 – Ellipses de covariance de la distribution gaussienne bi-variable.

La normalisation est un artifice qui permet de relier l'équation de l'ellipse de covariance au pourcentage p de la distribution X qu'elle entoure. La constante c de l'équation (B.9) et p sont reliés par les relations suivantes [197] :

$$c = \log\left(\frac{1}{(1-p)^2}\right) \quad \text{ou} \quad p = 1 - \frac{1}{\sqrt{e^c}}, \quad (\text{B.13})$$

où $e \sim 2,71828$.

Pour illustrer l'utilisation de cet outil, nous proposons l'exemple suivant.

Exemple B.1: Soit $X \sim \mathcal{N}(\bar{\mathbf{x}}, \Lambda)$ une v.a. telle que $\bar{\mathbf{x}} = \begin{pmatrix} 5,0 & 3,0 \end{pmatrix}^T$. Λ est présenté sous la forme décomposée (B.6). Ses valeurs propres sont $e_1 = 4$ et $e_2 = 9$, et les vecteurs propres correspondants sont $\mathbf{v}_1 = \begin{pmatrix} \cos(\pi/6) & \sin(\pi/6) \end{pmatrix}^T$ et $\mathbf{v}_2 = \begin{pmatrix} -\sin(\pi/6) & \cos(\pi/6) \end{pmatrix}^T$. Nous avons dessiné sur les graphiques (a) et (b) de la Figure B.2 les ellipses de covariance correspondant à Λ pour $c = 1$ et $c = 9$, respectivement. Nous présentons aussi 1000 réalisations de X , qui sont illustrées sous la forme de points gris. En fait, cet exemple illustre la signification de la constante c . Il suffit de regarder les projections de l'ellipse suivant les axes x_1 et x_2 . Ces projections s'étendent à l'intérieur des intervalles $[\bar{x}_1 - \sqrt{c}\sigma_{x_1}; \bar{x}_1 + \sqrt{c}\sigma_{x_1}]$ et $[\bar{x}_2 - \sqrt{c}\sigma_{x_2}; \bar{x}_2 + \sqrt{c}\sigma_{x_2}]$. Ces sont les $\sqrt{c}\sigma$ intervalles de confiance des composantes de la v.a. X . Les demi axes de l'ellipse sont donc $\sqrt{c \cdot e_1} \cdot \mathbf{v}_1$ et $\sqrt{c \cdot e_2} \cdot \mathbf{v}_2$. $\diamond$

Notre intérêt se porte principalement sur l'ellipse à $c = 9$ (Figure B.2(b)), qui représente $p = 0,9889$ (i.e. 98,89 %) de la distribution. Nous l'appelons "ellipse 3σ ", et elle est utilisée au cours de notre étude pour représenter graphiquement l'incertitude. Dans d'autres cas, nous estimons la matrice de covariance d'un modèle gaussien à partir de ses demi-axes lorsque ils correspondent à des intervalles limités connus. Par exemple, si nous avons une variable scalaire x bornée à $2 \leq x \leq 8$,

nous estimons la distribution gaussienne équivalente de cette variable par $\bar{x} = 5$, d'écart type $\sigma = 1$. Cette distribution retient 98,89 % de la population dans l'intervalle 3σ : $\bar{x} - 3\sigma \leq x \leq \bar{x} + 3\sigma$, qui est le même intervalle bornée de x . Pour cela, nous utilisons un raisonnement dit “ 3σ ” pour relier un intervalle à une distribution gaussienne.

B.3 Représentation de l'incertitude paramétrique

Une des caractéristiques d'un processus gaussien est qu'il est complètement caractérisé par deux paramètres : le vecteur des moyennes et la matrice covariance. Cela facilite son utilisation pour la modélisation d'incertitude, étant donné le nombre réduit de paramètres à maintenir. De plus, les outils de propagation d'incertitude et de mise en correspondance sont très simples à utiliser. Au cours de notre étude, nous avons présenté des entités géométriques (cartes d'environnement, indices géométriques ou algébriques, variables d'état, *etc.*) toujours associées à des processus de distribution gaussienne. Nous utilisons la notation suivante pour représenter une entité quelconque P :

$$P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}. \quad (\text{B.14})$$

P est ainsi supposée être une v.a. de distribution gaussienne dont $\mathbf{p}$ est une réalisation ayant pour matrice de covariance $\Lambda_{\mathbf{p}}$.

Parmi les autres formes de représentation de l'incertitude appliquées récemment à la localisation des robots mobiles, on peut citer l'analyse par intervalles [103, 121, 146], le mélange de gaussiennes [12] et les filtres à particules [205, 106]. Ces approches sont bien adaptées lorsque l'incertitude est soumise à des transformations non-linéaires, ou lorsque la vraie distribution de l'erreur est multimodale. Des résultats prometteurs ont été présentés dans les références citées.

B.4 Propagation de l'incertitude

Dans un grand nombre d'applications concernant des systèmes physiques, il est nécessaire de modéliser l'incertitude des mesures et de la propager dans le processus de calcul des variables d'état. Cette démarche a été utilisée pour améliorer le diagnostic [217][53] et la détection de défauts [82][46]. Mais cela suppose que la représentation de l'incertitude ne soit pas dégénérée par des erreurs de propagation dans le processus. Nous présentons dans cette section deux approches de propagation d'incertitude : la propagation par linéarisation, et l'échantillonnage sélectif.

B.4.1 Propagation par linéarisation

Soit un vecteur paramétrique $\mathbf{y} = (y_1, \dots, y_m)^T$ relié à $\mathbf{x} = (x_1, \dots, x_n)^T$ par une transformation déterministe notée

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})). \quad (\text{B.15})$$

Nous avons ainsi une relation analytique entre $\mathbf{y}$ et $\mathbf{x}$. Si $\mathbf{x}$ représente les paramètres d'une grandeur incertaine $X = \{\mathbf{x}, \Lambda_{\mathbf{x}}\}$, $Y = \{\mathbf{y}, \Lambda_{\mathbf{y}}\}$ sera aussi incertaine. Son incertitude représentée par $\Lambda_{\mathbf{y}}$ résulte de la propagation de celle de X à travers la transformation $\mathbf{f}(\cdot)$. L'incertitude de Y est généralement obtenue par linéarisation au premier ordre de $\mathbf{f}(\cdot)$ autour de la valeur nominale $\mathbf{x}$, *i.e.*,

$$\Delta \mathbf{y} = \nabla_{\mathbf{x}} \mathbf{f} \cdot \Delta \mathbf{x} + O(1). \quad (\text{B.16})$$

où $\Delta \mathbf{y}$ est la variation de $\mathbf{y}$ due à une variation $\Delta \mathbf{x}$ autour de $\mathbf{x}$, $O(1)$ est l'erreur de l'approximation.

En négligeant l'erreur de l'approximation (B.16), nous pouvons montrer que

$$\Lambda_{\mathbf{y}} = \nabla_{\mathbf{x}} \mathbf{f} \cdot \Lambda_{\mathbf{x}} \cdot \nabla_{\mathbf{x}}^T \mathbf{f}. \quad (\text{B.17})$$

Cette formule est très connue et largement utilisée pour propager $\Lambda_{\mathbf{x}}$ vers $\Lambda_{\mathbf{y}}$. Cependant, sa validité dépend du degré de linéarité de $\mathbf{f}$ par rapport à $\mathbf{x}$, ainsi que de l'ampleur de l'incertitude $\Lambda_{\mathbf{x}}$. L'exemple ci-dessous illustre bien l'application et les limitations de cet outil.

Exemple B.2: *Considérons la propagation d'incertitude lors de la transformation de coordonnées polaires $\mathbf{p} = (r, \theta)^T$ d'une cible en coordonnées cartésiennes $\mathbf{c} = (x, y)^T$:*

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cdot \cos(\theta) \\ r \cdot \sin(\theta) \end{pmatrix}.$$

Supposons que le télémètre laser fournisse une paire de mesures ($r = 1 \text{ m}, \theta = 90^\circ$) non corrélées ayant pour incertitude

$$\Lambda_{\mathbf{p}} = \begin{pmatrix} (0,03 \text{ m})^2 & 0 \\ 0 & (1,0^\circ)^2 \end{pmatrix}.$$

La Figure B.3(a) présente l'ellipse d'incertitude 3σ correspondant à la matrice $\Lambda_{\mathbf{c}}$ estimée par propagation de l'incertitude linéarisante (en tracé continu). Pour nous aider dans l'évaluation du résultat obtenu, nous présentons aussi une estimée de $\Lambda_{\mathbf{c}}$ obtenue par simulation de Monte Carlo (en tracé non-continu), dont les 10000 particules simulées sont aussi illustrés. Nous utilisons la technique de Monte Carlo comme référence parce qu'elle nous permet d'avoir une idée approximative de la vraie distribution de $\mathbf{c}$. Nous pouvons voir que l'approximation par linéarisation semble être satisfaisante dans le sens où elle ne diffère pas beaucoup de celle utilisant les particules. Cela est dû au fait que la bonne précision du capteur télémétrique est suffisante pour que la linéarisation soit une approximation valide.

Maintenant, supposons que le capteur utilisé soit un sonar, qui donne la même mesure que le télémètre laser, mais avec une incertitude angulaire bien plus importante [113] :

$$\Lambda_{\mathbf{p}} = \begin{pmatrix} (0,02 \text{ m})^2 & 0 \\ 0 & (15,0^\circ)^2 \end{pmatrix}.$$

Dans ce cas, les résultats de simulation sont montrés sur la Figure B.3(b). Nous pouvons voir que cette fois-ci que la distribution des particules n'est plus gaussienne. Ceci est dû à la grande incertitude du capteur qui ne permet pas de valider la linéarisation. En conséquence, la distribution gaussienne estimée par linéarisation est inconsistante. $\diamond$

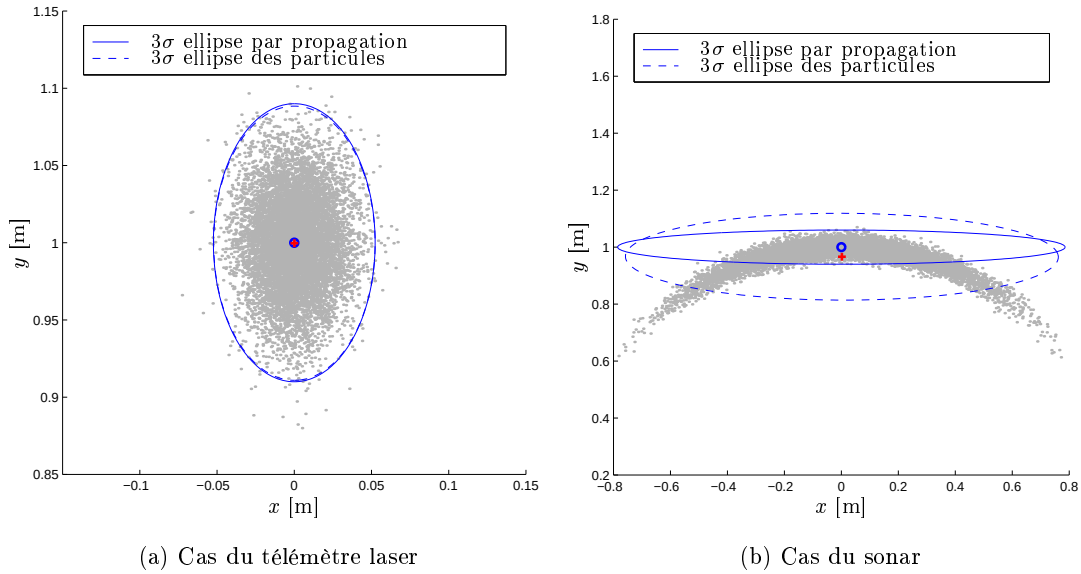


FIG. B.3 – Exemple d'application de la méthode de propagation d'incertitude par linéarisation.

En conclusion, la technique présentée ici est fondée sur des modèles linéarisés qui relient les variables d'intérêt. Nous avons vu par l'Exemple B.2 que la validité de cette technique dépend du degré de linéarité du modèle, et de l'amplitude de l'erreur.

Une étude intéressante sur la propagation d'incertitudes gaussiennes par linéarisation a été réalisé par Hartley & Zisserman [94]. Ils présentent des techniques adaptées aux cas où la transformation est connue sous une forme directe $\mathbf{y} = \mathbf{f}(\mathbf{x})$, comme dans le cas que nous venons de décrire, ou sous une forme inverse $\mathbf{y} = \mathbf{f}^{-1}(\mathbf{x})$.

B.4.1.1 Approche spécifique au problème d'estimation optimale

Un autre outil de propagation d'incertitude gaussienne que nous avons utilisé dans notre étude a été proposé par Haralick [93]. Cet outil est adapté à l'estimation de la matrice de covariance Λ_θ d'un vecteur de paramètres θ relié à N variables indépendantes $\mathbf{x}_n$, qui minimise un critère de la forme :

$$J = \sum_{n=1}^N d^2(\theta, \mathbf{x}_n) \quad (\text{B.18})$$

Les composantes de $\mathbf{x}_n$ peuvent être des mesures, où même les résidus d'estimation, et $d(\theta, \mathbf{x}_n)$ est une fonction scalaire de distance. Halalick montre qu'en linéarisant $\mathbf{g} = \partial J / \partial \theta$, Λ_θ peut être estimée par

$$\Lambda_\theta = \left(\frac{\partial \mathbf{g}}{\partial \theta} \right)^{-1} \cdot \left\{ \sum_{n=1}^N \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}_n} \right)^T \cdot \Lambda_{\mathbf{x}_n} \cdot \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}_n} \right) \right\} \cdot \left(\frac{\partial \mathbf{g}}{\partial \theta} \right)^{-1}, \quad (\text{B.19})$$

où $\Lambda_{\mathbf{x}_n}$ est la matrice de covariance associée à $\mathbf{x}_n$. θ étant un minimum de J , la matrice hessienne $\partial \mathbf{g} / \partial \theta$ est toujours de rang complet.

B.4.2 Propagation par échantillonnage sélectif

Dans l'exemple B.2 nous avons utilisé la technique de simulation de Monte Carlo pour estimer les paramètres de la distribution gaussienne qui est la plus consistante avec la distribution résultante non gaussienne. Les résultats obtenus ont été utilisés comme référence à des fins de comparaison. Malgré la précision obtenue par simulation de Monte Carlo, le nombre d'échantillons requis, souvent trop important (*e.g.*, 10000 particules), empêche son utilisation dans notre étude, étant donné les contraintes temps réel. Notre objectif est de propager l'incertitude gaussienne en minimisant les risques d'inconsistance lors de la cartographie globale (cf. section 7).

Récemment, Julier *et al.* ont présenté [114] une méthode de propagation des moyennes et des covariances en utilisant un échantillonnage sélectif. Cette technique est basée dans le calcul d'un nombre réduit $(2n + 1)$ d'échantillons $\mathcal{X}_i$ de la v.a. $X = \{\mathbf{x}, \Lambda_{\mathbf{x}}\}$, appelés points sigma. Au total, ce sont $2n + 1$ échantillons dont les poids $\mathcal{W}_i$ associés sont tels que :

$$\begin{aligned} \mathcal{X}_i &= \mathbf{x} & \text{et } \mathcal{W}_i &= \kappa/(n + \kappa) & \text{pour } i = 0, \\ \mathcal{X}_i &= \mathbf{x} + \left[\sqrt{(n + \kappa) \cdot \Lambda_{\mathbf{x}}} \right]_i & \text{et } \mathcal{W}_i &= \kappa/2(n + \kappa) & \text{pour } 1 \leq i \leq n, \\ \mathcal{X}_i &= \mathbf{x} - \left[\sqrt{(n + \kappa) \cdot \Lambda_{\mathbf{x}}} \right]_{i-n} & \text{et } \mathcal{W}_i &= \kappa/2(n + \kappa) & \text{pour } n + 1 \leq i \leq 2n, \end{aligned}$$

où $n = \dim(\mathbf{x})$, $[\mathbf{A}]_i$ est l'*i*-ième colonne de $\mathbf{A}$, et $\sqrt{\mathbf{A}}$ correspond à la décomposition de Cholesky d'une matrice $\mathbf{A}$ [179]. κ est un paramètre qui doit être fixé en fonction de la distribution. Pour une distribution gaussienne, il est recommandé $n + \kappa = 3$ [113], où encore $\kappa = \max(1, 3 - n)$ pour éviter des valeurs négatives sur κ .

A partir des échantillons de l'espace de X , l'algorithme calcule les échantillons correspondants dans l'espace transformé par

$$\mathcal{Y}_i = \mathbf{f}(\mathcal{X}_i). \quad (\text{B.20})$$

Ainsi, en utilisant les échantillons $\mathcal{Y}_i$ et les poids $\mathcal{W}_i$, les paramètres de la distribution $Y = \{\mathbf{y}, \Lambda_{\mathbf{y}}\}$ sont donnés par :

$$\mathbf{y} = \sum_{i=0}^{2n} \mathcal{W}_i \mathcal{Y}_i, \quad \Lambda_{\mathbf{y}} = \sum_{i=0}^{2n} \mathcal{W}_i (\mathcal{Y}_i - \mathbf{y}) \cdot (\mathcal{Y}_i - \mathbf{y})^T. \quad (\text{B.21})$$

Etant donné que la méthode utilise des échantillons représentatifs de la distribution de X , elle fournit d'excellents résultats dans l'approximation de la distribution de Y . Cela est montré dans l'exemple suivant.

Exemple B.3: Reprenons le cas de l'exemple B.2. La Figure B.3(a) présente les ellipses d'incertitude 3σ correspondant à la matrice $\Lambda_{\mathbf{c}}$ estimée par échantillonnage sélectif en tracé continu. L'estimée par simulation de Monte Carlo est présentée en tracé pointillé. Comme la technique de propagation d'incertitude par linéarisation, l'échantillonnage sélectif présente des résultats satisfaisants, et qui ne diffèrent pas beaucoup de ceux de Monte Carlo.

Pour le cas du sonar, les résultats obtenus confirment ceux présentés par Julier [113] (cf. Figure B.3(b)). En effet, les estimées de la moyenne et de la matrice de covariance sont très proches de celles obtenues par Monte Carlo, cela malgré la grande incertitude angulaire du capteur. En conséquence, la distribution gaussienne estimée par échantillonnage sélectif est consistante. $\diamond$

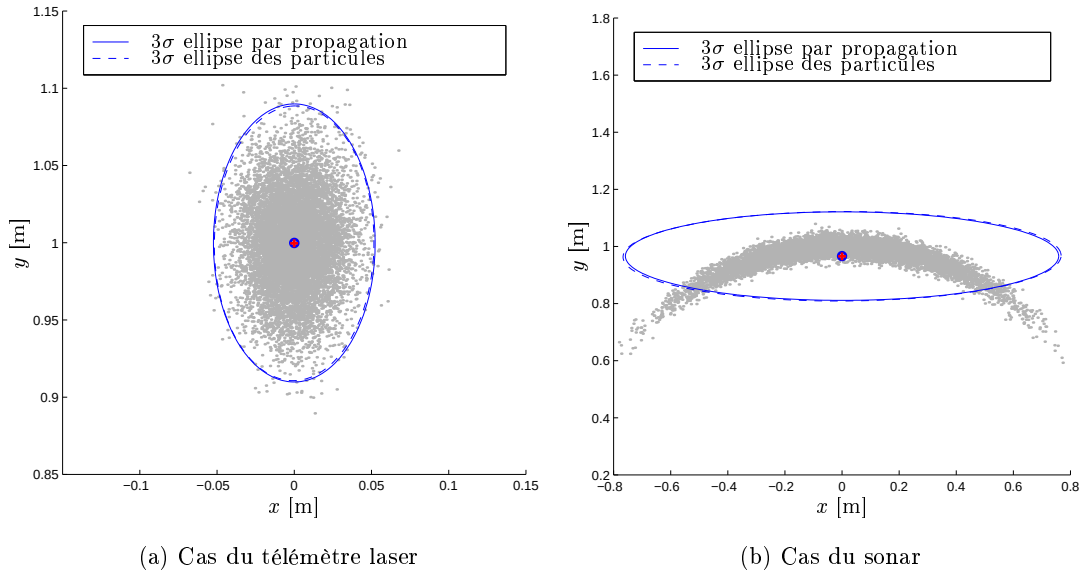


FIG. B.4 – Exemple d’application de la méthode de propagation d’incertitude par échantillonnage sélectif.

B.5 Distance de Mahalanobis

La distance de Mahalanobis est souvent utilisée comme mesure de correspondance entre deux variables aléatoires gaussiennes. Elle sert à évaluer la probabilité que deux mesures $X = \{\mathbf{x}, \Lambda_{\mathbf{x}}\}$ et $Y = \{\mathbf{y}, \Lambda_{\mathbf{y}}\}$ soient générées par la même source. La distance de Mahalanobis entre X et Y est ainsi définie :

$$D_{\chi^2}(\{\mathbf{x}, \Lambda_{\mathbf{x}}\}, \{\mathbf{y}, \Lambda_{\mathbf{y}}\}) = \mathbf{r}^T \cdot (\Lambda_{\mathbf{r}})^{-1} \cdot \mathbf{r}. \quad (\text{B.22})$$

Dans l’équation (B.22), $\mathbf{r} = \mathbf{x} - \mathbf{y}$ est l’écart entre les mesures X et Y , et $\Lambda_{\mathbf{r}} = \Lambda_{\mathbf{x}} + \Lambda_{\mathbf{y}}$ est la matrice de covariance de $\mathbf{r}$. $D_{\chi^2} \sim \chi_n^2$ ¹ sous l’hypothèse de la distribution gaussienne de X et Y [15]. L’hypothèse de correspondance entre X et Y est vraie si

$$D_{\chi^2}(X, Y) \leq Q_n(p) \quad (\text{B.23})$$

où $Q_n(p) = \chi_n^2(1 - p)$ est un seuil calculé à partir du pourcentage de l’intervalle de $\mathbf{r}$. $\chi_n^2(1 - p)$ est donné sous forme d’un tableau [15]. Par exemple, $Q_1(0, 95) = 3,84$ est le seuil correspondant à 95 % de D_{χ^2} lorsque X et Y sont de dimension un. Au cours de notre étude, nous avons réalisé des tests de correspondance pour des valeurs de D_{χ^2} à l’intérieur de l’intervalle à une probabilité de 95 %. Pour cela, nous avons utilisé les seuils suivants :

$$Q_1(0, 95) = 3,84 \quad Q_2(0, 95) = 5,99 \quad Q_3(0, 95) = 7,81. \quad (\text{B.24})$$

¹loi du χ^2 avec n degrés de liberté

Primitives géométriques 2-D

Cette annexe présente la modélisation et les propriétés des primitives de base utilisées dans le modèle de l'environnement.

C.1 Modélisation

C.1.1 Points

Un point est représenté par $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$, où $\mathbf{p} = (x, y)^T$ sont ses coordonnées dans un repère $\mathcal{X} \times \mathcal{Y}$ (Figure C.1(a)) et $\Lambda_{\mathbf{p}}$ est la matrice de covariance associée en supposant que $\mathbf{p}$ est perturbé par une v.a. ε telle que $\varepsilon \sim \mathcal{N}(\mathbf{0}, \Lambda_{\mathbf{p}})$.

