



HAL
open science

Contribution à la construction de mosaïques d'images sous-marines géo-référencées par l'introduction de méthodes de localisation

Manon Borgetto

► **To cite this version:**

Manon Borgetto. Contribution à la construction de mosaïques d'images sous-marines géo-référencées par l'introduction de méthodes de localisation. Interface homme-machine [cs.HC]. Université du Sud Toulon Var, 2005. Français. NNT: . tel-00009564

HAL Id: tel-00009564

<https://theses.hal.science/tel-00009564>

Submitted on 22 Jun 2005

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

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

UNIVERSITÉ DU SUD TOULON-VAR

THÈSE

pour obtenir le grade de DOCTEUR DE L'UNIVERSITÉ DU SUD TOULON-VAR

U.F.R. Sciences et Techniques

Spécialité Traitement du Signal et de l'Image

| |
|---|
| <p>CONTRIBUTION À LA CONSTRUCTION DE MOSAÏQUES D'IMAGES SOUS-MARINES GÉO-RÉFÉRENCÉES PAR L'INTRODUCTION DE MÉTHODES DE LOCALISATION</p> |
|---|

Présentée et soutenue publiquement le 5 avril 2005

par Manon BORGETTO

Devant le jury composé de :

Rapporteurs

Mme Marie-José ALDON, Chargée de recherche, laboratoire LIRMM, Montpellier
M. Gérard FAVIER, Directeur de recherche, laboratoire I3S, Sophia-Antipolis

Examineurs

M. Frédéric BOUCHARA, Maître de Conférences, Université du Sud Toulon-Var
M. Gilles ENÉA, Professeur, Université du Sud Toulon-Var
M. Vincent RIGAUD, Directeur de Département, Ifremer

Membres invités

Mme Camille BÉCHAZ, Ingénieur, ACSA Underwater GPS
Mme Maria-Jao RENDAS, Chargée de Recherche, laboratoire I3S, Sophia-Antipolis

Directeur de thèse

M. Claude JAUFFRET, Professeur, Université du Sud Toulon-Var

REMERCIEMENTS

Je tiens tout d'abord à remercier Monsieur Claude Jauffret, mon directeur de thèse, directeur du laboratoire GESSY, pour tout le temps qu'il a passé à m'encadrer et me guider pendant cette thèse. Je lui suis très reconnaissante de m'avoir fait confiance malgré ses réticences du début. Vous voyez, nous avons réussi à mener cette aventure à son terme !

Je remercie également Monsieur Vincent Rigaud, directeur du département Systèmes Sous-Marins d'Ifremer, de m'avoir donné la chance de poursuivre en thèse après mon stage de DEA dans le service RNV qu'il dirigeait en 2001.

Je remercie Madame Marie-José Aldon, chargée de recherche au laboratoire LIRMM de Montpellier, et Monsieur Gérard Favier, directeur de recherche au laboratoire I3S de Sophia-Antipolis, de m'avoir fait l'honneur d'être les rapporteurs de cette thèse.

Je tiens également à remercier Monsieur Frédéric Bouchara, Maître de Conférences à l'Université du Sud Toulon-Var, d'avoir accepté de faire partie de mon jury de thèse et Monsieur Gilles Enea, Professeur à l'Université du Sud Toulon-Var, d'avoir accepté de tenir le rôle de président du jury.

Je remercie Monsieur Hubert Thomas, directeur général d'ACSA underwater GPS, d'avoir suivi le fil de mes travaux dans le cadre de ma bourse de recherche PACA, et Mesdames Camille Béchaz, ingénieur ACSA underwater GPS, et Maria-Jao Rendas, Chargée de Recherche au laboratoire I3S, d'avoir accepté l'invitation à ma soutenance de thèse.

Je remercie également Monsieur Joël Denervaux, qui a été mon mentor pour la rédaction du nouveau chapitre de thèse : valorisation des compétences. Il m'a amenée à porter un regard différent sur mon travail, à l'apprécier comme un projet dans son ensemble (au sens de la gestion des ressources humaines, financières et du temps).

Cette thèse s'est déroulée pour moitié dans le laboratoire GESSY. Quels éclats de rire retentissants ! Merci à Jean de s'être toujours montré disponible pour gérer les problèmes d'imprimante (nombreux !) et infor-

matiques (plus rares...). Et puis bien sûr je remercie mes " co-bureautiers ", Saloua, qui va avoir la lourde tâche d'être la seule fille du bureau, et Nicolas le détecteur de neutrinos adepte de la basse (faudra que tu me signes un autographe, la célébrité arrive à grands pas ! !), et puis les nouveaux arrivants, Fabien et Rami. Après mes trois ans de thèse, on peut dire que le bureau a bien changé... D'une forêt de vieux PC, on est passé à quelques petits bosquets de plantes vertes (en espérant qu'ils survivent ! ! !).

L'autre moitié de cette thèse a été réalisée dans le service PRAO d'Ifremer. Tout d'abord, merci à Christian, qui s'est montré très disponible et m'a aidée à mettre en place mes expériences. Et puis surtout, je remercie tous ses membres pour leur chaleureuse amitié. Merci à Michel, Brigitte, Patricia, Michèle, Laurent, merci à tous ! Un merci tout spécial à Anne-Gaëlle, avec qui j'ai partagé un bureau... mais aussi bien plus qu'un bureau... et avec qui j'ai apprécié bon nombre de couchers de soleil qui ont rougi le port de la Seyne puis surtout le Faron ! ! Et puis, je n'oublie pas les danseurs de salsa, Toune et Henri en particulier, et notre prof Laurent aux fameux déhanchements... ! ! ! Alala, comment parler de ma thèse sans parler de salsa... ! !

Et puis un grand merci à Stellio, l'autre thésard Ifremer Toulon ;-). Je ne sais pas combien de mails de décompression on a pu s'écrire (oulala ces prises de tête ! ! !), mais bon, ça nous a bien aidés ces derniers mois ! ! Merci pour ton amitié et vive la vie ! !

Merci à ma grande copine Nathalie, qui m'a ouvert la voie, dans bien des domaines et pas seulement en auto-calibrage de caméra ;-)

Merci à ma famille, mes grands-parents, tontons et tatas, cousin et cousines, merci à mes parents, merci à ma petite soeur Julie (qui n'est pas si petite que ça mais qui le reste toujours un peu pour moi). Merci d'avoir toujours été là ! !

Enfin, un merci spécial à mon amoureux, Nicolas, qui m'a aidée à conclure cette aventure professionnelle et humaine que j'avais commencée toute seule, afin de pouvoir en entamer une autre à deux...

A Papi, qui n'est plus avec nous, mais qui nous suit toujours du coin de l'oeil...

Table des matières

| | |
|---|----------|
| Notations et définitions | I |
| 1 Introduction | 1 |
| 2 État de l’art des méthodes de localisation sous-marines | 5 |
| 2.1 Le milieu sous-marin et ses spécificités | 5 |
| 2.2 La navigation sous-marine | 6 |
| 2.2.1 Navigation à l’estime | 7 |
| 2.2.1.1 Capteurs d’attitude | 8 |
| 2.2.1.2 Capteurs de vitesse | 8 |
| 2.2.1.3 Tableau récapitulatif des capteurs employés | 9 |
| 2.2.1.4 Limitations de la navigation à l’estime | 9 |
| 2.2.2 Navigation inertielle | 10 |
| 2.2.2.1 Capteurs d’accélération | 10 |
| 2.2.2.2 Capteurs de vitesse angulaire | 10 |
| 2.2.2.3 La centrale inertielle | 11 |
| 2.2.3 Navigation acoustique | 11 |
| 2.2.3.1 Bases longues (BL) | 12 |
| 2.2.3.2 Système GIB | 13 |
| 2.2.3.3 Bases ultracourtes (BUC) | 13 |
| 2.2.3.4 Tableau récapitulatif | 14 |
| 2.2.3.5 Limitations du système | 14 |
| 2.2.3.6 Mesures d’immersion ou de profondeur | 15 |
| 2.2.4 Comparatif des différentes navigations en milieu sous-marin | 15 |
| 2.2.5 Navigation géophysique | 16 |
| 2.2.6 Fusion en milieu sous-marin | 17 |
| 2.2.6.1 Fusion estime/BUC | 17 |
| 2.3 La problématique plus générale de localisation | 19 |
| 2.3.1 Capteurs extéroceptifs | 19 |
| 2.3.2 Capteurs proprioceptifs | 20 |
| 2.3.3 Quelques exemples de navigation en milieu non sous-marin | 20 |
| 2.3.3.1 Navigation référencée terrain | 20 |