C.1.2 Droites

Une droite est constitué par un ensemble infini de points de coordonnées (x, y) dans un repère $\mathcal{X} \times \mathcal{Y}$ tels qu'il existe $a \in \mathbb{R}$ et $b \in \mathbb{R}$ vérifiant la relation :

$$a \cdot x + b \cdot y = 1 \tag{C.1}$$

Dans notre étude, ces primitives sont représentées par $L = \{\mathbf{l}, \Lambda_{\mathbf{l}}\}$, où $\mathbf{l} = (\rho, \alpha)^T$ sont les coordonnées polaires du point de la droite qui se trouve le plus proche de l'origine de $\mathcal{X} \times \mathcal{Y}$ (Figure C.1(b)). Ces coordonnées sont limitées aux intervalles $\rho \geq 0$ et $0 \leq \alpha \leq 360^\circ$. $\Lambda_{\mathbf{l}}$ est la

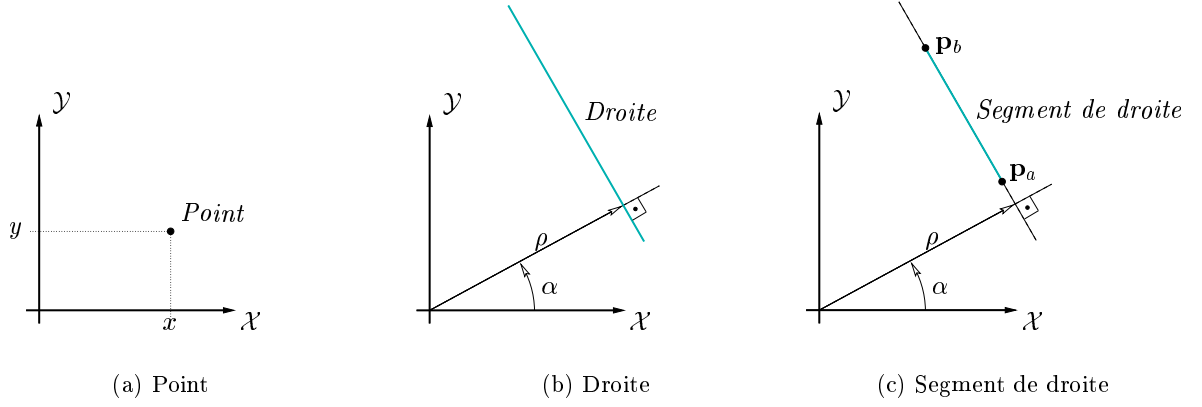


FIG. C.1 – Primitives géométriques dans l'espace 2-D.

matrice de covariance associée en supposant que $\mathbf{l}$ est perturbé par une v.a. ε telle que $\varepsilon \sim \mathcal{N}(\mathbf{0}, \Lambda_1)$. Tous les points (x, y) qui appartiennent à la droite L satisfont l'équation

$$\rho - x \cdot \cos(\alpha) - y \cdot \sin(\alpha) = 0. \quad (\text{C.2})$$

C.1.3 Segments de droite

Un segment de droite est formé par une droite de paramètres polaires $\mathbf{l} = (\rho, \alpha)^T$ délimitée par deux points d'extrémités dont les coordonnées sont $\mathbf{p}_a$ et $\mathbf{p}_b$ (Figure C.1(c)). Cette primitive est alors représentée par le quadruplet $S = \{\mathbf{l}, \Lambda_1, \mathbf{p}_a, \mathbf{p}_b\}$, où Λ_1 est la matrice de covariance associée à $\mathbf{l}$. Etant donné que $\mathbf{p}_a = (x_a, y_a)^T$ et $\mathbf{p}_b = (x_b, y_b)^T$ se trouvent sur la droite de paramètres $\mathbf{l}$, nous avons :

$$\rho - x_a \cdot \cos(\alpha) - y_a \cdot \sin(\alpha) = 0, \quad (\text{C.3})$$

$$\rho - x_b \cdot \cos(\alpha) - y_b \cdot \sin(\alpha) = 0, \quad (\text{C.4})$$

$$(y_a - y_b) \cdot \tan(\alpha) = -(x_a - x_b) \quad (\text{C.5})$$

Aucune mesure d'incertitude n'est associée aux points $\mathbf{p}_a$ et $\mathbf{p}_b$. En effet, leur incertitude dépend de l'incertitude des paramètres de la droite à laquelle ils appartiennent. Ainsi, si cela est nécessaire, des matrices de covariance associées à $\mathbf{p}_a$ et à $\mathbf{p}_b$ peuvent être estimées par propagation de covariances (cf. section B.4).

C.2 Changement de repère

Dans la section A.3, nous avons introduit la notion de positionnement du robot dans le repère global $\mathcal{R}^G$, qui est représenté par $Z = \{\mathbf{z}, \Lambda_z\}$. Nous avons aussi présenté $Z_L^R = \{\mathbf{z}_L^R, \mathbf{0}_3\}$ comme étant le positionnement du repère du système multicapteur $\mathcal{R}^S$ par rapport au repère robot $\mathcal{R}^R$.

Dans cette section, nous présentons des fonctions qui permettent de déterminer les nouveaux paramètres des indices géométriques de type point et droite lors d'un changement de repère.

La transformation du repère $\mathcal{R}^G$ au repère $\mathcal{R}^R$ est notée :

$$X^R = \mathcal{T}(X^G, Z) \quad (\text{C.6})$$

où $\mathcal{T}$ est un opérateur qui relie $X^G = \{\mathbf{x}^G, \mathbf{\Lambda}_{\mathbf{x}^G}\}$, la représentation d'un indice géométrique dans le repère $\mathcal{R}^G$, et $X^R = \{\mathbf{x}^R, \mathbf{\Lambda}_{\mathbf{x}^R}\}$, la représentation correspondante dans le repère $\mathcal{R}^R$. Z est exprimé dans le repère global $\mathcal{R}^G$. $\mathcal{T}$ implique deux opérations indépendantes : $\mathcal{T}_d$ associée aux composantes déterministes, et $\mathcal{T}_s$ associée à la propagation d'incertitude stochastique.

La transformation inverse est notée :

$$X^G = \mathcal{T}^{-1}(X^R, Z), \quad (\text{C.7})$$

qui, de la même façon que $\mathcal{T}$, est divisée en partie déterministe $\mathcal{T}_d^{-1}$ et en partie stochastique $\mathcal{T}_s^{-1}$.

Ces transformations sont formulées dans les sections qui suivent, pour chaque type d'indice géométrique.

C.2.1 Indices de type point

Soient $P^G = \{\mathbf{p}^G, \mathbf{\Lambda}_{\mathbf{p}^G}\}$ et $P^R = \{\mathbf{p}^R, \mathbf{\Lambda}_{\mathbf{p}^R}\}$ les représentations d'un indice de type point dans les repères $\mathcal{R}^G$ et $\mathcal{R}^R$, respectivement. Pour ces indices, la transformation $\mathcal{T}_d$ est directement donnée par l'application de l'équation (A.1) :

$$\mathbf{p}^R = \mathcal{T}_d(\mathbf{p}^G, \mathbf{z}) = \mathbf{R}(\theta) \cdot (\mathbf{p}^G - \mathbf{t}), \quad (\text{C.8})$$

et son inverse existe et est donnée par :

$$\mathbf{p}^G = \mathcal{T}_d^{-1}(\mathbf{p}^R, \mathbf{z}) = \mathbf{R}^T(\theta) \cdot \mathbf{p}^R + \mathbf{t}. \quad (\text{C.9})$$

C.2.2 Indices de type droite

Soient $L^G = \{\mathbf{l}^G, \mathbf{\Lambda}_{\mathbf{l}^G}\}$ et $L^R = \{\mathbf{l}^R, \mathbf{\Lambda}_{\mathbf{l}^R}\}$ les représentations d'un indice de type droite dans les repères $\mathcal{R}^G$ et $\mathcal{R}^R$, respectivement. Pour établir la relation $\mathbf{l}^R = \mathcal{T}_d(\mathbf{l}^G, \mathbf{z})$, nous considérons que la transformation de l'équation (A.1) s'applique également à tous les points qui appartiennent à la droite paramétrée par $\mathbf{l}^G = (\rho^G, \alpha^G)^T$. Ainsi, pour une paire de points de coordonnées (x_a^G, y_a^G) et (x_b^G, y_b^G) dans $\mathcal{R}^G$, nous pouvons montrer que les composantes ρ^G et α^G du vecteur de paramètres $\mathbf{l}^G$ de la droite qui relie ces points sont

$$\alpha^G = \arctan(-(x_a^G - x_b^G)/(y_a^G - y_b^G)), \quad (\text{C.10})$$

$$\rho^G = \frac{1}{2} \cdot (x_a^G + x_b^G) \cdot \cos(\alpha^G) + \frac{1}{2} \cdot (y_a^G + y_b^G) \cdot \sin(\alpha^G). \quad (\text{C.11})$$

De même, ces points ont des coordonnées (x_a^R, y_a^R) et (x_b^R, y_b^R) dans le repère $\mathcal{R}^R$ données par

$$\begin{pmatrix} x_a^R \\ y_a^R \end{pmatrix} = \mathbf{R}(\theta) \cdot \left\{ \begin{pmatrix} x_a^G \\ y_a^G \end{pmatrix} - \mathbf{t} \right\}, \quad \text{et} \quad \begin{pmatrix} x_b^R \\ y_b^R \end{pmatrix} = \mathbf{R}(\theta) \cdot \left\{ \begin{pmatrix} x_b^G \\ y_b^G \end{pmatrix} - \mathbf{t} \right\}. \quad (\text{C.12})$$

Les paramètres $\mathbf{l}^R = (\rho^R, \alpha^R)^T$ de la droite passant par ces points sont alors exprimés dans le repère local par :

$$\alpha^R = \arctan(-(x_a^R - x_b^R)/(y_a^R - y_b^R)), \quad (\text{C.13})$$

$$\rho^R = \frac{1}{2} \cdot (x_a^R + x_b^R) \cdot \cos(\alpha^R) + \frac{1}{2} \cdot (y_a^R + y_b^R) \cdot \sin(\alpha^R). \quad (\text{C.14})$$

A partir du développement mathématique des équations (C.10)-(C.14), nous pouvons montrer que

$$\rho^R = \rho^G - x \cdot \cos(\alpha^G) - y \cdot \sin(\alpha^G) = \rho^G - \beta^T(\alpha^G) \cdot \mathbf{t}, \quad (\text{C.15})$$

$$\alpha^R = \alpha^G - \theta, \quad (\text{C.16})$$

où $\beta(\phi) = (\cos(\phi) \quad \sin(\phi))^T$ pour tout angle ϕ . Ainsi, la transformation directe de ces indices est donnée par

$$\mathbf{l}^R = \mathcal{T}_d(\mathbf{l}^G, \mathbf{z}) = \mathbf{l}^G - \begin{pmatrix} \beta^T(\alpha^G) \cdot \mathbf{t} \\ \theta \end{pmatrix}. \quad (\text{C.17})$$

Son inverse existe et est donnée par

$$\mathbf{l}^G = \mathcal{T}_d^{-1}(\mathbf{l}^R, \mathbf{z}) = \mathbf{l}^R + \begin{pmatrix} \beta^T(\alpha^R + \theta) \cdot \mathbf{t} \\ \theta \end{pmatrix}. \quad (\text{C.18})$$

C.3 Relations

C.3.1 Angle de vue d'un point

Soit un point $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$ où $\mathbf{p} = (x, y)^T$. Cet indice géométrique est observé à l'origine du système de coordonnées dans la direction angulaire $\phi = \arctan(y/x)$ par rapport à l'axe $\mathcal{X}$. La variance Λ_{ϕ} sur ϕ est obtenue par propagation de la matrice d'incertitude $\Lambda_{\mathbf{p}}$ des coordonnées $\mathbf{p}$ (cf. section B.4).

C.3.2 Intersection de deux droites

Soit $L_a = \{\mathbf{l}_a, \Lambda_{\mathbf{l}_a}\}$ et $L_b = \{\mathbf{l}_b, \Lambda_{\mathbf{l}_b}\}$ deux droites, où $\mathbf{l}_a = (\rho_a, \alpha_a)^T$ et $\mathbf{l}_b = (\rho_b, \alpha_b)^T$. Leur point d'intersection $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$, où $\mathbf{p} = (x, y)^T$ satisfait simultanément

$$\rho_a - x \cdot \cos(\alpha_a) - y \cdot \sin(\alpha_a) = 0, \quad (\text{C.19})$$

$$\rho_b - x \cdot \cos(\alpha_b) - y \cdot \sin(\alpha_b) = 0. \quad (\text{C.20})$$

D'où

$$\mathbf{p} = \begin{pmatrix} \rho_b \sin(\alpha_a) - \rho_a \sin(\alpha_b) \\ \rho_a \cos(\alpha_b) - \rho_b \cos(\alpha_a) \end{pmatrix} \cdot \frac{1}{\sin(\alpha_1 - \alpha_2)}. \quad (\text{C.21})$$

Etant donné la forme analytique de (C.21), nous pouvons estimer $\Lambda_{\mathbf{p}}$ à partir des matrices de covariance $\Lambda_{\mathbf{l}_a}$ et $\Lambda_{\mathbf{l}_b}$ en utilisant la propagation d'incertitude (cf. section B.4).

Représentation géométrique d'environnements

D.1 Structures cartographiques

Pour la modélisation d'environnements, il est nécessaire de choisir des structures capables de représenter les caractéristiques qui ont le plus d'intérêt pour une application donnée. Nous avons choisi deux types de structures : les demi-plans et les arêtes. Ces structures diffèrent par les primitives géométriques qui définissent leur morphologie.

D.1.1 Structures de type demi-plan

Les demi-plans sont des structures correspondant à des surfaces planes de l'environnement. Ces surfaces sont traduites par des segments de droite dans les images télémétriques. De plus, une seule face a été observée par le capteur. Nous représentons ces structures par

$$D = \{S, \mathbf{v}\} \tag{D.1}$$

où $S = \{\mathbf{l}, \mathbf{A}_l, \mathbf{p}_a, \mathbf{p}_b\}$ décrit le segment de droite de la structure et $\mathbf{v} = (a, b, c)^T$ est un vecteur de paramètres permettant de représenter la zone de visibilité du demi-plan. La Figure D.1(a) illustre ce genre de structure, où le côté du demi-plan représenté par une région grise correspond à sa zone de non-visibilité.

Les paramètres de la zone de visibilité sont reliés aux paramètres $\mathbf{l} = (\rho, \alpha)^T$ du segment de

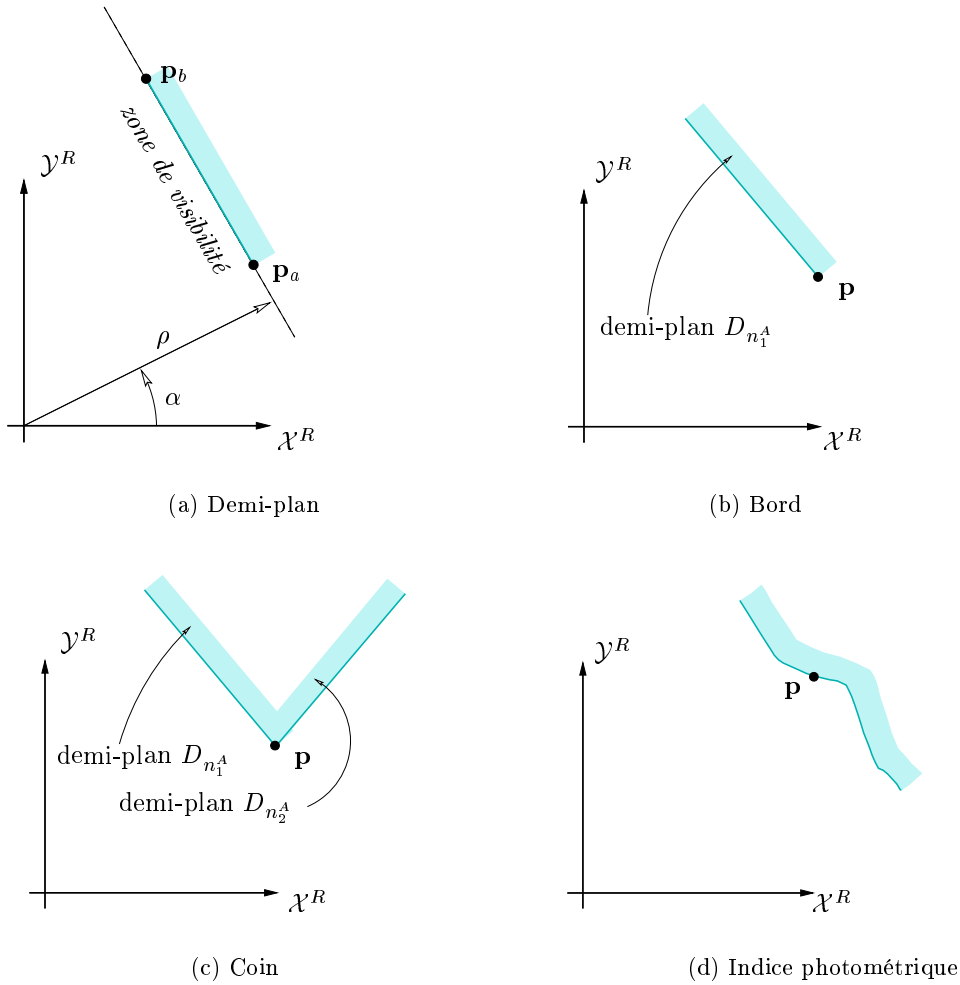


FIG. D.1 – Structures cartographiques.

droite associée. $\mathbf{v}$ permet de déterminer si un point de coordonnées (x, y) se trouve dans la zone de visibilité du demi-plan associé D . Etant donné la définition d'une primitive de type droite (section C.1.2), tous les points (x^*, y^*) se trouvant sur la droite de paramètres $\mathbf{l}$ satisfont l'équation (C.2).

Pour un point quelconque $\mathbf{p} = (x, y)^T$, une valeur positive de $\rho - x \cdot \cos(\alpha) - y \cdot \sin(\alpha)$ signifie que $\mathbf{p}$ se trouve du même côté de la droite $\mathbf{l}$ que l'origine du repère. Cependant, la région de visibilité d'un demi-plan peut se trouver soit du même côté que l'origine du repère par rapport à la droite, soit de l'autre côté. Ainsi, les paramètres de la zone de visibilité sont déterminés par

$$(a, b, c)^T = \begin{cases} (\rho, -\cos(\alpha), -\sin(\alpha))^T & \text{si } \rho - x_v \cdot \cos(\alpha) - y_v \cdot \sin(\alpha) > 0, \\ (-\rho, \cos(\alpha), \sin(\alpha))^T & \text{dans le cas contraire.} \end{cases} \quad (\text{D.2})$$

où $\mathbf{p}_v = (x_v, y_v)^T$ sont les coordonnées d'un point de contrôle qui se trouve dans la zone de visibilité du demi-plan. Ainsi, tout point $\mathbf{p}$ se trouvant du même côté de la droite $\mathbf{l}$ que le point $\mathbf{p}_v$ sera

identifié comme visible en utilisant l'opérateur suivant :

$$\mathcal{V}(\mathbf{p}, \mathbf{v}) = \begin{cases} \text{VRAI} & \text{si } a + b \cdot x + c \cdot y > 0, \\ \text{FAUX} & \text{dans le cas contraire.} \end{cases} \quad (\text{D.3})$$

$\mathcal{V}(\mathbf{p}, \mathbf{v}) = \text{VRAI}$ implique que le point $\mathbf{p}$ se trouve dans la zone de visibilité du demi-plan $D = \{S, \mathbf{v}\}$, définie par (D.2).

Nous avons choisi une représentation paramétrique de la zone de visibilité pour pouvoir lui appliquer directement les transformations de repère, tout en conservant l'information du côté d'observation.

D.1.2 Structures de type arête

Les arêtes sont des structures verticales de l'environnement qui interceptent le plan 2-D de balayage laser sous la forme d'un point. Ces structures sont classées en trois sous-types :

- Le sous-type *bord*, qui est attaché à l'extrémité d'un demi-plan de la carte (Figure D.1(b)). Cette structure est sensée représenter la limite physiques d'une surface plane balayée par le télémètre laser ;
- Le sous-type *coin*, qui est formé par l'intersection entre deux demi-plans de la carte (Figure D.1(c)) ;
- Le sous-type *indice photométrique*, issu d'une droite verticale dans l'image vidéo, et qui ne correspond pas toujours à l'observation d'un bord, ou d'un coin (Figure D.1(d)).

Ces indices sont représentés par le quadruplet

$$A = \{P, \kappa^A, n_1^A, n_2^A\}, \quad (\text{D.4})$$

où $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$ sont les paramètres stochastiques du point indiquant la position de l'arête et son incertitude, κ^A est une étiquette pour identifier les sous-types bord (lettre B), coin (lettre C) ou indice photométrique (lettre P). n_1^A et n_2^A sont les index des demi-plans associés à A (le cas échéant).

D.2 Changement inverse de repère

Nous présentons dans la section C.2 une notation pour le changement de repère pour des primitives de type point $P = \{\mathbf{p}, \Lambda_{\mathbf{p}}\}$, de type droite $L = \{\mathbf{l}, \Lambda_{\mathbf{l}}\}$. Pour les structures cartographiques, la transformation inverse de repère elle est largement utilisée pour la mise en correspondance de cartes, ainsi que pour la cartographie globale. Cette transformation requiert comme paramètre la position globale $Z = \{\mathbf{z}, \Lambda_{\mathbf{z}}\}$ du robot, et prend en compte la propagation de covariances. Nous étendons cette notation aux structures cartographiques de la façon suivante :

$$D_i^G = \mathcal{T}^{-1}(D_i^R, Z), \quad (\text{D.5})$$

$$A_i^G = \mathcal{T}^{-1}(A_i^R, Z). \quad (\text{D.6})$$

L'opérateur $\mathcal{T}^{-1}$ s'applique séparément à chacune des primitives géométriques de la structure cartographique :

$$\begin{aligned} D_i^G = \{S_i^G, \mathbf{v}_i^G\} &\implies S_i^G = \mathcal{T}^{-1}(S_i^R, Z) \text{ et } \mathbf{v}_i^G = \mathcal{T}^{-1}(\mathbf{v}_i^R, Z) \\ A_i^G = \{P_i^G, \kappa_i^A, n_{1,i}^A, n_{2,i}^A\} &\implies P_i^G = \mathcal{T}^{-1}(P_i^R, Z) \end{aligned} \quad (\text{D.7})$$

Il faut cependant définir le calcul de $\mathbf{v}_i^G$, représentation paramétrique de la zone de visibilité du demi-plan dans le repère global. La transformation $\mathcal{T}^{-1}(\mathbf{v}_i^R, Z)$ doit conserver les propriétés de visibilité de n'importe quel point statique de coordonnées $\mathbf{p}^R$ dans $\mathcal{R}^R$, ou $\mathbf{p}^G$ dans $\mathcal{R}^G$. Donc, cette transformation doit satisfaire à :

$$\mathcal{V}(\mathbf{p}^G, \mathbf{v}_i^G) = \mathcal{V}(\mathbf{p}^R, \mathbf{v}_i^R), \quad (\text{D.8})$$

où $\mathcal{V}$ est l'opérateur de visibilité défini par (D.3). La résolution de l'équation (D.8) nous permet de définir ainsi $\mathcal{T}^{-1}$ pour les paramètres de visibilité :

$$\mathbf{v}_i^G = \mathcal{T}^{-1}(\mathbf{v}_i^R, Z) \implies \begin{cases} \begin{pmatrix} a_i^G \\ b_i^G \end{pmatrix} = \mathbf{R}^T(\theta) \cdot \begin{pmatrix} a_i^R \\ b_i^R \end{pmatrix}, \\ c_i^G = c_i^R - \mathbf{t} \cdot \mathbf{R}^T(\theta) \cdot \begin{pmatrix} a_i^R \\ b_i^R \end{pmatrix}. \end{cases} \quad (\text{D.9})$$

D.3 Correspondance de structures cartographiques

La mise en correspondance de structures cartographiques consiste à rechercher celles pouvant être générées par le même objet de l'environnement. Les correspondances ne peuvent être vérifiées qu'entre structures du même type, *i.e.*, entre demi-plans ou entre arêtes, représentées dans le même repère et soumises aux conditions présentées dans cette section.