| | | |
|----------|--|-----------|
| 2.3.3.2 | SLAM | 21 |
| 2.4 | Fusion pour la navigation | 22 |
| 2.4.1 | Quelques exemples de fusion de données | 22 |
| 3 | Localisation par mesures de cap et vitesse et instants de rendez-vous estimés | 25 |
| 3.1 | Modélisation du problème en 2 dimensions | 25 |
| 3.1.1 | Trajectoire simple : connaissance seule des points extrémaux | 27 |
| 3.1.2 | Trajectoire complexe : connaissance supplémentaire des instants de rendez-vous | 27 |
| 3.1.2.1 | Définition des instants de rendez-vous | 27 |
| 3.1.2.2 | Problème de discrétisation | 27 |
| 3.1.3 | Modélisation proposée pour la trajectoire | 28 |
| 3.1.4 | Les mesures | 29 |
| 3.2 | Algorithmes de filtrage récursifs adaptés au problème | 29 |
| 3.2.1 | Le filtrage de Kalman discret | 30 |
| 3.2.2 | Lisseur par filtrage de Kalman Aller-Retour | 31 |
| 3.2.3 | Lissage RTS | 32 |
| 3.2.4 | Le filtrage de Kalman étendu (Extended Kalman Filter - EKF) | 32 |
| 3.3 | Estimation par la méthode des moindres carrés | 34 |
| 3.3.1 | Position du problème | 34 |
| 3.3.2 | Minimisation du critère $Q(\theta)$ | 34 |
| 3.4 | Une nouvelle méthode : l'auto-estimation | 36 |
| 3.4.1 | Estimation aller | 37 |
| 3.4.2 | Estimation retour | 37 |
| 3.4.3 | Aller-retour | 37 |
| 3.4.4 | Calcul des biais et variances | 38 |
| 3.4.5 | Réduction du biais | 38 |
| 3.5 | Généralisation aux trajectoires complexes | 38 |
| 3.5.1 | Stratégie de parcours par l'extérieur | 39 |
| 3.5.2 | Stratégie de parcours par utilisation de la théorie des graphes | 40 |
| 3.5.2.1 | Définitions | 40 |
| 3.5.2.2 | Recherche du plus court chemin | 41 |
| 3.5.2.3 | Algorithme de Dijkstra - Principe | 41 |
| 3.5.2.4 | Stratégie de parcours de graphe proposée | 42 |
| 3.5.3 | Cas particulier de la méthode des moindres carrés | 43 |
| 3.6 | Conclusion | 44 |
| 4 | Etude des performances et résultats relatifs à la localisation AR/(C,V) | 45 |
| 4.1 | Outils d'étude des performances | 45 |
| 4.1.1 | Matrice d'information de Fisher | 45 |
| 4.1.2 | Borne de Cramer-Rao | 46 |
| 4.1.3 | Calcul dans le cas d'une trajectoire simple | 46 |