D.3.1 Correspondance entre demi-plans

Soit deux demi-plans $D_i = \{S_i, \mathbf{v}_i\}$ et $D_j = \{S_j, \mathbf{v}_j\}$, tels que $S_i = \{L_i, \mathbf{p}_{a,i}, \mathbf{p}_{b,i}\}$ et $S_j = \{L_j, \mathbf{p}_{a,j}, \mathbf{p}_{b,j}\}$. Pour que la correspondance entre D_i et D_j soit confirmée, il faut que les critères suivants soient satisfaites :

- I. Critère statistique : la distance de Mahalanobis entre L_i et L_j doit satisfaire à (cf. section B.5) :

$$D_{\chi^2}(L_i, L_j) \leq Q_2(0, 95). \quad (\text{D.10})$$

- II. Critère géométrique de visibilité : le point de contrôle $\mathbf{p}_c$, qui se trouve à une distance orthogonale de $l = 5m^1$ par rapport au centre de gravité du segment S_i (*i.e.*, $\mathbf{p}_c = (\mathbf{p}_{a,i} +$

¹En fait, intuitivement avec $l > 5m$ nous pouvons être sûrs de que le point de contrôle soit fiable si le critère statistique se vérifie. Ainsi, les deux segments sont très proches, et le point de contrôle ne risque pas de se trouver dans la région entre les segments.

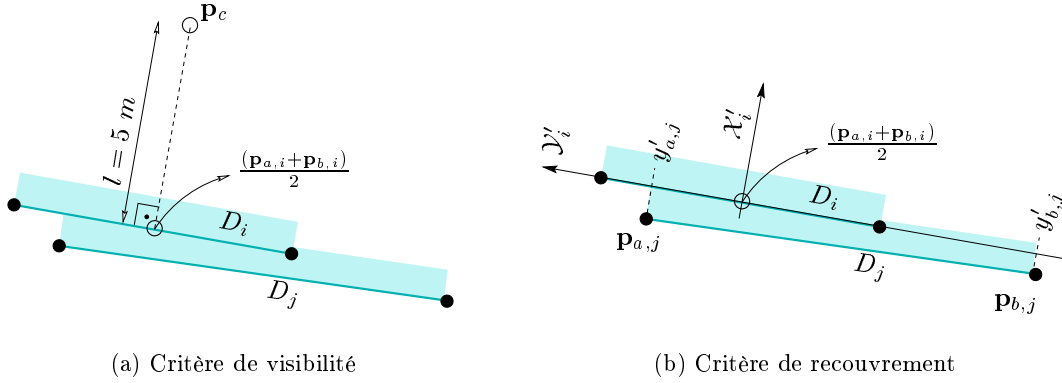


FIG. D.2 – Critères de correspondance géométrique entre demi-plans.

$\mathbf{p}_{b,i})/2$), doit avoir la même visibilité par rapport aux deux demi-plans :

$$\mathcal{V}(\mathbf{p}_c, \mathbf{v}_i) = \mathcal{V}(\mathbf{p}_c, \mathbf{v}_j), \quad (\text{D.11})$$

Ce test est illustré sur la Figure D.2(a) et il vise à déterminer si les deux demi-plans sont observables du même côté. Ce test est très efficace, surtout pour éviter que deux faces distinctes d'un mur soient mises en correspondance ;

- III. Critère géométrique de recouvrement : la projection du segment S_j sur la droite support du segment S_i doit indiquer un recouvrement total ou partiel entre ces structures. Pour déterminer les conditions de recouvrement, nous avons utilisé un repère auxiliaire $\mathcal{X}'_i \times \mathcal{Y}'_i$ dont l'origine est placée au centre de gravité du segment S_i , et dont l'axe $\mathcal{Y}'_i$ est porté par la droite L_i (voir Figure D.2(b)). Le test porte sur les coordonnées $y'_{a,j}$ et $y'_{b,j}$ des extrémités de S_j . l_i étant la longueur de S_i , le critère de recouvrement est satisfait si au moins une des trois conditions suivantes est vérifiée :

$$\begin{aligned} \text{Recouvrement total} & : \max(y'_{a,j}, y'_{b,j}) \leq \frac{l_i}{2} \quad \text{et} \quad \min(y'_{a,j}, y'_{b,j}) \geq -\frac{l_i}{2}, \\ \text{Recouvrement partiel} & : \min(y'_{a,j}, y'_{b,j}) \leq -\frac{l_i}{2} \quad \text{et} \quad \max(y'_{a,j}, y'_{b,j}) \geq -\frac{l_i}{2} + \delta, \\ \text{Recouvrement partiel} & : \max(y'_{a,j}, y'_{b,j}) \leq \frac{l_i}{2} \quad \text{et} \quad \min(y'_{a,j}, y'_{b,j}) \leq \frac{l_i}{2} - \delta. \end{aligned}$$

Ainsi, nous avons un recouvrement total lorsque les deux extrémités de S_j se projettent entre les extrémités de S_i . Nous considérons un recouvrement s'il existe au moins $\delta = 0,2 \text{ m}$ de longueur commune aux demi-plans D_i et D_j . Comme pour le paramètre l du critère de visibilité, δ n'a pas de valeur critique.

D.3.2 Correspondance entre arêtes

Soit deux arêtes $A_i = \{P_i, \kappa_i^A, n_{1,i}^A, n_{2,i}^A\}$ et $A_j = \{P_j, \kappa_j^A, n_{1,j}^A, n_{2,j}^A\}$. Pour que la correspondance entre A_i et A_j soit confirmée, il suffit que le critère statistique suivant soit satisfait : la distance de Mahalanobis entre P_i et P_j doit satisfaire (cf. section B.5) :

$$D_{\chi^2}(P_i, P_j) \leq Q_2(0, 95). \quad (\text{D.12})$$

D.3.3 Notation pour les correspondances

Pour faciliter la présentation des structures en correspondance, nous utilisons la notation suivante :

Définition D.1 (Correspondance de structures cartographiques): *La correspondance entre deux structures cartographiques X_1 et X_2 appartenant au même repère est représentée par*

$$X_1 \longleftrightarrow X_2. \quad (\text{D.13})$$

Supposons maintenant que nous avons deux structures X^G et X^R appartenant respectivement au repère global $\mathcal{R}^G$ et au repère local $\mathcal{R}^R$. Ce dernier est attaché au robot qui se trouve à la position $Z = \{\mathbf{z}, \mathbf{\Lambda}_z\}$. Si la représentation globale de X^R est en correspondance avec X^G , i.e.,

$$X^G \longleftrightarrow \mathcal{T}^{-1}(X^R, Z), \quad (\text{D.14})$$

alors cela est représenté par une notation équivalente :

$$X^G \xleftrightarrow{Z} X^R, \quad (\text{D.15})$$

où la structure de gauche doit être celle représentée dans le repère global. $\diamond$

D.4 Cartes d'environnement

Les structures présentées dans la section précédente sont assez fréquentes dans des environnements d'intérieur. Elles peuvent par exemple correspondre à des murs, à des portes, ou à des objets polygonaux en général. L'inclusion des indices photométriques nous permet aussi d'utiliser ce type d'information de façon non usuelle pour la localisation. En effet, dans la plupart des travaux présentés dans la littérature, on ne s'intéresse qu'à leur angle de vue, comme dans [8].

D.4.1 Symbolique

Les cartes d'environnement sont effect composées de structures cartographiques attachées à un repère donné $\mathcal{R} : \mathcal{X} \times \mathcal{Y}$. De façon générale, une carte $\mathcal{M}$ est formée par un ensemble de N_D structures de type demi-plan et de N_A arêtes, sous la forme

$$\mathcal{M} = \{D_1, \dots, D_{N_D}, A_1, \dots, A_{N_A}\}. \quad (\text{D.16})$$

La Figure D.3 nous montre une carte simulée. Cette figure introduit la symbolique utilisée pour la représentation visuelle des structures cartographiques (cf. légende). Les lignes d'observation des arêtes photométriques sont illustrées par des tracés non continus, qui partent du centre du repère caméra.

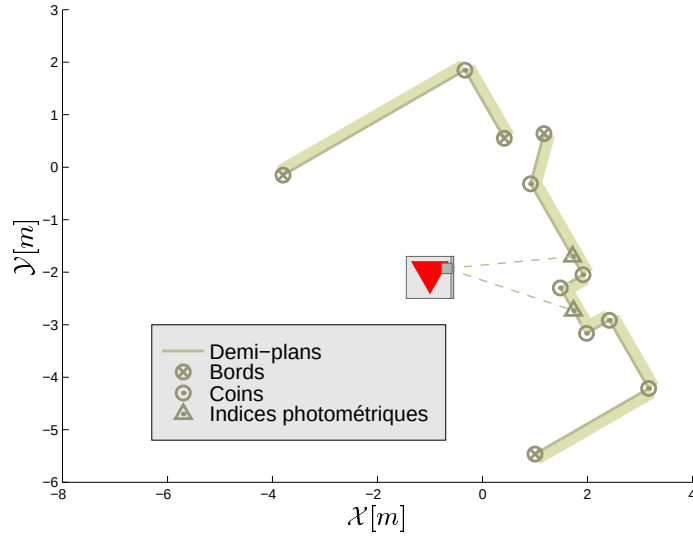


FIG. D.3 – Simulation d'une carte.

D.4.2 Carte locale et carte globale

Etant donné que dans notre étude nous avons à faire à deux repères d'intérêt, le repère global $\mathcal{R}^G$ et le repère local $\mathcal{R}^R$, il est nécessaire de distinguer les cartes d'environnement associées au niveau des notations. Ces cartes sont respectivement la carte globale $\mathcal{M}^G$ et la carte locale $\mathcal{M}^R$.

Lorsque nous faisons référence aux structures qui composent une carte, nous associons toujours un index. Ainsi, en considérant la carte locale $\mathcal{M}^R$, l'*i*-ième demi-plan local D_i^R est représenté par

$$D_i^R = \{S_i^R, \mathbf{v}_i^R\}, \quad (\text{D.17})$$

où S_i^R est le segment associé et $\mathbf{v}_i^R$ est le vecteur de visibilité. Dans $S_i^R = \{L_i^R, \mathbf{p}_{a,i}, \mathbf{p}_{b,i}\}$, $L_i^R = \{\mathbf{l}_i^R, \Lambda_{\mathbf{l}_i^R}\}$ est la droite support et $\mathbf{p}_{a,i}$ et $\mathbf{p}_{b,i}$ sont les points d'extrémité. L'*i*-ième arête locale A_i^R est représentée par

$$A_i^R = \{P_i^R, \kappa_i^R, n_{1,i}^R, n_{2,i}^R\}, \quad (\text{D.18})$$

où $P_i^R = \{\mathbf{p}_i^R, \Lambda_{\mathbf{p}_i^R}\}$ est la primitive de type point représentant son centre, κ_i^R est l'étiquette utilisée pour identifier les sous-type : $\kappa_i^R = \mathbf{B}$ pour les bords, $\kappa_i^R = \mathbf{C}$ pour les coins et $\kappa_i^R = \mathbf{P}$ pour les indices photométriques. $n_{1,i}^R$ et $n_{2,i}^R$ sont les index des demi-plans associés à A_i^R , pour les bords ou les coins.

On retrouve des notations similaires, avec la lettre G à la place de la lettre R pour les structures globales D_i^G et A_i^G et leurs primitives. Ainsi, la carte locale est représentée par

$$\mathcal{M}^R = \{D_1^R, \dots, D_{N_D^R}^R, A_1^R, \dots, A_{N_A^R}^R\}. \quad (\text{D.19})$$

De même, la carte globale est représentée par

$$\mathcal{M}^G = \{D_1^G, \dots, D_{N_D^G}^G, A_1^G, \dots, A_{N_A^G}^G\}. \quad (\text{D.20})$$

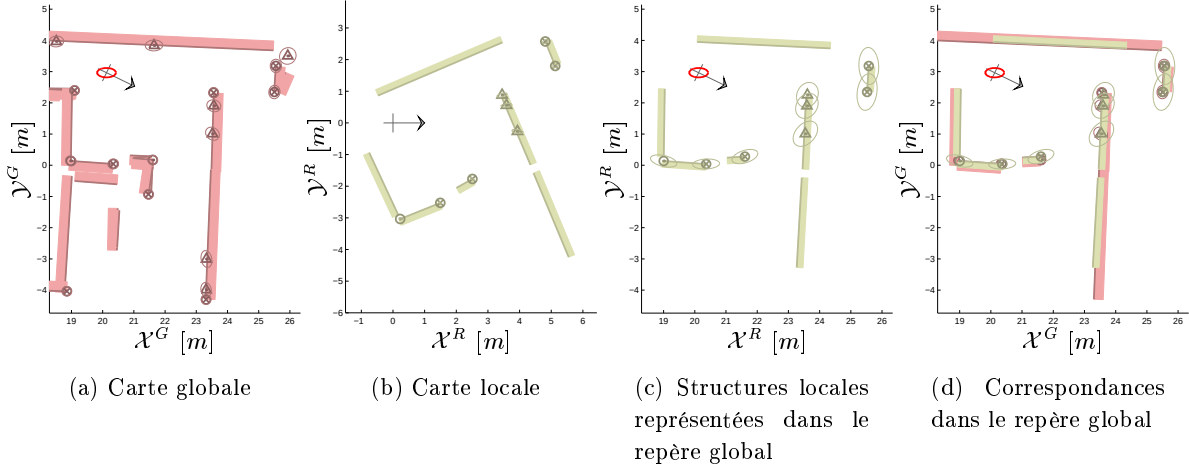


FIG. D.4 – Exemple réel de la mise en correspondance de cartes d'environnement.

D.5 Mise en correspondance de cartes

La mise en correspondance de cartes d'environnement consiste à déterminer les structures de deux cartes $\mathcal{M}^G$ et $\mathcal{M}^R$ qui sont en correspondance. Pour résoudre ce problème, nous pouvons utiliser une approche globale, ou une approche locale. Les approches globales s'appuient sur une recherche exhaustive pour trouver des correspondances, comme par exemple, dans [164], en utilisant arbre d'interprétation. Ces techniques sont connues pour être coûteuses en temps de calcul ; par exemple, dans [164] l'estimation de position d'un robot mobile se fait en 20 s en moyenne. Leur application est dirigée vers l'autolocalisation, où il est nécessaire d'estimer la position absolue du robot sans qu'aucune information *a priori* sur cette position soit disponible.

Dans notre étude, nous avons utilisé une approche locale de la mise en correspondance. En fait, les approches locales utilisent comme information *a priori* une estimée $\hat{Z} = \{\hat{\mathbf{z}}, \mathbf{\Lambda}_{\hat{\mathbf{z}}}\}$ de la position du robot. Ainsi, la complexité du problème est énormément réduite. Elle consiste à tester la correspondance de chaque structure X_i^G de $\mathcal{M}^G$ avec la représentation globale $X_j^{G|R}$ de chaque structure locale X_j^R (*i.e.*, vérifier si $X_i^G \longleftrightarrow \mathcal{T}^{-1}(X_j^R, \hat{Z})$). Bien entendu, les tests de correspondance ne sont faits qu'entre structures de même type.

Filtres stochastiques

E.1 Introduction

Les outils présentés ici sont destinés à l'estimation d'état de systèmes stochastique discrets. De façon générale, ces systèmes sont supposés évoluer dans le temps discret k en suivant

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k, \quad (\text{E.1})$$

où $\mathbf{x}_k \in \mathbb{R}^n$ représente le vecteur d'état, $\mathbf{u}_k \in \mathbb{R}^p$ représente les entrées d'excitation du système et $\mathbf{v}_k$ est une perturbation aléatoire. L'équation (E.1) est dite modèle du processus et $\mathbf{v}_k$ correspond au bruit du processus. Les mesures effectuées sur ce système sont reliées à $\mathbf{x}_k$ par le modèle de mesure

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{w}_k, \quad (\text{E.2})$$

où $\mathbf{y}_k \in \mathbb{R}^m$ est le vecteur de mesures et $\mathbf{w}_k$ est le bruit de mesure. $\mathbf{h}_k(\cdot)$ est dite fonction de mesure.

Le système (E.1)-(E.2) évolue à partir d'un état initial $\mathbf{x}_0$ de covariance $\mathbf{\Lambda}_{\mathbf{x}_0}$. Les hypothèses suivantes sont faites sur le modèle stochastique :

Hypothèse E.1: *L'état initial $\mathbf{x}_0$ est supposé indépendant de $\mathbf{v}_k$ et $\mathbf{w}_k$;* ◇

Hypothèse E.2: *Les fonctions $\mathbf{f}_k(\cdot)$ et $\mathbf{h}_k(\cdot)$ sont connues ;* ◇

Hypothèse E.3: *$\mathbf{v}_k$ et $\mathbf{w}_k$ sont des bruits non corrélés suivant une distribution gaussienne de moyenne nulle et de covariances $\mathbf{\Lambda}_{\mathbf{v}_k}$ et $\mathbf{\Lambda}_{\mathbf{w}_k}$, respectivement.* ◇

Le problème de l'estimation stochastique consiste alors à déterminer des estimées $\hat{\mathbf{x}}_k$ et $\Lambda_{\hat{\mathbf{x}}_k}$ respectives de $\mathbf{x}_k$ et $\Lambda_{\mathbf{x}_k}$, pour $k = 1, 2, \dots$. Dans notre étude nous avons évalué deux approches pour résoudre ce problème : l'approche *filtre de Kalman* et l'approche *intersection de covariances*. Leur formulation mathématique est présentée dans les sections suivantes.

Nous nous intéressons aux conditions d'utilisation et aux limitations de ces filtres, en mettant l'accent sur la consistance de l'estimée $\hat{\mathbf{x}}_k$. Ce concept est défini dans la section E.2. Ensuite, dans la section E.3, nous présentons des approches pour le filtrage de systèmes linéaires. Enfin, la section E.4 présente l'extension de ces méthodes de filtrage aux systèmes non-linéaires.

E.2 Consistance des estimées

Les définitions suivantes s'appliquent aux estimées $\hat{\mathbf{x}}_k$ et $\Lambda_{\hat{\mathbf{x}}_k}$ respectives de $\mathbf{x}_k$ et $\Lambda_{\mathbf{x}_k}$:

Définition E.1 (Consistance): $\hat{\mathbf{x}}_k$ est dite estimée consistante de $\mathbf{x}_k$ si $\mathcal{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)\} = \mathbf{0}$, et si les erreurs $\mathbf{x}_k - \hat{\mathbf{x}}_k$ sont compatibles avec $\Lambda_{\hat{\mathbf{x}}_k}$. $\diamond$

Définition E.2 (Pessimisme): L'estimée $\hat{\mathbf{x}}_k$ est dite pessimiste si $\mathcal{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)\} = \mathbf{0}$ et $\Lambda_{\hat{\mathbf{x}}_k} > \Lambda_{\mathbf{x}_k}$ ¹. $\diamond$

Définition E.3 (Optimisme): L'estimée $\hat{\mathbf{x}}_k$ est dite optimiste si $\mathcal{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)\} = \mathbf{0}$ et $\Lambda_{\hat{\mathbf{x}}_k} < \Lambda_{\mathbf{x}_k}$ ². $\diamond$

Définition E.4 (Inconsistance): L'estimée $\hat{\mathbf{x}}_k$ est dite inconsistante si $\mathcal{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)\} \neq \mathbf{0}$. $\diamond$

Les définitions ci-dessus expriment de manière simple la signification de termes utilisés couramment en estimation et filtrage stochastique [15, 157, 104]. Ainsi, la consistance implique l'absence de biais d'estimation. Le pessimisme signifie que l'incertitude de l'estimée $\hat{\mathbf{x}}_k$ doit être supérieure à celle de $\mathbf{x}_k$. Dans le cas contraire, nous avons un estimateur optimiste. Avec des estimées pessimistes, on peut éviter une défaillance du système. Pour illustrer cela, nous utilisons l'exemple suivant :

Exemple E.1: Imaginons par exemple le cas d'un système de guidage automatique de voitures sur la route. Supposons que l'objectif de ce système soit de corriger la trajectoire et de déterminer la vitesse de la voiture de sorte qu'elle suive en toute sécurité le centre de la route. Devant une estimée $\hat{e}$ de l'erreur latérale et de sa variance $\Lambda_{\hat{e}}$, on peut définir un contrôleur prudent par le comportement qu'il doit avoir en fonction de ces estimées :

- I. La partie déterministe du contrôleur présente une convergence exponentielle, avec une constante de temps qui est une fonction croissante de $\|\hat{e}\|$. Cela signifie que le retour à la trajectoire nominale se fait en douceur dans le temps pour des erreurs trop importantes de suivi ;

¹Pour A et B matrices définies positives, $A > B$ implique que $A - B$ est définie positive.

²Pour A et B matrices définies positives, $A < B$ implique que $B - A$ est définie positive.

II. La partie stochastique du contrôleur détermine la vitesse de croisière v du véhicule en fonction de $\|\hat{\varepsilon}\|$ et de $\Lambda_{\hat{\varepsilon}}$. Si $\|\hat{\varepsilon}\|$ ou $\Lambda_{\hat{\varepsilon}}$ sont trop importants, il faut réduire v pour augmenter la sécurité dans la manœuvre de correction de trajectoire.

Supposons maintenant le cas de figure suivant : à un moment donné, le véhicule présente une valeur moyenne pour $\hat{\varepsilon}$, mais trop optimiste (i.e., $\Lambda_{\hat{\varepsilon}} \leq \Lambda_{\varepsilon}$). Dans ce cas, le contrôleur va réaliser une manœuvre de correction à une vitesse v importante. $\Lambda_{\hat{\varepsilon}}$ étant trop optimiste, le véhicule peut même se trouver très proche du bord de la route une fois que, instantanément, la vraie erreur ε peut être au delà de $\hat{\varepsilon}$. Cela peut être plus dangereux si la correction ne se fait pas à une vitesse réduite. La correction étant exponentielle dans la temps, la voiture va parcourir une distance importante avant d'être au centre de la route. On risque dans ce cas de voir la voiture sortir de la piste avant que la manœuvre de correction amène le système à un état sûr. Cette défaillance est causée par une mauvaise estimation sur l'incertitude de l'erreur de suivi. Par contre, si $\Lambda_{\hat{\varepsilon}}$ était pessimiste, la manœuvre de correction serait réalisée à basse vitesse et à la même constante de temps que pour le cas précédent. Cela signifie que le véhicule parcourra une distance moins importante avant de trouver sa trajectoire nominale. $\diamond$

En conclusion, il vaut mieux avoir des estimées pessimistes que risquer une manœuvre incohérente à cause d'une estimée optimiste. Cependant, un estimateur trop pessimiste peut être un inconvénient, comme dans le cas de suivi multicibles, où, pour des d'environnements encombrés, il devient difficile de les différencier.

E.3 Filtrage de systèmes linéaires

Les systèmes stochastiques linéaires se présentent sous la forme suivante :

$$\mathbf{x}_{k+1} = \mathbf{F}_k \cdot \mathbf{x}_k + \mathbf{G}_k \cdot \mathbf{u}_k + \mathbf{v}_k, \quad (\text{E.3})$$

$$\mathbf{y}_k = \mathbf{H}_k \cdot \mathbf{x}_k + \mathbf{w}_k, \quad (\text{E.4})$$

A partir de ce modèle, la propriété suivante est directement vérifiée :

Propriété E.1: Si $\mathbf{x}_0$ est de distribution gaussienne et si l'hypothèse E.3 se vérifie, alors la distribution de $\mathbf{x}_k$ sera aussi gaussienne [15]; $\diamond$

Les filtres présentés dans cette section agissent lors de la prise de mesures. Entre-temps, comme ces mesures ne sont pas forcément acquises à chaque instant k , ces filtres réalisent ce que l'on appelle une prédiction de l'état du système. Cela se fait par

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1} \cdot \hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1} \cdot \mathbf{u}_{k-1}, \quad (\text{E.5})$$

$$\Lambda_{\hat{\mathbf{x}}_{k|k-1}} = \mathbf{F}_{k-1} \cdot \Lambda_{\hat{\mathbf{x}}_{k-1}} \cdot \mathbf{F}_{k-1}^T + \Lambda_{\mathbf{v}_{k-1}}, \quad (\text{E.6})$$

où l'indice $k|k-1$ indique une prédiction à l'instant k obtenue à partir des informations acquises à l'instant $k-1$.

L'étape qui s'exécute lors de la prise de mesures est dite étape de correction. C'est le calcul réalisé dans cette étape qui différencie le filtre de Kalman et la méthode intersection de covariances.

E.3.1 Filtre de Kalman

Le filtre de Kalman (*FK*) est une approche récursive pour l'estimation optimale de l'état de systèmes stochastiques linéaires. Sous les hypothèses E.1, E.2 et E.3, Kalman [118] a établi les équations du filtre optimal qui calcule des estimées $\hat{\mathbf{x}}_k$ et $\Lambda_{\hat{\mathbf{x}}_k}$ de $\mathbf{x}_k$ et $\Lambda_{\mathbf{x}_k}$ de sorte que le critère suivant soit satisfait [104] :

$$\min_{\hat{\mathbf{x}}_k} \mathcal{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_k) \cdot (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T | \mathbf{y}_k\} = \min_{\hat{\mathbf{x}}_k} \text{Tr}(\Lambda_{\hat{\mathbf{x}}_k}). \quad (\text{E.7})$$

Pour cela, ce filtre est un estimateur de variance minimale. L'étape de correction peut s'effectuer de différentes façons. Dans notre étude, nous avons utilisé deux formulations différentes : la formulation innovation et la formulation information, données par

$$\text{innovation : } \begin{cases} \Lambda_{\hat{\mathbf{x}}_k} = (\mathbf{I}_n - \mathbf{K}_k \cdot \mathbf{H}_k) \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}}, \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \cdot \nu_k, \end{cases} \quad (\text{E.8})$$

$$\text{information : } \begin{cases} \Lambda_{\hat{\mathbf{x}}_k}^{-1} = \Lambda_{\hat{\mathbf{x}}_{k|k-1}}^{-1} + \mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k, \\ \hat{\mathbf{x}}_k = \Lambda_{\hat{\mathbf{x}}_k} \cdot (\Lambda_{\hat{\mathbf{x}}_{k|k-1}}^{-1} \cdot \hat{\mathbf{x}}_{k|k-1} + \mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{y}_k). \end{cases} \quad (\text{E.9})$$

La différence principale entre ces formulations est que, pour la seconde, la mise à jour de la matrice de covariance $\Lambda_{\hat{\mathbf{x}}_k}$ se fait à partir de son inverse $\Lambda_{\hat{\mathbf{x}}_k}^{-1}$, appelée aussi *matrice d'information* du système. Toutefois, ces formulations sont équivalentes.

Pour la formulation innovation, la correction de la prédiction est proportionnelle à l'erreur de prédiction de la sortie : $\nu_k = \mathbf{y}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_{k|k-1}$, désignée par terme d'*innovation*. Le terme $\mathbf{K}_k$ constitue le gain de Kalman, calculé par :

$$\mathbf{K}_k = \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \mathbf{H}_k^T \cdot \left(\mathbf{H}_k \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \mathbf{H}_k^T + \Lambda_{\mathbf{w}_k} \right)^{-1}. \quad (\text{E.10})$$

Ainsi, le filtre de Kalman est un estimateur en boucle fermée, où le gain $\mathbf{K}_k$ est déterminé en fonction seulement des incertitudes du modèle. Intuitivement, le terme $\mathbf{H}_k \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \mathbf{H}_k^T$ correspond à la projection de l'incertitude de prédiction sur l'espace de mesure. Si l'incertitude due à la prédiction dans cet espace est moins importante que l'incertitude de la mesure, *i.e.*, $\mathbf{H}_k \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \mathbf{H}_k^T \ll \Lambda_{\mathbf{w}_k}$, alors le module du gain $\mathbf{K}_k$ sera petit. Nous disons, dans ce cas, que le filtre fait plus confiance à la prédiction qu'à la mesure, ce qui se traduit dans l'équation (E.8) par le fait que l'estimée de l'état est proche de la prédiction. Au contraire, avec des mesures très précises, le gain $\mathbf{K}_k$ est élevé, et les mesures vont beaucoup influencer l'estimation.

Les propriétés d'optimalité et de convergence du filtre de Kalman linéaire sont aujourd'hui assez bien connues. Cependant, pour que le filtre soit appliqué en pratique, il faut passer par une identification des paramètres du modèle (E.3)-(E.4). Si nous désignons $\hat{\mathbf{F}}_k$, $\hat{\mathbf{G}}_k$, $\hat{\mathbf{H}}_k$, $\Lambda_{\hat{\mathbf{x}}_0}$, $\Lambda_{\hat{\mathbf{v}}_k}$ et $\Lambda_{\hat{\mathbf{w}}_k}$ les estimées approximatives des matrices du modèle linéaire, l'hypothèse E.2 ne se vérifie pas. Ainsi, il est évident que l'estimée $\Lambda_{\hat{\mathbf{x}}_k}$ n'est plus la matrice de covariance de l'erreur d'estimation une fois que le modèle réel diffère de celui utilisé par le filtre [104]. En conséquence, le filtre de Kalman ne sera plus l'estimateur de variance minimale pour le système réel. Pour minimiser le

risque de divergence du filtre, une solution souvent employée consiste à augmenter artificiellement la matrice $\Lambda_{\hat{\mathbf{v}}_k}$ (resp. $\Lambda_{\hat{\mathbf{w}}_k}$) de façon à prendre en compte l'incertitude due à $\hat{\mathbf{F}}_k$ ou à $\hat{\mathbf{G}}_k$ (resp. $\hat{\mathbf{H}}_k$). Nous avons alors un estimateur sous-optimal.

Pour des applications telles que la fusion de données (cf. Exemple E.2), les matrices $\mathbf{F}_k$, $\mathbf{G}_k$ et $\mathbf{H}_k$ sont connues. Dans ces conditions, le filtre de Kalman linéaire peut fournir des estimées optimales pessimistes, comme le dit le théorème suivant :

Théorème E.1: *Supposons connues les matrices $\mathbf{F}_k$, $\mathbf{G}_k$ et $\mathbf{H}_k$ du modèle (E.3)-(E.4). Ainsi, si $\Lambda_{\hat{\mathbf{x}}_0} \geq \Lambda_{\mathbf{x}_0}$, $\Lambda_{\hat{\mathbf{v}}_k} \geq \Lambda_{\mathbf{v}_k}$ et $\Lambda_{\hat{\mathbf{w}}_k} \geq \Lambda_{\mathbf{w}_k}$ pour tout $k > 0$, alors $\Lambda_{\hat{\mathbf{x}}_k} \geq \Lambda_{\mathbf{x}_k}$ et $\Lambda_{\hat{\mathbf{x}}_{k|k-1}} \geq \Lambda_{\mathbf{x}_{k|k-1}}$, où $\Lambda_{\hat{\mathbf{v}}_k}$ et $\Lambda_{\hat{\mathbf{w}}_k}$ sont des estimées consistantes ou pessimistes des covariances réelles $\Lambda_{\mathbf{v}_k}$ et $\Lambda_{\mathbf{w}_k}$ des bruits $\mathbf{v}_k$ et $\mathbf{w}_k$, respectivement.*

Preuve : voir [104], théorème 7.6. ◇

Ainsi, pour ces cas, le réglage du filtre repose sur un choix approprié des matrices $\Lambda_{\hat{\mathbf{x}}_0}$, $\Lambda_{\hat{\mathbf{v}}_k}$ et $\Lambda_{\hat{\mathbf{w}}_k}$. L'incertitude des mesures $\Lambda_{\hat{\mathbf{w}}_k}$ est déterminée expérimentalement. $\Lambda_{\hat{\mathbf{v}}_k}$ et $\Lambda_{\hat{\mathbf{x}}_0}$ doivent en général être déterminées de façon empirique. Ce qu'il est important de noter dans le théorème E.1 c'est que si $\Lambda_{\hat{\mathbf{x}}_0}$, $\Lambda_{\hat{\mathbf{v}}_k}$ et $\Lambda_{\hat{\mathbf{w}}_k}$ sont supérieurs mais proches des vrais valeurs $\Lambda_{\mathbf{x}_0}$, $\Lambda_{\mathbf{v}_k}$ et $\Lambda_{\mathbf{w}_k}$, alors nous pouvons avoir des estimées consistantes et pas trop pessimistes.

E.3.2 Intersection de covariances

Le filtre basé sur l'intersection de covariances a été proposé par Julier et Uhlmann dans [115]. Ce filtre a été présenté comme une approche pour la fusion de données incertaines quand on ne peut pas déterminer leur degré de corrélation. Son application au problème de filtrage est présentée ici, suivant les lignes de conception données dans [85].

L'étape de correction du filtre intersection de covariances peut s'effectuer sous l'une des formes suivantes :

$$\text{innovation : } \begin{cases} \Lambda_{\hat{\mathbf{x}}_k} = \frac{1}{\omega} \cdot (\mathbf{I}_n - \mathbf{K}_k \cdot \mathbf{H}_k) \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}}, \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \cdot \nu_k. \end{cases} \quad (\text{E.11})$$

$$\text{information : } \begin{cases} \Lambda_{\hat{\mathbf{x}}_k}^{-1} = \omega \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}}^{-1} + (1 - \omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k, \\ \hat{\mathbf{x}}_k = \Lambda_{\hat{\mathbf{x}}_k} \cdot \left(\omega \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}}^{-1} \cdot \hat{\mathbf{x}}_{k|k-1} + (1 - \omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{y}_k \right). \end{cases} \quad (\text{E.12})$$

Pour la forme innovation, le gain du filtre est calculé par :

$$\mathbf{K}_k = (1 - \omega) \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \mathbf{H}_k^T \cdot \left((1 - \omega) \cdot \mathbf{H}_k \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \mathbf{H}_k^T + (\omega) \cdot \Lambda_{\hat{\mathbf{w}}_k} \right)^{-1} \quad (\text{E.13})$$

Comme nous pouvons le voir, l'intersection de covariances utilise un scalaire ω tel que $0 \leq \omega \leq 1$. Ce paramètre doit être optimisé en ligne. En effet, l'équation (E.12) montre que la matrice d'information corrigée $\Lambda_{\hat{\mathbf{x}}_k}^{-1}$ est une combinaison convexe entre la matrice d'information de la prédiction $\Lambda_{\hat{\mathbf{x}}_{k|k-1}}^{-1}$ et la matrice d'information de la mesure $\mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k$, d'où le nom intersection

de covariances [115]. Ainsi, ω représente un degré de liberté en plus, qui permet à l'algorithme de fournir des estimées conservatives même si des inter-corrélations sont inconnues. Cette propriété est énoncée dans le théorème suivant, adapté de [115][85] :

Théorème E.2: *Supposons que, à l'instant k , nous disposons d'une prédiction conservative de l'état du système (E.3)-(E.4) et d'une mesure conservative de la sortie :*

$$\mathcal{E}\{\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_{k|k-1}\} = \mathbf{0} \quad \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \geq \Lambda_{\mathbf{x}_{k|k-1}} \quad , \quad (\text{E.14})$$

$$\mathcal{E}\{\hat{\mathbf{y}}_k - \mathbf{y}_k\} = \mathbf{0} \quad \Lambda_{\hat{\mathbf{w}}_k} \geq \Lambda_{\mathbf{w}_k} \quad , \quad (\text{E.15})$$

où $\hat{\mathbf{y}}_k$ est la mesure effectuée à l'instant k qui correspond à la vraie sortie $\mathbf{y}_k$ du modèle perturbée par le bruit $\mathbf{w}_k$. Supposons aussi que les erreurs de prédiction comme aussi celles de mesure soient corrélées :

$$\mathcal{E}\{(\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_{k|k-1}) \cdot (\hat{\mathbf{y}}_k - \mathbf{y}_k)^T\} = \Lambda_{\mathbf{xy}} \geq \mathbf{0}. \quad (\text{E.16})$$

Alors, le filtre intersection de covariances fournit toujours une estimée corrigée $\hat{\mathbf{x}}_k$ conservative, même si $\Lambda_{\mathbf{xy}}$ est inconnu.

Preuve : Soit $\tilde{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_{k|k-1}$, $\tilde{\mathbf{y}}_k = \hat{\mathbf{y}}_k - \mathbf{y}_k$ et $\tilde{\mathbf{x}}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k$ les vraies erreurs de prédiction, de mesure et de correction. Par propagation d'erreurs sur l'équation (E.12), nous avons

$$\tilde{\mathbf{x}}_k = \Lambda_{\tilde{\mathbf{x}}_k} \cdot \left(\omega \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \tilde{\mathbf{x}}_{k|k-1} + (1 - \omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \tilde{\mathbf{y}}_k \right) \quad (\text{E.17})$$

et ainsi

$$\mathcal{E}\{\tilde{\mathbf{x}}_k\} = \Lambda_{\tilde{\mathbf{x}}_k} \cdot \left(\omega \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \mathcal{E}\{\tilde{\mathbf{x}}_{k|k-1}\} + (1 - \omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathcal{E}\{\tilde{\mathbf{y}}_k\} \right). \quad (\text{E.18})$$

comme la prédiction et la mesure sont pessimistes, alors $\mathcal{E}\{\tilde{\mathbf{x}}_{k|k-1}\} = \mathcal{E}\{\tilde{\mathbf{y}}_k\} = \mathbf{0}$, et par conséquent $\mathcal{E}\{\tilde{\mathbf{x}}_k\} = \mathbf{0}$. Ainsi, nous avons déjà montré que

$$\mathcal{E}\{\hat{\mathbf{x}}_k - \mathbf{x}_k\} = \mathbf{0}. \quad (\text{E.19})$$

Pour prouver que $\hat{\mathbf{x}}_k$ est pessimiste, il nous reste alors démontrer que $\Lambda_{\hat{\mathbf{x}}_k} \geq \Lambda_{\mathbf{x}_k}$. Pour le faire, nous partons du calcul de $\mathcal{E}\{\tilde{\mathbf{x}}_k \cdot \tilde{\mathbf{x}}_k^T\}$ à partir de l'équation (E.17) :

$$\begin{aligned} \mathcal{E}\{\tilde{\mathbf{x}}_k \cdot \tilde{\mathbf{x}}_k^T\} &= \mathcal{E}\left\{ \Lambda_{\tilde{\mathbf{x}}_k} \cdot \left(\omega \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \tilde{\mathbf{x}}_{k|k-1} + (1 - \omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \tilde{\mathbf{y}}_k \right) \right. \\ &\quad \cdot \left(\omega \cdot \tilde{\mathbf{x}}_{k|k-1}^T \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} + (1 - \omega) \cdot \tilde{\mathbf{y}}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k \right) \cdot \Lambda_{\tilde{\mathbf{x}}_k} \left. \right\} \\ &= \Lambda_{\tilde{\mathbf{x}}_k} \cdot \left(\omega^2 \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{x}_{k|k-1}} \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} + (1 - \omega)^2 \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \tilde{\mathbf{y}}_k \right. \\ &\quad \left. + \omega \cdot (1 - \omega) \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{xy}} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k + \omega \cdot (1 - \omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{xy}}^T \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \right) \cdot \Lambda_{\tilde{\mathbf{x}}_k} \end{aligned}$$

Pour vérifier si l'estimée $\hat{\mathbf{x}}_k$ est consistante, il faut démontrer que

$$\Lambda_{\hat{\mathbf{x}}_k} - \Lambda_{\mathbf{x}_k} = \Lambda_{\tilde{\mathbf{x}}_k} - \mathcal{E}\{\tilde{\mathbf{x}}_k \cdot \tilde{\mathbf{x}}_k^T\} \geq \mathbf{0}, \quad (\text{E.20})$$

qui, après pré- et post-multiplication par $\Lambda_{\tilde{\mathbf{x}}_k}^{-1}$, devient équivalent à :

$$\Phi = \Lambda_{\tilde{\mathbf{x}}_k}^{-1} - \Lambda_{\tilde{\mathbf{x}}_k}^{-1} \cdot \mathcal{E}\{\tilde{\mathbf{x}}_k \cdot \tilde{\mathbf{x}}_k^T\} \cdot \Lambda_{\tilde{\mathbf{x}}_k}^{-1} \geq \mathbf{0}. \quad (\text{E.21})$$

Si l'on remplace $\mathcal{E}\{\tilde{\mathbf{x}}_k \cdot \tilde{\mathbf{x}}_k^T\}$ par le développement qui nous avons obtenu précédemment, nous avons

$$\begin{aligned} \Phi &= \Lambda_{\tilde{\mathbf{x}}_k}^{-1} - \omega^2 \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{x}_{k|k-1}} \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} - (1-\omega)^2 \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \tilde{\mathbf{y}}_k \\ &\quad - \omega \cdot (1-\omega) \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{xy}} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k + \omega \cdot (1-\omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{xy}}^T \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1}. \end{aligned} \quad (\text{E.22})$$

Après pré- et post-multiplication des inéquations (E.14) et (E.15) par $\Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1}$ et $\Lambda_{\tilde{\mathbf{w}}_k}^{-1}$, respectivement, nous obtenons

$$\Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \geq \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{x}_{k|k-1}} \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1}, \quad (\text{E.23})$$

$$\Lambda_{\tilde{\mathbf{w}}_k}^{-1} \geq \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{w}_k} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1}. \quad (\text{E.24})$$

On calcule la covariance de l'estimée $\hat{\mathbf{x}}_k$ par la formule (E.11), ce qui donne

$$\Lambda_{\hat{\mathbf{x}}_k}^{-1} = \omega \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} + (1-\omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k, \quad (\text{E.25})$$

$$\geq \omega \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{x}_{k|k-1}} \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} + (1-\omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{w}_k} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k. \quad (\text{E.26})$$

Ainsi, en utilisant les équations (E.23) et (E.24), nous avons déterminé une borne inférieure pour $\Lambda_{\hat{\mathbf{x}}_k}^{-1}$. Cela nous permet de déterminer aussi une borne inférieure de Φ :

$$\begin{aligned} \Phi &\geq \omega \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{x}_{k|k-1}} \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} + (1-\omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{w}_k} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k \\ &\quad - \omega^2 \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{x}_{k|k-1}} \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} - (1-\omega)^2 \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \tilde{\mathbf{y}}_k \\ &\quad - \omega \cdot (1-\omega) \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{xy}} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k + \omega \cdot (1-\omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{xy}}^T \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1}, \\ \Phi &\geq \omega \cdot (1-\omega) \cdot \left(\Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{x}_{k|k-1}} \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} + \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{w}_k} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k \right. \\ &\quad \left. - \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \Lambda_{\mathbf{xy}} \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k - \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \Lambda_{\mathbf{xy}}^T \cdot \Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \right) \\ \Phi &\geq \omega \cdot (1-\omega) \cdot \mathcal{E} \left\{ (\Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \tilde{\mathbf{x}}_{k|k-1} - \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \tilde{\mathbf{y}}_k) \cdot (\Lambda_{\tilde{\mathbf{x}}_{k|k-1}}^{-1} \cdot \tilde{\mathbf{x}}_{k|k-1} - \mathbf{H}_k^T \cdot \Lambda_{\tilde{\mathbf{w}}_k}^{-1} \cdot \tilde{\mathbf{y}}_k)^T \right\}. \end{aligned}$$

Comme, par définition la covariance du terme $\mathcal{E}\{\cdot\}$ est définie positive et $\omega \cdot (1-\omega) \geq 0$ pour $0 \leq \omega \leq 1$, alors $\Phi \geq \mathbf{0}$, ce qui prouve que l'équation (E.20) est vraie. $\diamond$

Le filtre de Kalman ne peut être utilisé pour ce problème de filtrage que si $\Lambda_{\mathbf{xy}}$ est connu [59]. En présence de corrélations croisées inconnues, le filtre de Kalman peut présenter des estimées trop optimistes, même en semblant converger. Le filtre intersection de covariances a été initialement proposé pour traiter le problème d'estimation décentralisée [115]. Il garantit une estimation conservative en l'absence de mesure de corrélation.

Nous avons vu dans le Théorème E.2 que le pessimisme des estimées est préservé pour tout ω dans l'intervalle $0 \leq \omega \leq 1$. Cependant, pour éviter des résultats trop pessimistes, ce qui pourrait rajouter un bruit excessif au modèle, ω est déterminé de sorte que le déterminant de $\Lambda_{\hat{\mathbf{x}}_k}$ soit minimisé. La solution à ce problème est calculée en ligne de façon itérative en utilisant

la technique *golden section search* [179], qui est bien adaptée aux problèmes d'optimisation de variables limitées par des intervalles. Cette méthode requiert la connaissance de la forme analytique de la dérivée du déterminant de $\Lambda_{\hat{\mathbf{x}}_k}$ par rapport à ω . Nous citons ici cette relation :

$$\frac{d}{d\omega} \det(\Lambda_{\hat{\mathbf{x}}_k}) = \frac{-\text{Tr} \left(\left(\omega \cdot \Lambda_{\hat{\mathbf{x}}_k|k-1}^{-1} + (1-\omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k \right)^{-1} \cdot \left(\Lambda_{\hat{\mathbf{x}}_k|k-1}^{-1} - \mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k \right) \right)}{\det \left(\omega \cdot \Lambda_{\hat{\mathbf{x}}_k|k-1}^{-1} + (1-\omega) \cdot \mathbf{H}_k^T \cdot \Lambda_{\hat{\mathbf{w}}_k}^{-1} \cdot \mathbf{H}_k \right)}. \quad (\text{E.27})$$

Une autre formulation présentée dans [90] facilite la procédure d'optimisation, en changeant la formulation du filtre.

E.3.3 Exemple d'application

Exemple E.2: *Considérons le problème de la fusion de deux mesures $\hat{\mathbf{p}}_a$ et $\hat{\mathbf{p}}_b$, qui sont des estimées pessimistes de la même grandeur $\mathbf{p}$. Ce problème est généralement modélisé par le système stochastique linéaire suivant*

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k, \quad (\text{E.28})$$

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k, \quad (\text{E.29})$$

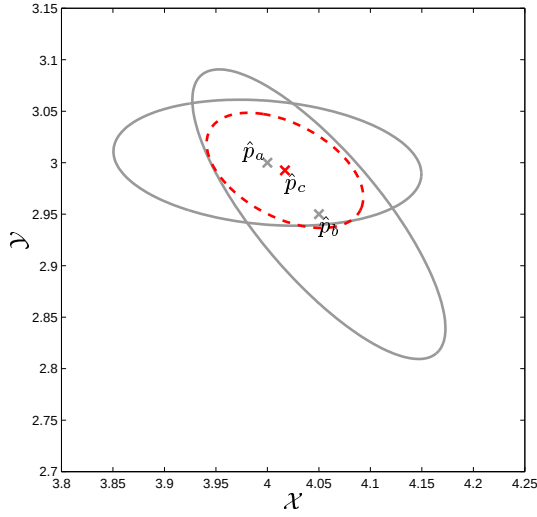
où l'on considère que, lorsque la mesure $\mathbf{y}_k = \hat{\mathbf{p}}_b$ est disponible, la prédiction de l'état est $\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{p}}_a$, avec pour covariances respectives $\Lambda_{\hat{\mathbf{p}}_b}$ et $\Lambda_{\hat{\mathbf{p}}_a}$. La fusion entre $\hat{\mathbf{p}}_a$ et $\hat{\mathbf{p}}_b$ conduit à calculer $\hat{\mathbf{p}}_c$ par application des équations de correction des filtres stochastiques. Les illustrations sur la Figure E.1 montrent en couleur grise $\hat{\mathbf{p}}_a$, $\hat{\mathbf{p}}_b$ et leurs ellipses d'incertitude 3σ . Le résultat d'estimation $\mathbf{p}_c$ et l'ellipse d'incertitude 3σ de sa covariance $\Lambda_{\hat{\mathbf{p}}_c}$, calculés par le filtre de Kalman et par intersection de covariances sont présentés sur les Figures E.1(a) et E.1(b), respectivement. Nous remarquons que l'intersection de covariances fournit une estimée plus pessimiste que le filtre de Kalman. Dans ce cas, la procédure d'optimisation de ω a déterminé $\omega = 0,5623$ comme valeur optimale pour le filtre intersection de covariances. En fait, si nous traçons les ellipses d'incertitude correspondant à $\Lambda_{\hat{\mathbf{p}}_c}$ pour différentes valeurs de ω , nous observons que l'incertitude de l'estimée reste toujours entre celles correspondant à $\Lambda_{\hat{\mathbf{p}}_b}$ et $\Lambda_{\hat{\mathbf{p}}_a}$ (voir Figure E.1(c)). Ainsi, le fameux concept "deux capteurs valent mieux qu'un seul"³ [3] n'est pas valable pour ce filtre.

Cependant, sous certaines conditions, le minimum du critère d'optimisation de ω peut être sur une des limites de l'intervalle $0 \leq \omega \leq 1$. Cela serait le cas de notre problème de fusion si la matrice covariance $\Lambda_{\hat{\mathbf{p}}_a}$ était divisée par quatre, comme nous pouvons le voir sur la Figure E.1(d). Pour ce cas, $\omega = 1$, ce qui donne $\hat{\mathbf{p}}_c = \hat{\mathbf{p}}_a$ et $\Lambda_{\hat{\mathbf{p}}_c} = \Lambda_{\hat{\mathbf{p}}_a}$. Ce résultat est encore pessimiste, une fois que $\hat{\mathbf{p}}_a$ l'est aussi (cf. théorème E.2). $\diamond$

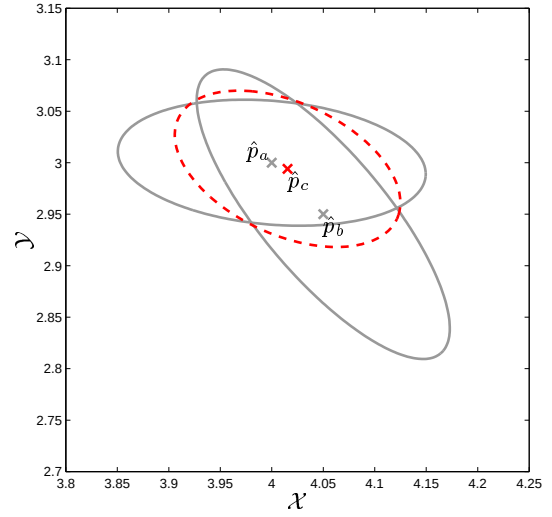
E.3.4 Intersection de covariances modifiée

L'Exemple E.2 illustre bien le fait que dans certains cas l'optimisation de ω peut résulter en $\omega = 0$ ou en $\omega = 1$ (cf. Figure E.1(d)). Pour certaines applications cela n'est pas souhaitable, une

³L'incertitude résultant de la fusion de deux observations est moins importante que l'incertitude des observations.



(a) Fusion par filtre de Kalman



(b) Fusion par Intersection de Covariances

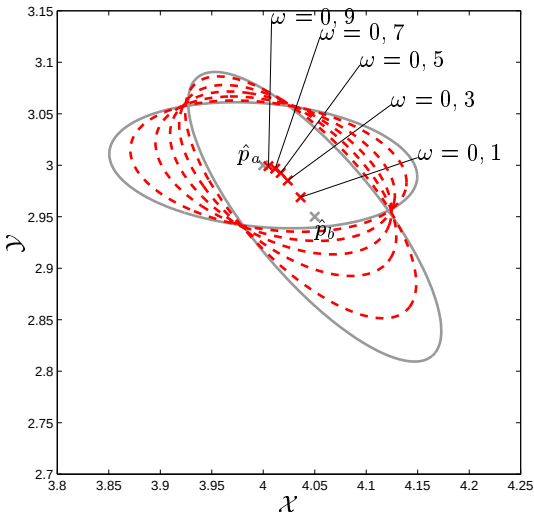
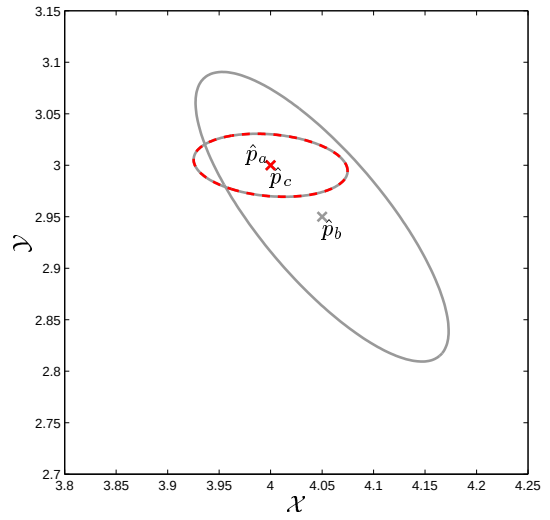
(c) Trajectoire de l'estimée en fonction de ω (d) Cas pour lequel $\omega = 1$

FIG. E.1 – Exemple d'utilisation des filtres stochastiques pour la fusion de données.

fois que l'estimée sera égale à la prédiction (resp. les mesures) si nous avons des mesures (resp. des prédictions) très incertaines. Cela peut réduire la vitesse de réaction du filtre à des changements rapides sur le modèle, qui peuvent se traduire par une injection extra d'incertitude.

Ainsi, pour que le filtre présente une dynamique plus importante, nous proposons que l'intervalle d'optimisation de ω soit $\gamma \leq \omega \leq 1 - \gamma$, où $\gamma \in [0, \frac{1}{2}]$ est un paramètre défini en fonction de l'application. Si $\gamma = 0$, nous sommes dans le cas du filtre classique décrit dans E.3.2. A l'opposé, avec $\gamma = \frac{1}{2}$, le seul résultat d'optimisation possible est $\omega = 0,5$. Dans ce cas, les estimées fournies par l'intersection de covariances et le filtre de Kalman sont les mêmes, à la différence que la matrice $\Lambda_{\hat{\mathbf{x}}_k}$ calculée par intersection de covariances est deux fois celle calculée par le filtre de Kalman. Pour les valeurs intermédiaires de γ , les mesures vont toujours influencer les estimées (sauf, si $\Lambda_{\mathbf{w}_k}^{-1} = \mathbf{0}$).

E.4 Filtrage de systèmes non-linéaires

Le problème de filtrage de systèmes non-linéaires est plus difficile que celui de systèmes linéaires [15]. Pour ces systèmes, les fonctions $\mathbf{f}_k$ et $\mathbf{h}_k$ du modèle (E.1)-(E.2) étant non-linéaires, la distribution de $\mathbf{x}_k$ ne sera plus gaussienne même si $\mathbf{v}_k$ et $\mathbf{w}_k$ le sont. Pour la majeure partie des cas non-linéaires, la détermination de la distribution de $\mathbf{x}_k$ ne peut se faire que de façon non-récursive en utilisant, par exemple, une intégration de Monte Carlo. Cela est très coûteux, pas seulement en temps de calcul, mais aussi en mémoire car il est nécessaire stocker toutes les données du système dès que $k = 0$. Dans ce cas, des approches sous-optimales récursives telles que filtre de Kalman étendu et ses variantes ont rapidement séduit les ingénieurs et scientifiques par leur simplicité d'implantation.

E.4.1 Filtre de Kalman étendu

Le filtre de Kalman étendu (*FKE*) est obtenu en considérant une approximation au premier ordre du modèle (E.1)-(E.2). Cette approximation donne les équations suivantes pour la prédiction :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k), \quad (\text{E.30})$$

$$\Lambda_{\hat{\mathbf{x}}_{k|k-1}} = \nabla_{\hat{\mathbf{x}}_{k-1}} \mathbf{f}_{k-1} \cdot \Lambda_{\hat{\mathbf{x}}_{k-1}} \cdot \nabla_{\hat{\mathbf{x}}_{k-1}}^T \mathbf{f}_{k-1} + \Lambda_{\mathbf{v}_k}, \quad (\text{E.31})$$

L'étape de correction s'écrit :

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \cdot \nu_k, \\ \Lambda_{\hat{\mathbf{x}}_k} &= \left(\mathbf{I}_n - \mathbf{K}_k \cdot \nabla_{\hat{\mathbf{x}}_{k|k-1}}^T \mathbf{h}_k \right) \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}}, \end{aligned} \quad (\text{E.32})$$

où le terme d'innovation est donné par $\nu_k = \mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})$, et le gain du *FKE* est

$$\mathbf{K}_k = \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \nabla_{\hat{\mathbf{x}}_{k|k-1}}^T \mathbf{h}_k \cdot \left(\nabla_{\hat{\mathbf{x}}_{k|k-1}} \mathbf{h}_k \cdot \Lambda_{\hat{\mathbf{x}}_{k|k-1}} \cdot \nabla_{\hat{\mathbf{x}}_{k|k-1}}^T \mathbf{h}_k + \Lambda_{\mathbf{w}_k} \right)^{-1}. \quad (\text{E.33})$$

E.4.2 Filtre de Kalman étendu itératif

Le filtre de Kalman étendu itératif (*FKEI*) [104, 15] est une approche qui utilise des itérations locales dans l'objectif de minimiser le risque d'estimation biaisée due à des erreurs importants de linéarisation de l'équation de mesure du modèle stochastique. Pour le *FKEI*, les équations de prédiction sont les mêmes que celles du *FKE*. Cependant, l'étape de correction se réalise dans une boucle où l' i -ième+1 itération correspond à :

$$\begin{aligned}\hat{\mathbf{x}}_k^{(i+1)} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k^{(i)} \cdot \left\{ \mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^{(i)}) - \nabla_{\hat{\mathbf{x}}_k^{(i)}}^T \mathbf{h}_k \cdot (\hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_k^{(i)}) \right\}, \\ \mathbf{\Lambda}_{\hat{\mathbf{x}}_k^{(i+1)}} &= \left(\mathbf{I}_n - \mathbf{K}_k^{(i)} \cdot \nabla_{\hat{\mathbf{x}}_k^{(i)}}^T \mathbf{h}_k \right) \cdot \mathbf{\Lambda}_{\hat{\mathbf{x}}_{k|k-1}},\end{aligned}\tag{E.34}$$

où

$$\mathbf{K}_k^{(i)} = \mathbf{\Lambda}_{\hat{\mathbf{x}}_{k|k-1}} \cdot \nabla_{\hat{\mathbf{x}}_k^{(i)}}^T \mathbf{h}_k \cdot \left(\nabla_{\hat{\mathbf{x}}_k^{(i)}} \mathbf{h}_k \cdot \mathbf{\Lambda}_{\hat{\mathbf{x}}_{k|k-1}} \cdot \nabla_{\hat{\mathbf{x}}_k^{(i)}}^T \mathbf{h}_k + \mathbf{\Lambda}_{\mathbf{w}_k} \right)^{-1}\tag{E.35}$$

$$\hat{\mathbf{x}}_k^{(0)} = \hat{\mathbf{x}}_{k|k-1}, \quad \mathbf{\Lambda}_{\hat{\mathbf{x}}_k^{(0)}} = \mathbf{\Lambda}_{\hat{\mathbf{x}}_{k|k-1}}.\tag{E.36}$$

A la fin des itérations, les estimées $\hat{\mathbf{x}}_k$ et $\mathbf{\Lambda}_{\hat{\mathbf{x}}_k}$ sont données par $\hat{\mathbf{x}}_k^{(i+1)}$ et $\mathbf{\Lambda}_{\hat{\mathbf{x}}_k^{(i+1)}}$. On remarque que pour une seule itération, nous avons les mêmes équations que le *FKE*. Il a été montré dans [18]⁴, que le *FKEI* est équivalent à l'application de la méthode de Gauss-Newton pour la maximisation de la fonction de vraisemblance de l'état étant donné les mesures $\mathbf{y}_k$, *i.e.*,

$$\hat{\mathbf{x}}_k = \arg_{\mathbf{x}} \max(f(\mathbf{x}|\mathbf{y}_k)).\tag{E.37}$$

Pour utiliser des itérations locales, cette méthode présente garantie de convergence au maximum local de $f(\mathbf{x}|\mathbf{y}_k)$ le plus proche de la prédiction $\hat{\mathbf{x}}_{k|k-1}$. Les maximum locaux d'une fonction de densité de probabilité sont les modes de la distribution. Ainsi, si cette fonction correspond à une distribution gaussienne, alors pour un nombre important d'itérations nous avons comme estimée $\hat{\mathbf{x}}_k$ la moyenne, ce qui est optimal.

E.4.3 Limitations

Les approches ici exposées présentent une forme récurrente, ce qui est une caractéristique de grand intérêt pour des applications temps réel. Cependant, pour le *FKE*, il existe un certain nombre de risques potentiels dans des applications réelles :

- I. **Risque de divergence** : Le modèle (E.1)-(E.2) étant linéarisé, l'Hypothèse E.2 ne se vérifie pas. Ainsi, l'estimée $\mathbf{\Lambda}_{\hat{\mathbf{x}}_k}$ n'est plus la matrice de covariance de l'erreur d'estimation une fois que le modèle réel diffère de celui utilisé par le filtre [104]. En conséquence, le *FKE* ne peut pas fournir une estimation de variance minimale et diverger. La divergence se vérifie quand le filtre devient trop optimiste (*i.e.*, $\mathbf{\Lambda}_{\hat{\mathbf{x}}_k} \ll \mathbf{\Lambda}_{\mathbf{x}_k}$), et ainsi les nouvelles observations sont pratiquement ignorées et il ne va plus arriver à suivre l'état du système ;

⁴Peut être les auteurs ne le savaient pas, mais cela était déjà un résultat connu dans les années 70 [104].

- II. **Risque d'estimation biaisée** : Une estimée biaisée de $\hat{\mathbf{x}}_{k-1}$ se traduit par une prédiction $\hat{\mathbf{x}}_{k|k-1}$ biaisée, ainsi que par une erreur de calcul de $\mathbf{\Lambda}_{\hat{\mathbf{x}}_{k|k-1}}$;
- III. **Risque de distributions fortement non-gaussiennes** : Si les vraies incertitudes $\mathbf{v}_k$ et $\mathbf{w}_k$ sont trop importants et si $\mathbf{f}_{k-1}$ et $\mathbf{h}_k$ sont fortement non-linéaires autour du point de linéarisation $\hat{\mathbf{x}}_{k|k-1}$, alors il est probable que la vraie distribution de $\mathbf{x}_k$ soit multimodale.

Heureusement, il y a des solutions bien connues pour limiter les dégâts et obtenir un algorithme de performances satisfaisantes :

- Pour minimiser le risque de divergence du filtre, une solution souvent employée consiste à augmenter artificiellement les matrices $\mathbf{\Lambda}_{\mathbf{v}_k}$ et $\mathbf{\Lambda}_{\mathbf{w}_k}$ de sorte à prendre en compte l'incertitude due à la linéarisation [15][104]. Cependant, cette solution dépend beaucoup de l'application, et reste encore empirique. En plus, la matrice de covariance finale aura perdu son sens physique, et ne correspondrait plus à la vraie dispersion des erreurs sur $\hat{\mathbf{x}}_k$;
- Une analyse de l'observabilité locale du modèle peut permettre de sélectionner les mesures qui apportent le plus d'information pour le filtre sur l'état du système [24] ;
- Comme nous l'avons présenté dans la section E.4.2, le *FKEI* est une solution proposée pour minimiser le risque d'estimation biaisée. Pour utiliser la méthode de Gauss-Newton pour la maximisation de la fonction de vraisemblance de l'état $\mathbf{x}_k$ étant donné les mesures $\mathbf{y}_k$ [18, 104], sa convergence est garantie si cette fonction est gaussienne. Cependant, cette approche reste encore très sensible à des maximums locaux du critère (E.37), causés par une distribution multimodale de $\mathbf{x}_k$, qui peut être en conséquence des mesures $\mathbf{y}_k$ qui permettent des hypothèses multiples à $\mathbf{x}_k$. En plus, une fois un maximum local trouvé, le filtre présente en général une estimation très optimiste, tout en supposant que le résultat obtenu est le mode principal de la distribution ;
- Une autre solution pour minimiser le risque d'estimation biaisée est celle proposée par Julier *et al.* [114], et nommée plus récemment *Unscented Kalman Filter* [209]. Cette méthode utilise l'échantillonnage sélectif (cf. annexe B.4.2) pour propager correctement les moyennes et les covariances des distributions du modèle stochastique ;
- Si $\mathbf{\Lambda}_{\mathbf{v}_k}$ et $\mathbf{\Lambda}_{\mathbf{w}_k}$ ne sont pas connues par avance, ces matrices peuvent être estimées dans le cadre du filtre de Kalman adaptatif [111]. En plus, si les erreurs de mesure et du processus sont corrélées, l'intersection de covariances peut être employée pour fournir des estimées plus conservatives en utilisant le modèle linéarisé du système ;
- Le filtre de Kalman multi-hypothèses est une formulation permettant d'attaquer le problème des distributions multimodales [21, 15]. Une nouvelle formulation pour ce filtre est introduite dans [54], où des distributions multimodales sont approximées par des mélanges de gaussiennes. Pour des distributions plus complexes, le filtre à particules ou filtre de Monte Carlo séquentiel est une approche qui a trouvé des applications réelles récemment [23, 205, 106]. En effet, ce filtre approxime la distribution de l'estimée par un ensemble important d'échantillons ;
- La complexité de calcul de la phase de correction du filtre de Kalman s'accroît en $O(N^3)$, si l'on considère que N mesures sont disponibles à l'instant k . Si les mesures sont indépendantes, elles peuvent être intégrées les unes après les autres dans le filtre. Cependant, si ces mesures

sont corrélées comme pour la localisation de robots mobiles par cartes géométriques, une solution très employée consiste alors à négliger les corrélations et à les considérer comme des mesures indépendantes [8]. Nous avons ainsi le filtre de Kalman séquentiel [59] qui présente une complexité en $O(N)$. Cependant, ce filtre peut diverger si la séquence de présentation des données n'est pas bien choisie. Une stratégie très utilisée consiste à intégrer les mesures les plus précises en premier [21].

En conclusion, toutes ces problématiques contribuent à la difficulté de mise au point du filtre de Kalman étendu. Mais, des résultats satisfaisants ont été obtenus pour des systèmes sans fortes non-linéarités et d'incertitude réduite [46][53]. Pour ces cas, maintenir des estimées pessimistes est primordial pour la convergence du filtre.

Etalonnage du système multicapteur

Notre approche de construction de cartes locales est fondée sur la fusion d'indices extraits d'une image télémétrique $\mathcal{L}$ et d'une image vidéo $\mathcal{I}$. Pour effectuer la fusion d'indices, nous avons besoin de connaître la relation suivante entre les repères $\mathcal{R}^C$ et $\mathcal{R}^L$:

$$\phi = g(u). \quad (\text{F.1})$$

Avec la relation (F.1), nous pouvons calculer, dans le plan du télémètre laser, l'angle d'observation ϕ d'une droite verticale qui se trouve sur la colonne u de l'image vidéo. Cette annexe présente une approche pour l'estimation de la relation $g(\cdot)$. Pour y arriver, nous avons d'abord fait une analyse de la géométrie du système multicapteur. Ensuite, une simulation nous a permis de caractériser l'influence des différents paramètres sur la forme de $g(\cdot)$. Enfin, nous avons mis au point une technique simplifiée d'étalonnage du système multicapteur en utilisant des données expérimentales.

F.1 Analyse géométrique

Nous avons sur la Figure F.1(a) une représentation simplifiée des repères associés au système multicapteur, *i.e.*, le repère image $\mathcal{R}^I$, le repère caméra $\mathcal{R}^C$ et le repère télémètre laser $\mathcal{R}^L$ (voir section 3.2). Notre analyse concerne la projection des repères capteur sur le plan de balayage $\mathcal{X}^L \times \mathcal{Y}^L$ du télémètre. Nous supposons valides les hypothèses suivantes sur la géométrie de ce système :

- Les axes $\mathcal{X}^C$ et $\mathcal{X}^I$ sont parallèles et de sens contraires, comme le sont aussi $\mathcal{Y}^C$ et $\mathcal{Y}^I$;

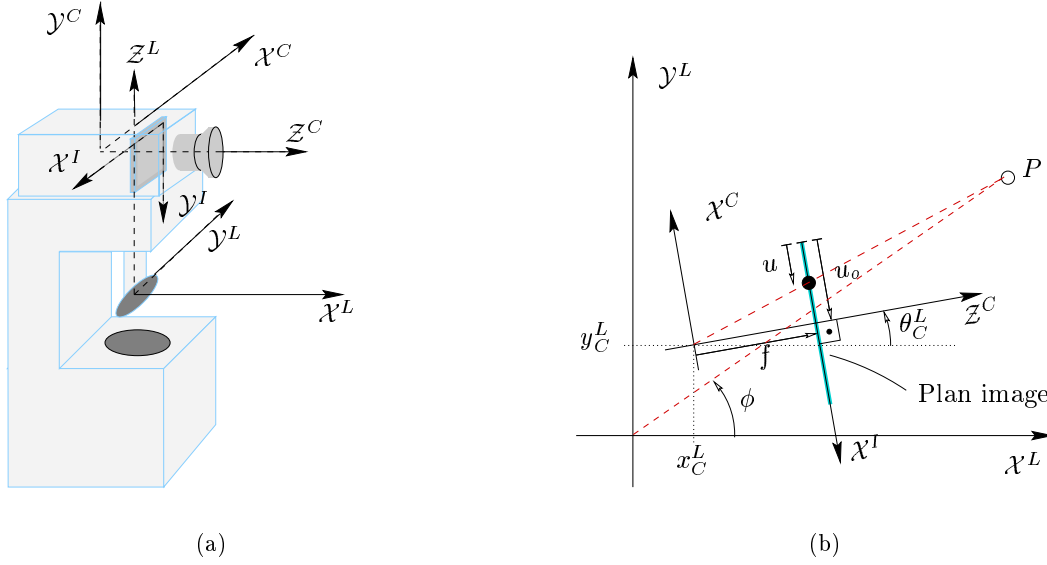


FIG. F.1 – (a) Configuration simplifiée des repères du système multicapteur. (b) Projection des repères dans le plan de balayage laser.

- Les axes $\mathcal{Z}^L$ et $\mathcal{Y}^C$ sont parallèles et de même sens ;

En accord avec ces hypothèses, la Figure F.1(b) illustre la projection des axes d'intérêt $\mathcal{X}^C$, $\mathcal{Z}^C$ et $\mathcal{X}^I$ sur le plan de balayage $\mathcal{X}^L \times \mathcal{Y}^L$. Dans la suite, lorsque nous citons ces axes, nous faisons en fait référence aux projections dans le plan laser. Sous les hypothèses ci-dessus, l'écart de positionnement de ces deux capteurs peut être donné par $\mathbf{t}_C^L = (x_C^L, y_C^L)$, qui sont les coordonnées de l'origine du repère $\mathcal{R}^C$ sur le plan laser, et θ_C^L qui est l'angle entre $\mathcal{Z}^C$ et $\mathcal{X}^L$. Le triplet $(x_C^L, y_C^L, \theta_C^L)$ représente l'écart de positionnement relatif des deux capteurs lors de leur assemblage. Nous utilisons ces variables pour analyser la validité de la transformation $g(\cdot)$ proposée dans cette annexe.

Dans la Figure F.1(b), nous avons un point P qui est observé par le télémètre laser sous la direction ϕ . Sa projection dans le plan image s'effectue sur la colonne u de l'image vidéo. Soit $\mathbf{p}^L = (x^L, y^L)^T$ les coordonnées de P dans le système $\mathcal{X}^L \times \mathcal{Y}^L$, et $\mathbf{p}^C = (z^C, x^C)^T$ les coordonnées de P dans le système $\mathcal{Z}^C \times \mathcal{X}^C$. Etant donné $(x_C^L, y_C^L, \theta_C^L)$, $\mathbf{p}^L$ et $\mathbf{p}^C$ sont reliés par

$$\mathbf{p}^C = \mathbf{R}(\theta_C^L) \cdot (\mathbf{p}^L - \mathbf{t}_C^L), \quad (\text{F.2})$$

dans laquelle $\mathbf{R}(\cdot)$ est la matrice de rotation 2-D (voir A.2).

Le point P est observé dans le plan image sous la colonne

$$u = u_o - \frac{x^C}{z^C} \cdot \frac{f}{h_u}, \quad (\text{F.3})$$

où f est la distance focale, h_u est la largeur d'un pixel suivant l'axe $\mathcal{X}^I$ et u_o est l'index de la colonne du point d'intersection de l'axe optique sur le plan image. De cette relation, nous pouvons

seulement déterminer à partir de u la ligne d'observation de P , ses coordonnées étant connues à un facteur d'échelle près.

Maintenant, pour déterminer la transformation $g(\cdot)$, il nous faut relier $\mathbf{p}^L$ à l'angle d'observation ϕ de P dans le repère télémètre. Cette relation est paramétrable par la distance d du point P à l'origine du repère télémètre :

$$\mathbf{p}^L = d \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}. \quad (\text{F.4})$$

A partir des équations (F.2), (F.3) et (F.4), nous pouvons dériver la forme analytique de $\phi = g(u)$. Cependant, nous ne la donnons pas ici car elle n'a pas d'intérêt pour cette analyse. Nous nous contentons simplement de constater que $g(\cdot)$ dépend implicitement de la distance d du point observé P par rapport au capteur télémétrique. Ceci dit, pour calculer l'angle d'observation ϕ nous devons connaître non seulement u , mais aussi la distance d de l'obstacle générateur d'une droite verticale. Bien sûr, d peut être estimée à partir des mesures télémétriques.

Nous pouvons vérifier aussi bien que $g(\cdot)$ dépend des paramètres de positionnement intercapteur $(x_C^L, y_C^L, \theta_C^L)$ et des paramètres intrinsèques f , h_u , et u_o de la caméra. Ainsi, à première vue, l'étalonnage du système multicapteur passe par une procédure d'estimation de ces paramètres. Une telle procédure peut être réalisée de deux façons :

- *Par estimation directe*, à partir de mesures $\{\phi_m, u\}$ effectuées à l'aide de balises facilement identifiables ;
- *Par estimation directe découplée*, où les paramètres intrinsèques de la caméra sont d'abord estimés en utilisant des procédures classiques de calibrage [229]. En suite $(x_C^L, y_C^L, \theta_C^L)$ sont déterminés à partir des mesures $\{\phi_m, u\}$, comme pour la solution précédente, mais avec un nombre moins important de paramètres, ce qui facilite l'estimation.

Pour ces solutions, le problème d'estimation peut être résolu par une procédure itérative de minimisation d'un critère de moindres-carrés non-linéaires.

F.2 Simulation

Avant de proposer une procédure d'étalonnage, nous proposons d'évaluer $g(\cdot)$ par le biais de simulations. Ces simulations portent sur une caméra ayant comme paramètres $f = 12,5 \text{ mm}$, $h_u = 8/640 \text{ mm}$ et $u_o = 150$. Ces paramètres sont des approximations empiriques correspondant à une caméra construite avec un capteur CCD de 8 mm de largeur, 640 colonnes, et une optique de focale $f = 12,5 \text{ mm}$. u_o a été choisi au hasard. Pour cette caméra, nous avons simulé trois configurations pour le positionnement relatif :

- I. Cas de figure A : $x_C^L = 1 \text{ cm}$, $y_C^L = 0,6 \text{ cm}$ et $\theta_C^L = 2^\circ$;
- II. Cas de figure B : $x_C^L = 5 \text{ cm}$, $y_C^L = 3 \text{ cm}$ et $\theta_C^L = 5^\circ$;
- III. Cas de figure C : $x_C^L = 10 \text{ cm}$, $y_C^L = 6 \text{ cm}$ et $\theta_C^L = 5^\circ$.

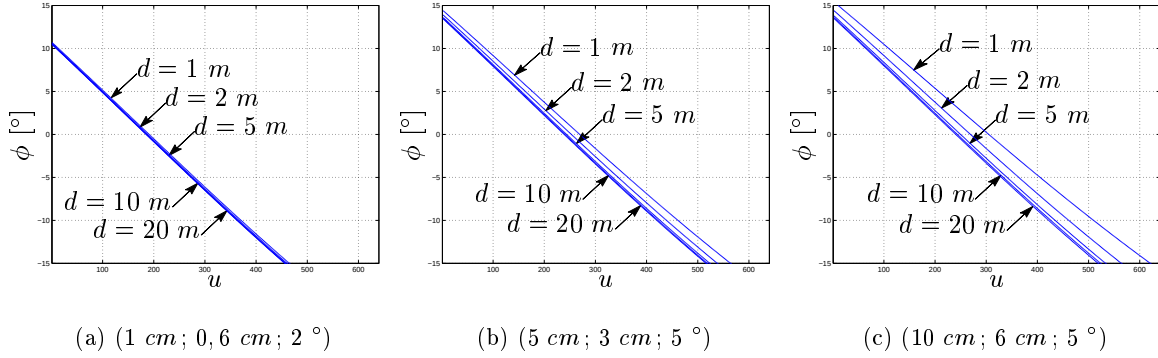


FIG. F.2 – Résultats de simulation de la transformation $\phi = g(u)$ pour différentes configurations $(x_C^L, y_C^L, \theta_C^L)$ du positionnement inter-capteurs.

Nous remarquons que ces cas de figure présentent différents degrés d'écarts de positionnement des capteurs, A étant le cas le plus précis. Nous observons aussi que l'écart y_C^L est toujours moins important que x_C^L . Ce choix s'explique par le fait que, étant donné la configuration du système multicapteur, il est plus facile faire coïncider de façon manuelle les axes $\mathcal{X}^L$ et $\mathcal{Z}^C$ que les axes $\mathcal{Y}^L$ et $\mathcal{X}^C$. Les résultats de simulation pour les cas de figure A, B et C sont montrés sous la forme de courbes $u \times \phi$ sur les Figures F.2(a)-(c). Pour chaque cas de figure, nous présentons des courbes de $g(\cdot)$ pour différentes valeurs de d .

D'après la Figure F.2(a), nous observons que pour le cas A la transformation $g(\cdot)$ ne change pas tellement en fonction de la distance d . En conséquence, des erreurs sur la mesure ou sur l'estimation de d sont moins conséquentes sur ϕ . Nous pouvons même négliger d , et modéliser l'incertitude sur ϕ de façon à prendre en compte l'erreur de modélisation. Pour les cas de figure B et C, nous pouvons voir qu'un écart majeur sur le positionnement relatif des capteurs a une influence directe dans la fonction $g(\cdot)$. En plus, $g(\cdot)$ varie largement pour des obstacles plus proches du capteur, comme nous pouvons le voir pour le cas C avec $d = 1\text{ m}$, $d = 2\text{ m}$ et $d = 5\text{ m}$.

Nous concluons avec ces simulations que moins important sera l'écart de positionnement relatif entre les capteurs, moins importante sera l'incertitude de ϕ par rapport à la distance d de l'obstacle. En plus, ces résultats nous permettent proposer une nouvelle procédure d'étalonnage du système multicapteur :

- *Par estimation indirecte*, où un polynôme serait utilisé comme une approximation de la relation $\phi = g(u)$ pour les cas où l'écart de positionnement caméra-télémetre n'est pas important. Ainsi, nous n'avons pas besoin d'estimer directement x_C^L , y_C^L , θ_C^L , f , h_u , et u_o .

F.3 Procédure d'étalonnage

Avant de décider de la procédure la plus adéquate pour estimer la transformation $g(\cdot)$, nous avons mené une expérimentation pour acquérir des paires de mesure $\{\phi_m, u\}$. Cette expérimentation a consisté à utiliser des balises artificielles pour représenter des obstacles, et à acquérir des

paires d'images $(\mathcal{L}, \mathcal{I})$ avec le robot dans différentes positions par rapport aux balises. Ainsi, nous avons choisi 50 positions d'observation de façon à obtenir des mesures $\{\phi_m, u\}$ couvrant au maximum le champ du système multicapteur. Cependant, certaines positions ont donné des mesures non fiables qui n'ont pas été utilisées. Les mesures ont été déterminées manuellement à partir de chaque paire d'images $(\mathcal{L}, \mathcal{I})$, comme illustré sur la Figure F.3. Avec cette figure, nous pouvons voir sur l'image vidéo que les balises sont en fait des supports en bois avec un marquage noir sur les bords. Les balises sont facilement identifiables sur l'image télémétrique. Nous avons cherché les colonnes u de l'image vidéo correspondantes à ces bords. Dans les image télémétriques, les angles d'observation mesurés ϕ_m sont calculés à partir des lignes d'observation de ces mêmes bords.

De l'ensemble d'images capturées, nous avons extrait 152 mesures valides $\{\phi_m, u\}$. Les distances d par rapport aux balises étaient distribuées à l'intérieur de l'intervalle $0,5 \text{ m} \leq d \leq 10,5 \text{ m}$. Ces mesures sont affichées sous la forme de points sur le graphique de la Figure F.4(a). Nous pouvons bien vérifier que ces données indiquent une configuration géométrique semblable à celle de la simulation de la Figure F.2(a). Ceci permet de simplifier l'étalonnage du système en utilisant une approche indirecte, sans qu'il soit nécessaire de développer un outil mathématique plus complexe. Ainsi, une procédure d'indentification est effectuée, et le modèle linéaire suivant est obtenu :

$$\hat{g}(u) = \hat{a} \cdot u + \hat{b}, \quad (\text{F.5})$$

avec

$$\hat{a} = -0,00162944206159, \quad \hat{b} = 0,32413516615443. \quad (\text{F.6})$$

La courbe correspondante à $\hat{g}(u)$ est aussi affichée sous la forme d'un tracé continu sur la Figure F.4(a). Les résidus d'estimation en fonction de u sont affichés sur la Figure F.4(b). Ainsi, si nous modélisons la transformation g par

$$g(u) = a \cdot u + b + \varepsilon, \quad (\text{F.7})$$

où $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$, l'écart-type des résidus ε est alors donné par

$$\hat{\sigma}_\varepsilon = 0,412^\circ. \quad (\text{F.8})$$

Nous remarquons que $\hat{\sigma}_\varepsilon$ inclut aussi l'incertitude des mesures u utilisées pour cette expérience. En effet, la composante de $\hat{\sigma}_\varepsilon$ due à u est approximée par $\hat{a}\sqrt{\Lambda_u}$. Comme les mesures u ont été déterminées à partir d'une seule colonne de l'image vidéo, nous avons à partir de l'équation (4.5) : $\hat{a}\sqrt{\Lambda_u} = 0,015^\circ$, ce qui est négligeable par rapport à $\hat{\sigma}_\varepsilon$ (équation (F.8)). Ainsi, nous concluons que la majeure partie de l'incertitude de $\hat{g}(u)$ est due aux paramètres implicites de la géométrie de la caméra, et aussi à l'écart de positionnement relatif entre les capteurs. Pour prendre en compte toutes les sources d'incertitude lors de l'application de la transformation $\hat{g}(u)$, nous proposons que la variance $\Lambda_{\hat{\phi}}$ de l'angle d'observation estimé $\hat{\phi}$ soit donnée par propagation au premier ordre (voir annexe B) :

$$\Lambda_{\hat{\phi}} = \hat{a}^2 \cdot \Lambda_u + \hat{\sigma}_\varepsilon. \quad (\text{F.9})$$

En conclusion, nous remarquons que le montage du système multicapteur a permis de simplifier la procédure d'étalonnage. En plus, et même si les hypothèses sur le parallélisme des axes (Section

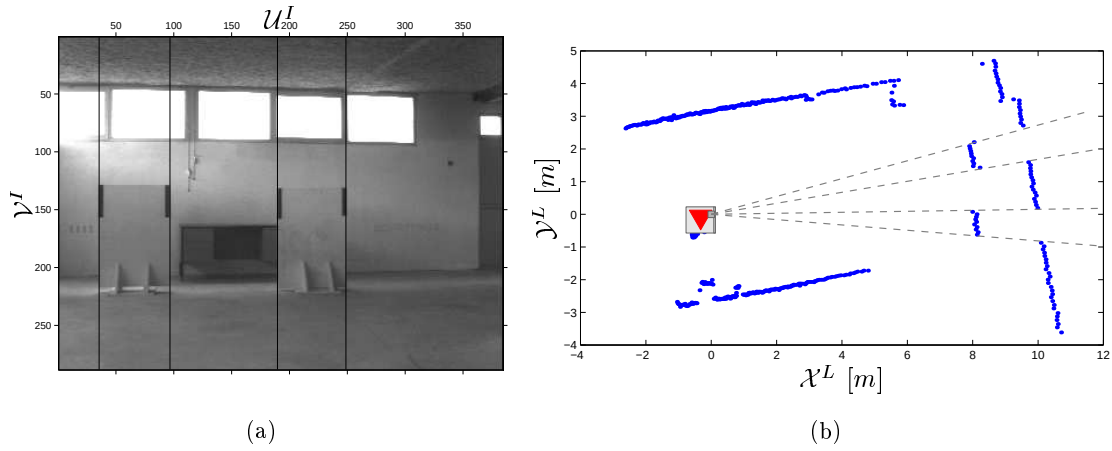


FIG. F.3 – Exemple de paire d'images utilisée pour extraire les mesures $\{\phi_m, u\}$ de la procédure d'étalonnage du système multicapteur. (a) Image vidéo et colonnes correspondantes à u (tracés verticaux). (b) Image télémétrique et directions des mesures ϕ_m (tracés pointillés).

F.1) ne sont pas valides, cela n'affecte en rien les résultats obtenus avec la procédure d'étalonnage par estimation indirecte. En effet, ce dont nous avons besoin c'est de pouvoir relier les observations entre les espaces image vidéo et image télémétrique, et de pouvoir déterminer aussi l'incertitude de cette relation.

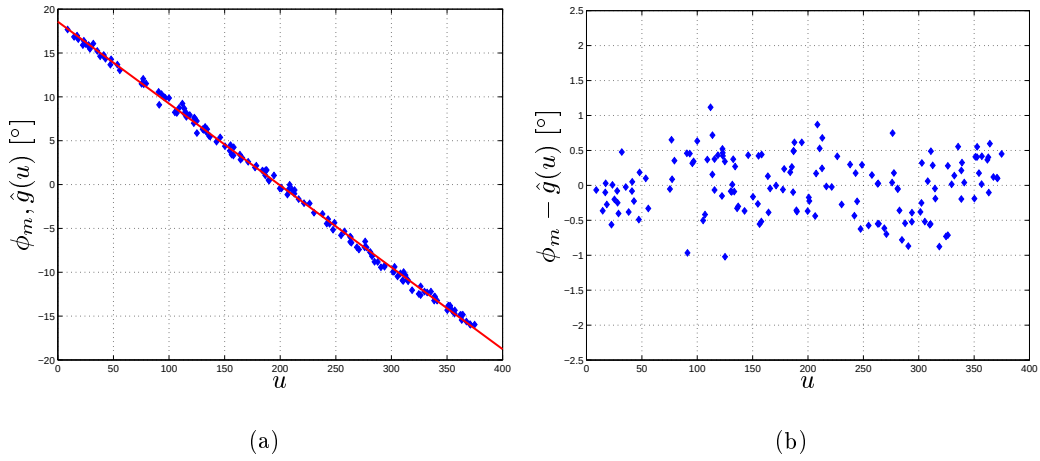


FIG. F.4 – Données d'étalonnage du système multicapteur : (a) les mesures $\phi_m \times u$ et de la transformation estimée $\hat{g}(u)$, et (b) les résidus $\phi_m - \hat{g}(u)$.

Données des expériences

G.1 Introduction

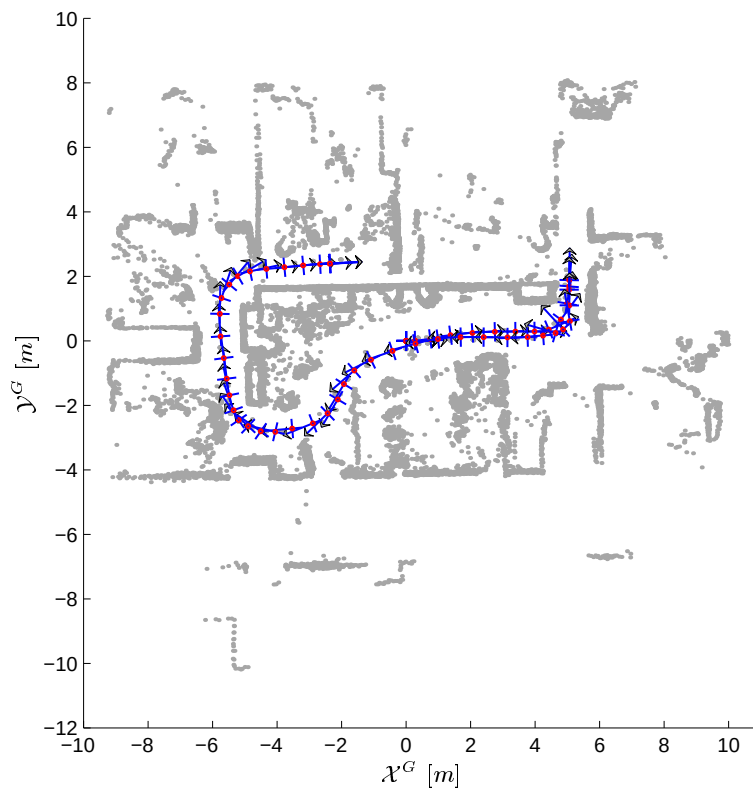
Nous présentons dans cette annexe des données réelles utilisées pour l'évaluation hors-ligne des techniques ici développées. Ces données ont été acquises par le système sensoriel du robot Omni lors d'expériences de navigation effectuées dans certains locaux du LIRMM. Nous présentons aussi, à titre informatif, le nombre d'images télémétriques et vidéo, la durée, la distance parcourue, et les vitesses maximales de translation v_t et de rotation v_θ . Les seules données présentées de façon graphique sont les images télémétriques, ramenées au repère global par gyrodométrie en utilisant des paramètres géométriques mesurés λ_0 (cf. équ. 6.1).

G.2 Halle robotique - A

Données de l'expérience :

Nombre d'images :	60	v_t max. :	38,68 cm/s
Temps d'expérience :	120,5 s	v_θ max. :	16,29 $^\circ/s$
Distance parcourue :	27,67 m		

Images télémétriques et position du robot par gyrométrie :



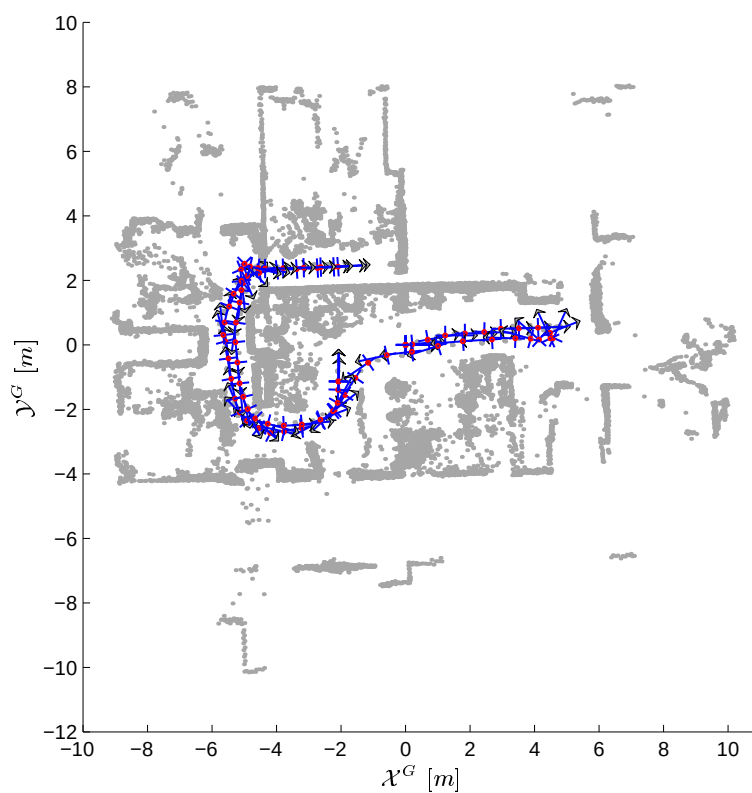
G.3 Halle robotique - B

Données de l'expérience :

Nombre d'images : 79

Temps d'expérience : 158,6 s

Distance parcourue : 34,45 m

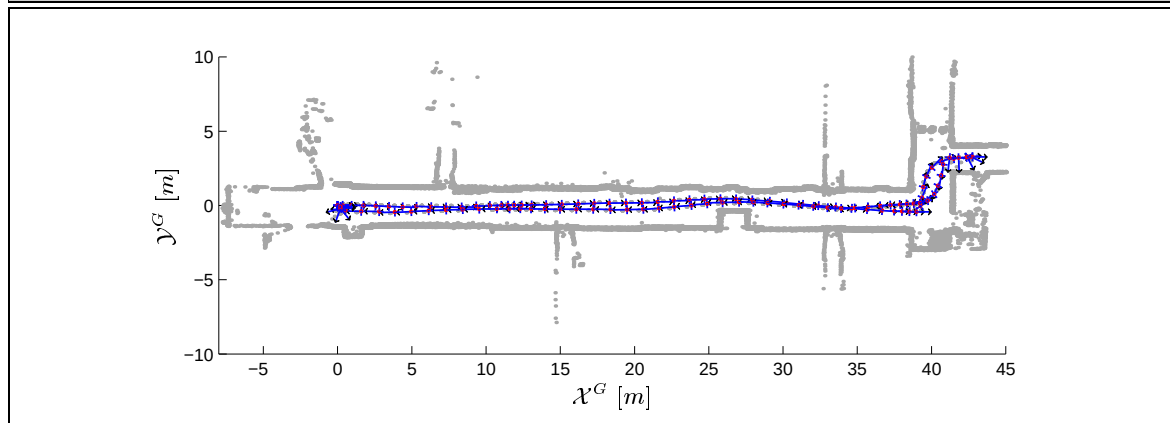
 v_t max. : 43,5 cm/s v_θ max. : 25,90 °/s**Images télémétriques et position du robot par gyrométrie :**

G.4 Couloir robotique - A

Données de l'expérience :

Nombre d'images :	113	v_t max. :	60,74 cm/s
Temps d'expérience :	226,6 s	v_θ max. :	30,30 °/s
Distance parcourue :	90,79 m		

Images télémétriques et position du robot par gyrométrie :

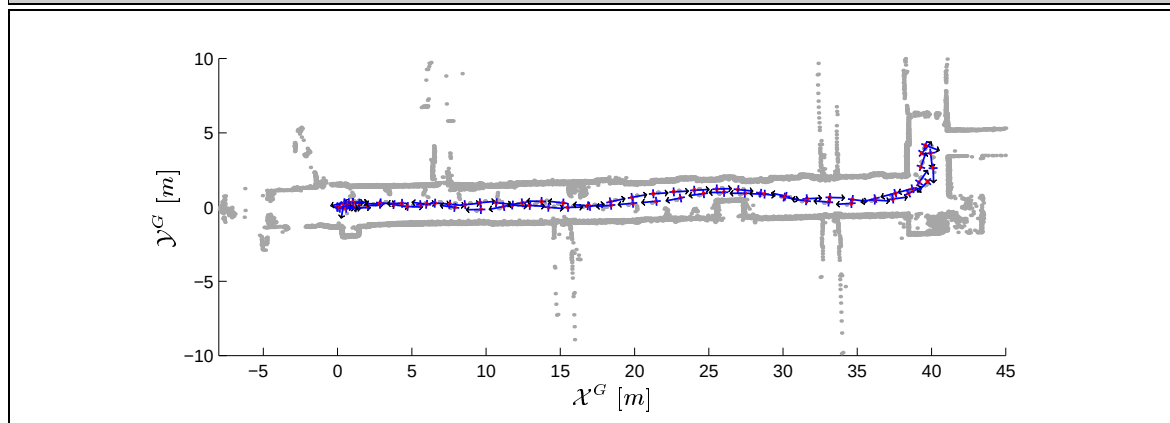


G.5 Couloir robotique - B

Données de l'expérience :

Nombre d'images :	75	v_t max. :	91,83 cm/s
Temps d'expérience :	150,6 s	v_θ max. :	39,4 °/s
Distance parcourue :	85,38 m		

Images télémétriques et position du robot par gyrométrie :

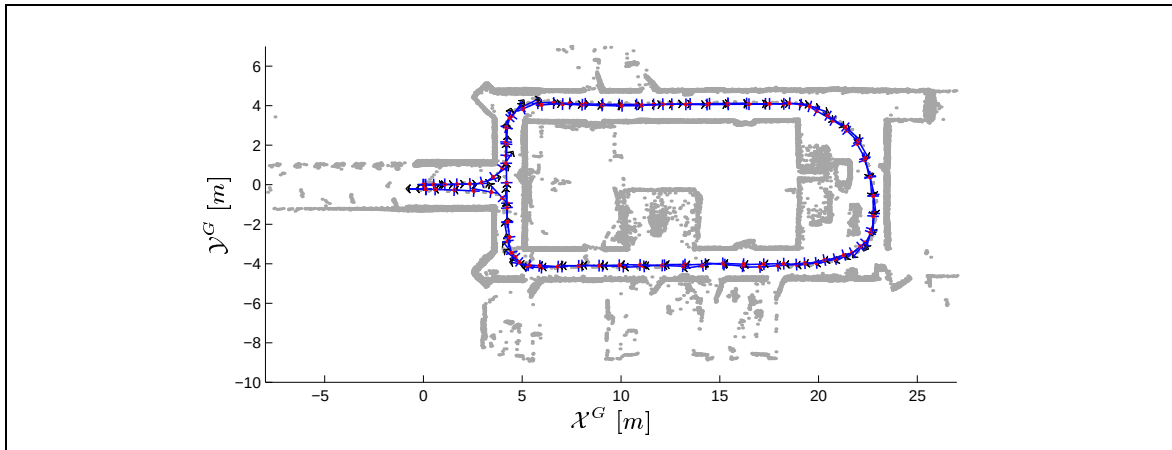


G.6 Premier étage extension - A

Données de l'expérience :

Nombre d'images :	123	v_t max. :	55,33 cm/s
Temps d'expérience :	246,7 s	v_θ max. :	18,42 $^\circ/s$
Distance parcourue :	105,73 m		

Images télémétriques et position du robot par gyrométrie :

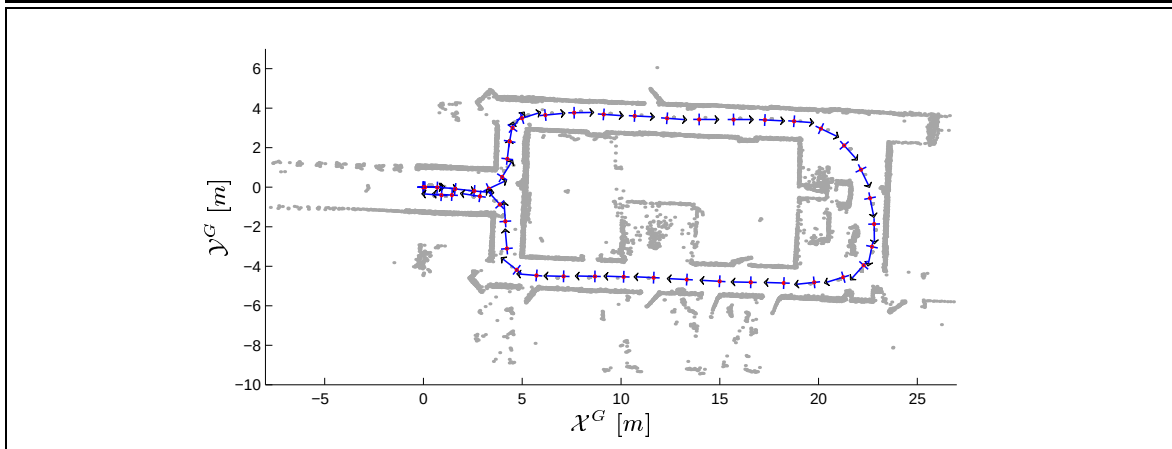


G.7 Premier étage extension - B

Données de l'expérience :

Nombre d'images :	48	v_t max. :	85,94 cm/s
Temps d'expérience :	96,6 s	v_θ max. :	24,61 $^\circ/s$
Distance parcourue :	55,93 m		

Images télémétriques et position du robot par gyrométrie :



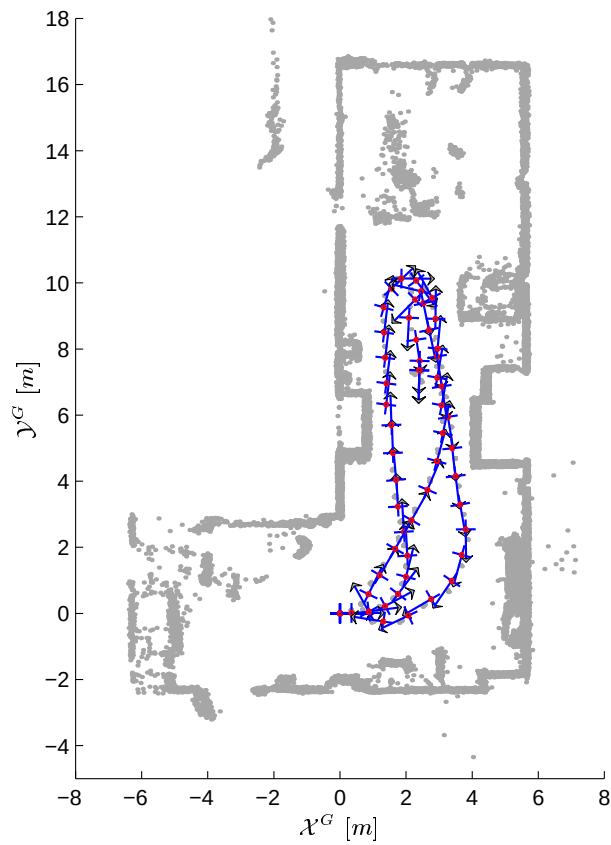
G.8 Rez-de-chaussée extension - A

Données de l'expérience :

Nombre d'images : 60

Temps d'expérience : 120,6 s

Distance parcourue : 37,47 m

 v_t max. : 50,43 cm/s v_θ max. : 30,75 °/s**Images télémétriques et position du robot par gyrométrie :**

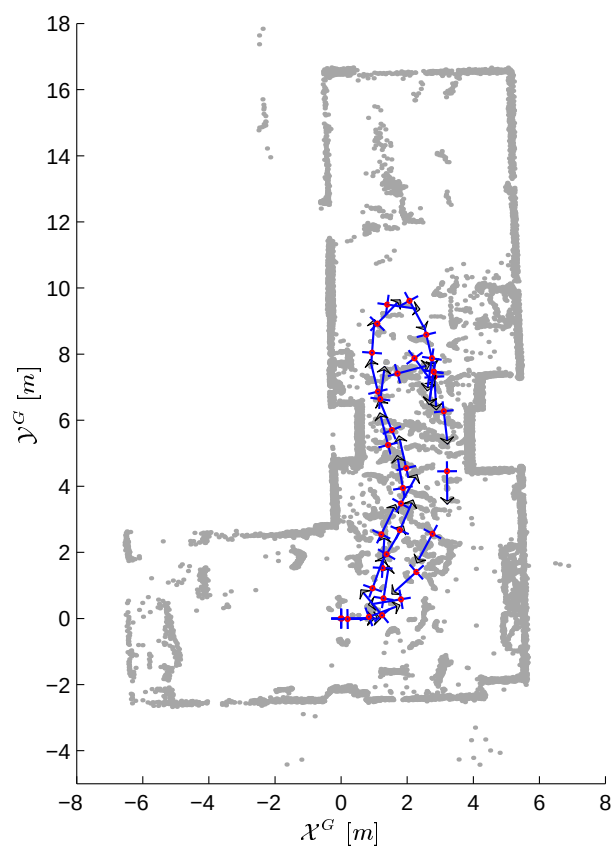
G.9 Rez-de-chaussée extension - B

Données de l'expérience :

Nombre d'images : 35

Temps d'expérience : 70,6 s

Distance parcourue : 31,24 m

 v_t max. : 92,31 cm/s v_θ max. : 35,71 °/s**Images télémétriques et position du robot par gyrométrie :**

G.10 Halle mécatronique

Données de l'expérience :

Nombre d'images : 162

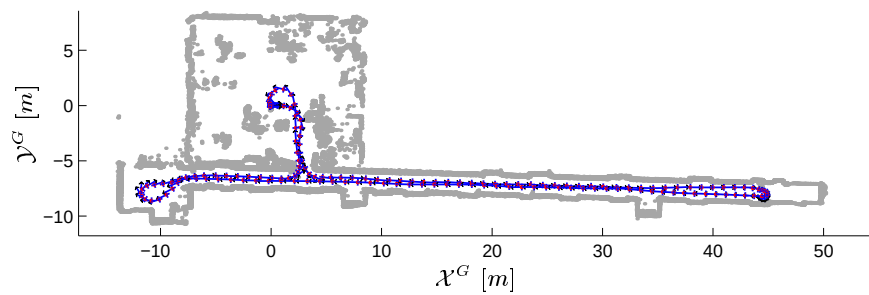
Temps d'expérience : 323,5 s

Distance parcourue : 133,18 m

v_t max. : 58,20 cm/s

v_θ max. : 25,97 °/s

Images télémétriques et position du robot par gyrométrie :



Tableaux de performance des méthodes de localisation

Les tableaux qui suivent présentent des résultats d'estimation des différentes approches de localisation du chapitre 6. Ils contiennent des mesures sur les écarts $\varepsilon_{\hat{x}}$, $\varepsilon_{\hat{y}}$ et $\varepsilon_{\hat{\theta}}$ d'estimation par rapport aux composantes de position x , y et θ fournies par gyrodométrie dans les site *Premier étage extension - B* (annexe G.7) et *Couloir robotique - B* (annexe G.5). Des différentes exécutions ont été effectuées, en fonction de la technique d'estime employée (voir section 6.5 pour plus de détails). Les mesures statistiques sont donc :

- Mesures de biais, données par mes moyennes empiriques non-biaisées $\bar{\varepsilon}_{\hat{x}}$, $\bar{\varepsilon}_{\hat{y}}$ et $\bar{\varepsilon}_{\hat{\theta}}$ respectivement de $\varepsilon_{\hat{x}}$, $\varepsilon_{\hat{y}}$ et $\varepsilon_{\hat{\theta}}$, pour chaque exécution ;
- Mesures d'écartement, données par $\overline{\|\varepsilon_{\hat{x}}\|}$, $\overline{\|\varepsilon_{\hat{y}}\|}$ et $\overline{\|\varepsilon_{\hat{\theta}}\|}$ qui sont respectivement les moyennes des valeurs absolues des séquences $\varepsilon_{\hat{x}}$, $\varepsilon_{\hat{y}}$ et $\varepsilon_{\hat{\theta}}$ pour chaque exécution ;
- Proportion d'estimées par intervalle de confiance 1σ et 3σ , déterminés par les composantes de la diagonale de $\mathbf{\Lambda}_{\hat{\mathbf{z}}_k}$. En supposant une distribution gaussienne des estimées, nous espérons avoir des proportions autour de 68 % à l'intérieur de l'intervalle 1σ , et proches à 100 % pour l'intervalle 3σ . Des proportions sensiblement inférieures à 100 % pour l'intervalle 3σ indiquent optimisme dans l'estimation, et des proportions sensiblement supérieures à 68 % pour l'intervalle 1σ indiquent résultats pessimistes.

Les méthodes de localisation qui ont divergé ont leurs mesures remplacées par “—”. Nous considérons qu'une méthode a divergé si $\overline{\|\varepsilon_{\hat{x}}\|} \geq 10 \text{ cm}$, $\overline{\|\varepsilon_{\hat{y}}\|} \geq 10 \text{ cm}$ ou $\overline{\|\varepsilon_{\hat{\theta}}\|} \geq 5^\circ$. Le temps maximum d'exécution de chaque approche est représenté par $t_{\max}$. Les trois meilleures performances pour les mesures de biais et d'écartement sont écrites en gras.

TAB. H.1 – Mesures statistiques de localisation dans le site *Premier étage extension - B* et estime par odométrie calibrée.

Estimateur	Mesures de biais			Mesures d'écartement			Proportion d'estimées par intervalle de confiance			t_{max}
	$\bar{\varepsilon}_{\hat{x}}$	$\bar{\varepsilon}_{\hat{y}}$	$\bar{\varepsilon}_{\hat{\theta}}$	$\ \varepsilon_{\hat{x}}\ $	$\ \varepsilon_{\hat{y}}\ $	$\ \varepsilon_{\hat{\theta}}\ $	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{x}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{y}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{\theta}}$	
<i>FKE</i> parallèle	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel	-0,23cm	1,12cm	-0,05°	2,14cm	1,41cm	0,39°	100,0%(100,0%)	97,8%(97,8%)	41,3%(80,4%)	11,0ms
<i>FKEI</i> séquentiel	0,77cm	5,90cm	0,11°	5,58cm	6,39cm	1,39°	100,0%(100,0%)	89,1%(89,1%)	71,7%(87,0%)	23,3ms
<i>FKE</i> parallèle (rep. unifiée)	-0,27cm	0,93cm	-0,08°	2,11cm	1,59cm	0,38°	100,0%(100,0%)	95,7%(95,7%)	60,9%(91,3%)	14,0ms
<i>FKE</i> séquentiel (rep. unifiée)	-0,26cm	0,95cm	-0,08°	2,11cm	1,60cm	0,38°	100,0%(100,0%)	95,7%(95,7%)	58,7%(91,3%)	8,9ms
<i>FKEI</i> séquentiel (rep. unifiée)	1,12cm	6,31cm	0,17°	5,73cm	6,81cm	1,51°	95,7%(95,7%)	89,1%(89,1%)	73,9%(89,1%)	32,9ms
<i>MCP</i> incomplet (μ -Mahalanobis)	0,15cm	0,40cm	-0,18°	1,88cm	1,08cm	0,39°	100,0%(100,0%)	100,0%(100,0%)	97,8%(100,0%)	9,1ms
<i>MCP</i> incomplet (μ -Médiane)	0,18cm	0,35cm	-0,01°	1,28cm	0,94cm	0,23°	100,0%(100,0%)	100,0%(100,0%)	100,0%(100,0%)	9,1ms
M-estimateur (μ -Tukey)	-0,09cm	0,77cm	0,02°	1,58cm	1,19cm	0,26°	100,0%(100,0%)	100,0%(100,0%)	97,8%(100,0%)	45,5ms
M-estimateur (μ -Huber)	0,03cm	0,63cm	-0,04°	1,42cm	1,10cm	0,27°	100,0%(100,0%)	100,0%(100,0%)	97,8%(100,0%)	47,3ms

TAB. H.2 – Mesures statistiques de localisation dans le site *Premier étage extension - B* et estime par odométrie calibrée pessimiste.

Estimateur	Mesures de biais			Mesures d'écartement			Proportion d'estimées par intervalle de confiance			t_{max}
	$\bar{\varepsilon}_{\hat{x}}$	$\bar{\varepsilon}_{\hat{y}}$	$\bar{\varepsilon}_{\hat{\theta}}$	$\ \varepsilon_{\hat{x}}\ $	$\ \varepsilon_{\hat{y}}\ $	$\ \varepsilon_{\hat{\theta}}\ $	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{x}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{y}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{\theta}}$	
<i>FKE</i> parallèle	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel	-3,19cm	2,39cm	-0,08°	4,47cm	3,08cm	0,53°	100,0%(100,0%)	76,1%(76,1%)	34,8%(67,4%)	8,8ms
<i>FKEI</i> séquentiel	-1,35cm	3,59cm	-0,08°	7,47cm	6,09cm	2,04°	100,0%(100,0%)	89,1%(89,1%)	67,4%(84,8%)	24,6ms
<i>FKE</i> parallèle (rep. unifiée)	-1,05cm	0,55cm	0,04°	2,50cm	2,68cm	0,60°	100,0%(100,0%)	87,0%(87,0%)	52,2%(84,8%)	18,6ms
<i>FKE</i> séquentiel (rep. unifiée)	-0,96cm	0,28cm	0,07°	2,44cm	2,62cm	0,56°	100,0%(100,0%)	89,1%(89,1%)	50,0%(87,0%)	9,0ms
<i>FKEI</i> séquentiel (rep. unifiée)	-3,18cm	2,17cm	-0,26°	7,40cm	4,97cm	1,86°	100,0%(100,0%)	95,7%(95,7%)	71,7%(82,6%)	32,9ms
<i>MCP</i> incomplet (μ -Mahalanobis)	-0,53cm	1,23cm	-0,15°	2,27cm	1,51cm	0,50°	100,0%(100,0%)	100,0%(100,0%)	95,6%(100,0%)	9,3ms
<i>MCP</i> incomplet (μ -Médiane)	-0,18cm	0,42cm	-0,02°	1,27cm	1,00cm	0,23°	100,0%(100,0%)	100,0%(100,0%)	100,0%(100,0%)	9,3ms
M-estimateur (μ -Tukey)	-0,34cm	0,77cm	-0,06°	1,65cm	1,11cm	0,28°	100,0%(100,0%)	100,0%(100,0%)	100,0%(100,0%)	51,5ms
M-estimateur (μ -Huber)	-0,38cm	0,71cm	-0,09°	1,54cm	1,19cm	0,33°	100,0%(100,0%)	100,0%(100,0%)	100,0%(100,0%)	51,6ms

TAB. H.3 – Mesures statistiques de localisation dans le site *Premier étage extension - B* et estime par odométrie calibrée optimiste.

Estimateur	Mesures de biais			Mesures d'écartement			Proportion d'estimées par intervalle de confiance			t_{max}
	$\bar{\varepsilon}_{\hat{x}}$	$\bar{\varepsilon}_{\hat{y}}$	$\bar{\varepsilon}_{\hat{\theta}}$	$\ \varepsilon_{\hat{x}}\ $	$\ \varepsilon_{\hat{y}}\ $	$\ \varepsilon_{\hat{\theta}}\ $	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{x}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{y}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{\theta}}$	
<i>FKE</i> parallèle	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel	-0,51cm	0,86cm	-0,03°	1,81cm	1,22cm	0,36°	100,0%(100,0%)	97,8%(97,8%)	43,5%(87,0%)	8,6ms
<i>FKEI</i> séquentiel	-0,13cm	2,42cm	0,08°	2,54cm	3,21cm	0,70°	100,0%(100,0%)	97,8%(97,8%)	80,4%(93,5%)	23,8ms
<i>FKE</i> parallèle (rep. unifiée)	-0,29cm	0,67cm	-0,08°	1,62cm	1,22cm	0,33°	100,0%(100,0%)	95,7%(95,7%)	65,2%(95,7%)	12,6ms
<i>FKE</i> séquentiel (rep. unifiée)	-0,29cm	0,68cm	-0,08°	1,62cm	1,22cm	0,33°	100,0%(100,0%)	95,7%(95,7%)	65,2%(95,7%)	8,8ms
<i>FKEI</i> séquentiel (rep. unifiée)	-0,28cm	2,26cm	0,16°	2,61cm	3,08cm	0,73°	100,0%(100,0%)	97,8%(97,8%)	82,6%(93,5%)	33,4ms
<i>MCP</i> incomplet (μ -Mahalanobis)	0,04cm	0,27cm	-0,10°	1,27cm	1,00cm	0,32°	100,0%(100,0%)	100,0%(100,0%)	95,7%(100,0%)	9,0ms
<i>MCP</i> incomplet (μ -Médiane)	0,05cm	0,39cm	0,04°	0,96cm	0,88cm	0,22°	100,0%(100,0%)	100,0%(100,0%)	97,8%(100,0%)	9,0ms
M-estimateur (μ -Tukey)	0,28cm	0,33cm	-0,03°	1,09cm	0,96cm	0,25°	100,0%(100,0%)	100,0%(100,0%)	100,0%(100,0%)	43,3ms
M-estimateur (μ -Huber)	-0,10cm	0,45cm	0,00°	0,98cm	0,90cm	0,24°	100,0%(100,0%)	100,0%(100,0%)	97,8%(100,0%)	38,2ms

TAB. H.4 – Mesures statistiques de localisation dans le site *Couloir robotique - B* et estime par odométrie calibrée.

Estimateur	Mesures de biais			Mesures d'écartement			Proportion d'estimées par intervalle de confiance			t_{max}
	$\bar{\varepsilon}_{\hat{x}}$	$\bar{\varepsilon}_{\hat{y}}$	$\bar{\varepsilon}_{\hat{\theta}}$	$\ \varepsilon_{\hat{x}}\ $	$\ \varepsilon_{\hat{y}}\ $	$\ \varepsilon_{\hat{\theta}}\ $	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{x}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{y}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{\theta}}$	
<i>FKE</i> parallèle	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel	—	—	—	—	—	—	—	—	—	—
<i>FKEI</i> séquentiel	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> parallèle (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>FKEI</i> séquentiel (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>MCP</i> incomplet (μ -Mahalanobis)	-0,80cm	2,00cm	0,09°	4,60cm	3,12cm	0,53°	100,0%(100,0%)	97,3%(97,3%)	86,3%(98,6%)	23,6ms
<i>MCP</i> incomplet (μ -Médiane)	0,21cm	0,66cm	0,12°	4,02cm	2,84cm	0,48°	100,0%(100,0%)	97,3%(97,3%)	83,6%(100,0%)	14,0ms
M-estimateur (μ -Tukey)	-0,02cm	1,28cm	0,03°	5,05cm	3,32cm	0,54°	100,0%(100,0%)	95,8%(95,8%)	83,3%(100,0%)	62,4ms
M-estimateur (μ -Huber)	1,06cm	0,98cm	0,06°	4,59cm	3,26cm	0,54°	100,0%(100,0%)	95,9%(95,9%)	84,9%(100,0%)	62,9ms

TAB. H.5 – Mesures statistiques de localisation dans le site *Couloir robotique - B* et estime par odométrie calibrée pessimiste.

Estimateur	Mesures de biais			Mesures d'écartement			Proportion d'estimées par intervalle de confiance			t_{max}
	$\bar{\varepsilon}_{\hat{x}}$	$\bar{\varepsilon}_{\hat{y}}$	$\bar{\varepsilon}_{\hat{\theta}}$	$\ \varepsilon_{\hat{x}}\ $	$\ \varepsilon_{\hat{y}}\ $	$\ \varepsilon_{\hat{\theta}}\ $	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{x}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{y}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{\theta}}$	
<i>FKE</i> parallèle	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel	-3,37cm	-4,57cm	0,05°	8,90cm	8,56cm	1,23°	89,0%(89,0%)	63,0%(63,0%)	13,7%(45,2%)	14,6ms
<i>FKEI</i> séquentiel	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> parallèle (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>FKEI</i> séquentiel (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>MCP</i> incomplet (μ -Mahalanobis)	0,09cm	1,58cm	0,13°	4,44cm	3,73cm	0,58°	100,0%(100,0%)	98,6%(98,6%)	84,9%(100,0%)	14,2ms
<i>MCP</i> incomplet (μ -Médiane)	-0,59cm	0,20cm	0,08°	4,48cm	3,24cm	0,53°	100,0%(100,0%)	98,6%(98,6%)	87,7%(100,0%)	14,1ms
M-estimateur (μ -Tukey)	-0,06cm	-1,48cm	-0,24°	4,52cm	4,95cm	0,86°	100,0%(100,0%)	94,4%(94,4%)	74,6%(100,0%)	68,7ms
M-estimateur (μ -Huber)	-0,16cm	-0,70cm	0,03°	4,77cm	4,50cm	0,75°	100,0%(100,0%)	93,2%(93,2%)	82,2%(100,0%)	64,9ms

TAB. H.6 – Mesures statistiques de localisation dans le site *Couloir robotique - B* et estime par odométrie calibrée optimiste.

Estimateur	Mesures de biais			Mesures d'écartement			Proportion d'estimées par intervalle de confiance			t_{max}
	$\bar{\varepsilon}_{\hat{x}}$	$\bar{\varepsilon}_{\hat{y}}$	$\bar{\varepsilon}_{\hat{\theta}}$	$\ \varepsilon_{\hat{x}}\ $	$\ \varepsilon_{\hat{y}}\ $	$\ \varepsilon_{\hat{\theta}}\ $	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{x}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{y}}$	$1\sigma(3\sigma)$ sur $\varepsilon_{\hat{\theta}}$	
<i>FKE</i> parallèle	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel	-0,48cm	6,67cm	-0,41°	3,37cm	9,87cm	1,00°	100,0%(100,0%)	54,8%(54,8%)	27,4%(63,0%)	14,4ms
<i>FKEI</i> séquentiel	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> parallèle (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>FKE</i> séquentiel (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>FKEI</i> séquentiel (rep. unifiée)	—	—	—	—	—	—	—	—	—	—
<i>MCP</i> incomplet (μ -Mahalanobis)	0,48cm	5,49cm	-0,03°	2,14cm	5,79cm	0,50°	100,0%(100,0%)	95,5%(95,5%)	81,8%(100,0%)	14,0ms
<i>MCP</i> incomplet (μ -Médiane)	-0,09cm	3,70cm	-0,05°	2,91cm	4,35cm	0,50°	100,0%(100,0%)	95,5%(95,5%)	78,8%(100,0%)	14,0ms
M-estimateur (μ -Tukey)	0,10cm	1,91cm	-0,04°	3,58cm	3,34cm	0,56°	100,0%(100,0%)	95,2%(95,2%)	82,5%(98,4%)	58,3ms
M-estimateur (μ -Huber)	0,29cm	3,96cm	-0,07°	3,26cm	4,60cm	0,55°	100,0%(100,0%)	92,4%(92,4%)	78,8%(98,5%)	57,4ms

Notations

Scalars, vectors and matrices

- I. Les scalaires sont représentés par des lettres minuscules en italique. Par exemple, x .
- II. Les vecteurs sont représentés par des lettres minuscules en gras. Par exemple, $\mathbf{x}$.
 - La forme explicite $\mathbf{x} = \{x_i\}$ est parfois employée pour faciliter les références à l' i -ième composante x_i , $i = 1, 2, \dots$, dans le texte.
 - Par convention, nous utilisons des vecteurs colonne.
 - Certains vecteurs importants :
 - Le vecteur $\beta(\phi) = \begin{pmatrix} \cos(\phi) & \sin(\phi) \end{pmatrix}^T$.
 - Les paramètres de position $\mathbf{z} = (x, y, \theta)^T$.
- III. Les matrices sont représentées par des lettres majuscules en gras : Par exemple, $\mathbf{X}$.
 - La forme explicite $\mathbf{X} = \{x_{ij}\}$ est parfois employée pour faciliter les références à la composante x_{ij} qui se trouve dans l' i -ième ligne et la j -ième colonne de la matrice.
 - Certaines matrices et opérateurs importantes :
 - La matrice identité d'ordre n : $\mathbf{I}_n$.
 - La matrice diagonale d'ordre n est représentée par
$$\mathbf{Diag}(x_1, \dots, x_n).$$
dont $x_1, \dots, x_n$ sont les éléments de la diagonale principale.
 - La trace d'une matrice carrée $\mathbf{X}$ est représentée par
$$\mathbf{Tr}(\mathbf{X}).$$
 - Le déterminant d'une matrice carrée $\mathbf{X}$ est représenté par
$$\det(\mathbf{X}).$$

– La matrice de rotation 2-D :

$$\mathbf{R}(\phi) = \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}.$$

– Les matrices de covariance utilisent en générale le symbole $\mathbf{\Lambda}$.

IV. Dérivés partielles : soit a un scalaire, $\mathbf{x} = \{x_i\}$ et $\mathbf{y} = \{y_i\}$ des vecteurs de dimension n_x et n_y , respectivement, et $\mathbf{W} = \{w_{ij}\}$ une matrice de dimension $n_w \times m_w$. L'opérateur ∇ est utilisé pour représenter les dérivées partielles. Par exemple,

$$\frac{\partial \mathbf{x}}{\partial a} = \nabla_a \mathbf{x}, \quad \frac{\partial \mathbf{x}}{\partial \mathbf{y}} = \nabla_{\mathbf{y}} \mathbf{x}, \quad \frac{\partial \mathbf{W}}{\partial a} = \nabla_a \mathbf{W},$$

avec

$$\nabla_a \mathbf{x} = \begin{pmatrix} \frac{\partial x_1}{\partial a} \\ \frac{\partial x_2}{\partial a} \\ \vdots \\ \frac{\partial x_{n_x}}{\partial a} \end{pmatrix}, \quad \nabla_{\mathbf{y}} \mathbf{x} = \begin{pmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} & \dots & \frac{\partial x_1}{\partial y_{n_y}} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} & \dots & \frac{\partial x_2}{\partial y_{n_y}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_{n_x}}{\partial y_1} & \frac{\partial x_{n_x}}{\partial y_2} & \dots & \frac{\partial x_{n_x}}{\partial y_{n_y}} \end{pmatrix},$$

et $\nabla_a \mathbf{W} = \{\frac{\partial w_{ij}}{\partial a}\}$.

• Transposition : $\nabla_{\mathbf{y}}^T \mathbf{x} = (\nabla_{\mathbf{y}} \mathbf{x})^T$

Données capteur

I. Les images :

- Image vidéo : $\mathcal{I}$.
- Image télémétrique : $\mathcal{L}$.

Incertitude

I. La distribution normale de moyenne $\bar{\mathbf{x}}$ et covariance $\mathbf{\Lambda}_{\bar{\mathbf{x}}}$ est représentée par $\mathcal{N}(\bar{\mathbf{x}}, \mathbf{\Lambda}_{\bar{\mathbf{x}}})$;

II. Une variable incertaine est représentée par $X = \{\mathbf{x}, \mathbf{\Lambda}_{\mathbf{x}}\}$, où $\mathbf{x}$ est le vecteur de paramètres descriptifs et $\mathbf{\Lambda}_{\mathbf{x}}$ est sa matrice de covariance. En générale, $X \sim \mathcal{N}(\mathbf{x}, \mathbf{\Lambda}_{\mathbf{x}})$;

III. La distance de Mahalanobis entre deux variables incertaines X et Y est représentée par :

$$D_{\chi^2}(X, Y).$$

IV. Le seuil de pourcentage p sur la distance de Mahalanobis suivant la loi du χ^2 avec n degrés de liberté :

$$Q_n(p) = \chi_n^2(1 - p)$$

Structures géométriques, cartographiques et listes

I. Primitives géométriques :

- Une droite : $L = \{\mathbf{l}, \mathbf{\Lambda}_l\}$.
- Un point : $P = \{\mathbf{p}, \mathbf{\Lambda}_p\}$.
- Un segment de droite : $S = \{L, \mathbf{p}_a, \mathbf{p}_b\}$ (composition d'une droite infinie L et deux points des extrémités $\mathbf{p}_a$ et $\mathbf{p}_b$).
- $U = \{u, \mathbf{\Lambda}_u\}$ les coordonnées d'une droite verticale dans le plan image.

II. Structures cartographiques (voir Figure D.3 pour leur représentation graphique) :

- Demi-plan : $D = \{S, V\}$ (composition d'un segment de droite S et d'une forme paramétrique de la zone de visibilité $V = \{a, b, c\}$).
- Arête : $A = \{P, \kappa^A, n_1^A, n_2^A\}$ (composition d'un point P , le sous type exprimé par κ^A et les index n_1^A et n_2^A des demi-plans associés).

III. Listes :

- $\Omega^S = \{S_k \mid k = 1, \dots, N_{\Omega^S}\}$ est une liste de segments de droite extraites de la segmentation d'images télémétriques.
- $\Omega^U = \{U_k \mid k = 1, \dots, N_{\Omega^U}\}$ est une liste de coordonnées des droites verticales extraites de la segmentation d'images.

IV. Correspondances :

- $\mathcal{C}^L : \{C_n^L = \{\mathbf{l}_n^G, \mathbf{l}_n^R\} \mid n = 1, \dots, N_L\}$ les paramètres des primitives de type droite en correspondance entre une carte globale $\mathcal{M}^G$ et une carte locale $\mathcal{M}^R$.
- $\mathcal{C}^P : \{C_n^P = \{\mathbf{p}_n^G, \mathbf{p}_n^R\} \mid n = 1, \dots, N_P\}$ les paramètres des primitives de type point en correspondance entre une carte globale $\mathcal{M}^G$ et une carte locale $\mathcal{M}^R$.
- $\mathcal{U} : \{U_n = \{\mathbf{u}_n^G, \mathbf{u}_n^R\} \mid n = 1, \dots, N_U\}$ les paramètres en représentation unifiée des primitives en correspondance entre une carte globale $\mathcal{M}^G$ et une carte locale $\mathcal{M}^R$.

Repères et cartes

I. Les repères :

- Repère 2-D global : $\mathcal{R}^G : \mathcal{X}^G \times \mathcal{Y}^G$.
- Repère 2-D local attaché au robot : $\mathcal{R}^R : \mathcal{X}^R \times \mathcal{Y}^R$.
- Repère 3-D caméra : $\mathcal{R}^C : \mathcal{X}^C \times \mathcal{Y}^C \times \mathcal{Z}^C$.
- Repère 3-D télémètre laser : $\mathcal{R}^L : \mathcal{X}^L \times \mathcal{Y}^L \times \mathcal{Z}^L$.

II. Les cartes :

- Carte globale : $\mathcal{M}^G$.
- Carte locale attachée au repère robot : $\mathcal{M}^R$.

Opérateurs

I. Changement inverse de repère (du repère local vers le repère global), étant donné la position Z (cf. section D.2) :

$$X^G = \mathcal{T}^{-1}(X^R, Z).$$

II. Correspondances (cf. section D.3):

- Entre structures X_1 et X_2 dans le même repère :

$$X_1 \longleftrightarrow X_2.$$

- Entre structures X^G et X^R représentées respectivement dans les repères $\mathcal{R}^G$ et $\mathcal{R}^R$ reliés par la position Z :

$$X^G \longleftrightarrow \mathcal{T}^{-1}(X^R, Z),$$

ou encore

$$X^G \xleftrightarrow{Z} X^R.$$

Abréviations

Abréviation	Signification	Référence principale
<i>CLC</i>	Cartographie et localisation concurrentes	section 7.1, page 127.
<i>FK</i>	filtre de Kalman	annexe E.3.1, page 204.
<i>FKE</i>	filtre de Kalman étendu	annexe E.4, page 210.
<i>LT</i>	<i>Line Tracking</i>	section 4.3.4.2, page 57
<i>MCP</i>	Moindres carrés pondérés	section 6.3.4.5, page 110
<i>IEPF</i>	<i>Iterative end-point fit</i>	section 4.3.4.3, page 58
<i>SMF</i>	<i>Split-and-Merge Fuzzy</i>	section 4.3.4.5, page 60
v.a.	variable aléatoire	annexe B.2, page 177
<i>RTX</i>	Real-Time eXtension	section 1.3, page 3.

Index

- arêtes photométriques, 195
 - contre-exemple (cartographie locale), 86
 - correspondance, 197
 - détermination par bootstrap, 81
 - fusion, 130
 - inclusion (cartographie globale), 138
 - mise à jour (cartographie globale), 138
- bords, 195
 - correspondance, 197
 - détermination (cartographie locale), 75
 - fusion, 130
 - inclusion (cartographie globale), 138
 - mise à jour (cartographie globale), 138
 - mise à jour (cartographie locale), 80
- capteurs
 - caméra vidéo, 4
 - codeurs absolus, 4
 - codeurs incrémentaux, 4
 - extéroceptifs, 2, 4
 - goniomètre laser, 4
 - gyromètre laser, 4
 - proprioceptifs, 2, 3
 - télémètre laser, 4
- cartes d'environnement, 198
 - mise en correspondance, 200
- cartographie, 3
 - globale, 136
 - élimination de structures, 142
 - classification des structures, 137
 - conservation de l'espace libre, 141
 - fusion des observations, 138
 - fusion des ressemblances, 142
 - inclusion de nouvelles structures, 137
 - inconsistance, 147
 - locale, 74
 - fusion avec les indices vidéo, 79
 - traitement d'indices télémétriques, 75
- cohérence
 - de la carte globale, 3
 - de la carte locale, 3
- coins, 195
 - correspondance, 197
 - détermination (cartographie locale), 78
 - fusion, 130
 - inclusion (cartographie globale), 138
 - mise à jour (cartographie globale), 138
 - mise à jour (cartographie locale), 80
- consistance, 202
 - d'estimation, 202
 - d'estimation (définition), 202
 - de position, 2
 - stochastique, 6
- contraintes déterministes, 132
- contraintes stochastiques, 132
 - propagation, 132
- correspondance, 2
 - notation, 198

- demi-plans, 193
 - correspondance, 196
 - détermination (cartographie locale), 75
 - fusion, 129
 - inclusion (cartographie globale), 137
 - mise à jour (cartographie globale), 138
- distance de Mahalanobis, 185
- filtrage stochastique
 - de systèmes linéaires, 203
 - de systèmes non-linéaires, 210
 - filtre de Kalman, 204
 - filtre de Kalman étendu, 210
 - limitations, 211
 - filtre de Kalman étendu itératif, 211
 - limitations, 211
 - filtre intersection de covariances, 205
 - filtre intersection de covariances modifié, 208
- filtre de Canny, 41
- filtre de Kalman, 204
 - théorème sur la consistance, 205
- filtre de Kalman étendu (FKE), 210
- filtre de Kalman étendu itératif (FKEI), 211
- filtre intersection de covariances, 205
 - optimisation du déterminant, 207
 - théorème sur la consistance, 206
- filtre intersection de covariances modifié, 208
- fusion de structures, 128
 - sous contraintes, 132, 133
 - modélisation, 134
- gyrodométrie, 4, 94
 - propagation de l'erreur, 97
- images
 - acquisition, 25
 - critères de cohérence, 24
 - datation, 25
 - prétraitement, 28
 - pseudo-synchronisation, 34
 - télémétrie, 23, 45
 - caractérisation, 29
 - compensation de la déformation, 33
 - correction du biais, 33
 - vidéo, 22
- incertitude
 - paramétrique, 181
 - propagation, 181
 - échantillonnage sélectif, 184
 - par linéarisation, 181
- inconsistance
 - d'estimation (définition), 202
- liste d'indices du télémètre, 74
- liste d'indices vidéo, 74
- localisation, 2, 91
 - estime, 2, 92
 - recalage, 2, 98
 - analyse de complexité, 112
 - approche FKE parallèle, 100
 - approche FKE séquentiel, 101
 - approche FKEI séquentiel, 101
 - approche MCP, 104
 - approche MCP incomplet, 108
 - approches M-estimateur, 111
 - approches MCP incomplet, 110
 - difficultés, 99
 - limitation des approches stochastiques, 101
- mise en correspondance, 2
- modèle cinématique, 92
 - identification, 95
 - paramètres identifiés, 96
 - paramètres mesurés, 92
- odométrie, 4, 94
 - propagation de l'erreur, 97
- Omni, 3
- optimisme, 202
- pessimisme, 202
- primitives géométriques
 - changement de repère, 188
 - droites, 187
 - intersection, 190
 - points, 187

- angle de vue, 190
- représentation unifiée, 103
- segments de droite, 188
- processus gaussiens, 177
 - ellipse de covariance, 180
- repères, 173
 - relations, 175
- RTX, 3
- segmentation
 - d'une image télémétrique, 45
 - détecteur adaptatif, 48
 - détecteur FKE, 50
 - détection de points de cassure, 47
 - détection de points de cassure forts, 52
 - extraction de segments, 53
 - indices d'intérêt, 45
 - points de cassure (définition), 46
 - d'une image télémétrique (noyaux), 54
 - C-moyennes prototypées, 58
 - IEPF, 58
 - LT, 57
 - SMF, 60
 - d'une image vidéo, 40
 - calcul de l'incertitude, 45
 - contours verticaux, 41
 - droites verticales, 42
- structures cartographiques, 193
 - élimination (cartographie globale), 142
 - changement de repère, 195
 - correspondance, 196
 - mise à jour (cartographie globale), 138
 - modélisation des contraintes, 134
- système multicapteur, 22
 - étalonnage, 215, 218
 - analyse géométrique, 215

RESUME :

Ce travail constitue une étude approfondie sur la conception d'un système de cartographie et de localisation de robots mobiles pour des milieux d'intérieur. Divers aspects sont étudiés, tels que l'acquisition de données de capteurs asynchrones, la segmentation d'images, la fusion de données pour la cartographie locale, la localisation par mise en correspondance de cartes géométriques et la mise à jour des cartes d'environnement. Des méthodes originales de segmentation d'images télémétriques, de localisation et de cartographie sont proposées. Pour extraire des segments de droite dans les images télémétriques, nous utilisons un algorithme dérivé des c-moyennes floues. Une approche d'estimation de la position du robot basée sur les statistiques robustes a été mise en œuvre. Une première version de cette approche présentait certaines limitations liées à l'incomplétude de l'espace des solutions. Ceci a été résolu à l'aide de la décomposition en valeurs singulières et du filtrage de Kalman linéaire. Par ailleurs, nous avons développé une méthode découplée de cartographie stochastique qui emploie des contraintes pour limiter les risques de dérive d'estimation. Cela a été possible à partir d'une étude théorique rigoureuse du filtre de Kalman et de ses variantes. Le problème de cartographie est formulé différemment par rapport aux techniques actuelles, dans le but de respecter des conditions théoriques nécessaires à une estimation correcte de la carte d'environnement, ainsi que des mesures d'incertitude qui lui sont associées. Les différents algorithmes de localisation et de cartographie ont été validés expérimentalement et comparés à d'autres solutions sur une plate-forme mobile équipée de deux capteurs extéroceptifs : une caméra vidéo et un télémètre laser.

TITRE (en anglais) :

Environment mapping and robust localization for mobile robots navigation.

RESUME (en anglais) :

This work presents a deep investigation on the conception of a mobile robot mapping and localization system for indoor environments. It includes different studies about non-synchronous sensor data acquisition, image segmentation, data fusion for local environment mapping, pose estimation by using geometrical map matching, and environment maps updating. In this context, we propose original approaches for 2-D range images segmentation, mobile robot localization and environment mapping. For range images segmentation, an algorithm derived from the fuzzy C-means is proposed for reliable line extraction. A new robust estimator using nonlinear optimization and robust statistics is proposed for mobile robot pose estimation. In a first version, this approach presented some limitations, allowing only incomplete solutions. This has been solved using singular value decomposition and linear Kalman filtering. Further, a decoupled approach for stochastic environment mapping using constraints to reduce the divergence risk has been developed. This has been achieved from a rigorous theoretical study on the Kalman filter and its variants. The environment mapping problem is formulated in a different manner with respect to the current techniques. It aims at respecting the theoretical necessary conditions for consistent map building. The proposed solutions have been validated experimentally and compared against other methods on a mobile platform equipped with two exteroceptive sensors : a video camera and a laser rangefinder.

MOTS-CLES :

Cartographie de l'environnement, localisation de robots mobiles, statistiques robustes, filtrage de Kalman, intersection de covariances, modélisation stochastique, systèmes multicapteurs, segmentation d'images.

DISCIPLINE :

Génie Informatique, Automatique et Traitement du signal.

INTITULE ET ADRESSE DE L'U.F.R. OU DU LABORATOIRE :

Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier (LIRMM)
UMR CNRS/Université Montpellier II, no. C55060
161, rue ADA. 34392 - Montpellier - Cedex 5 - France