| | | |
|----------|--|-----------|
| 4.1.4 | Calcul dans le cas d'une trajectoire complexe à un seul instant de rendez-vous | 48 |
| 4.1.5 | Généralisation | 48 |
| 4.1.6 | Performances désirées d'un estimateur | 49 |
| 4.2 | Contexte d'évaluation des méthodes | 49 |
| 4.2.1 | Hypothèses sur le bruit | 49 |
| 4.2.2 | Présentation des trajectoires simulées | 50 |
| 4.3 | Comparaison des algorithmes proposés | 50 |
| 4.3.1 | Kalman aller-retour et lissage RTS | 50 |
| 4.3.2 | Méthode des moindres carrés | 52 |
| 4.3.3 | Méthode d'auto-estimation | 56 |
| 4.3.4 | Conclusion | 57 |
| 4.4 | Comparaison des résultats selon le type de stratégie employée | 59 |
| 4.5 | Influence de la modélisation des bruits de mesure | 60 |
| 4.5.1 | Comparaison des résultats obtenus avec un bruit blanc uniforme | 60 |
| 4.5.2 | Comparaison des résultats obtenus avec un mélange de bruits de lois différentes | 61 |
| 4.6 | Résultats obtenus sur différentes trajectoires | 62 |
| 4.6.1 | Trajectoire quadrillée | 62 |
| 4.6.2 | Trajectoire "graphe" | 64 |
| 4.7 | Résumé et conclusions | 66 |
| 5 | Estimation des instants de rendez-vous et localisation à partir d'une séquence d'images | 69 |
| 5.1 | Les images acquises en milieu sous-marin | 70 |
| 5.1.1 | Capteurs d'images | 70 |
| 5.1.2 | Les problèmes dus aux grands fonds | 71 |
| 5.1.3 | Un pré-traitement : la correction d'intensité lumineuse | 73 |
| 5.1.3.1 | Correction radiométrique | 73 |
| 5.1.3.2 | Filtrage homomorphique | 75 |
| 5.2 | Techniques nécessaires à l'estimation de la transformation entre deux images | 77 |
| 5.2.1 | Détection de caractéristiques | 77 |
| 5.2.1.1 | Détection de contours | 77 |
| 5.2.1.2 | Extraction de points dans une image | 78 |
| 5.2.2 | Caractérisation des points extraits | 80 |
| 5.2.3 | Mise en correspondance de points | 81 |
| 5.2.3.1 | Quelques critères mesurant la ressemblance entre deux primitives | 81 |
| 5.2.3.2 | Méthodes de mise en correspondance d'images | 81 |
| 5.3 | Recherche des "instants de rendez-vous" : utilisation de l'algorithme PISA [Sis00] | 83 |
| 5.3.1 | L'algorithme PISA [Sis00] | 83 |
| 5.3.2 | Extension proposée à la PISA : les moindres carrés linéaires (MCL) | 87 |
| 5.3.3 | Utilisation de la PISA à la recherche des instants de rendez-vous | 88 |
| 5.4 | Estimation de mouvement inter-image et construction de mosaïques d'images à partir d'un flux vidéo | 89 |

| | | |
|---------|--|-----|
| 5.4.1 | Construction de mosaïques d'images sous-marines | 89 |
| 5.4.2 | Suivi de points par l'algorithme KLT [ST94] | 91 |
| 5.4.3 | Flot optique et pyramide multi-résolution : cas particulier du RMR [OB95] | 93 |
| 5.4.3.1 | Flot optique | 93 |
| 5.4.3.2 | Intérêt d'une approche multi-résolution pour la robustesse | 94 |
| 5.4.3.3 | Modèle introduit | 94 |
| 5.4.3.4 | Critère de minimisation | 95 |
| 5.4.3.5 | Estimation multi-résolution | 98 |
| 5.4.4 | Comparaison des méthodes d'estimation KLT et RMR | 98 |
| 5.5 | Localisation AR/I | 99 |
| 5.6 | Cadre expérimental | 100 |
| 5.6.1 | PISCATOR et matériel | 100 |
| 5.6.2 | Le système optique | 101 |
| 5.6.3 | Les trajectoires programmées | 102 |
| 5.6.4 | Instants de rendez-vous | 102 |
| 5.7 | Résultats relatifs à la recherche des instants de rendez-vous | 102 |
| 5.7.1 | Etude des paramètres de l'algorithme PISA | 102 |
| 5.7.1.1 | Paramètres de détection des points d'intérêt et mise en correspondance avant estimation | 103 |
| 5.7.1.2 | Etude du nombre d'itérations des estimateurs proposées par Sistiaga | 104 |
| 5.7.1.3 | Comparaison des différents estimateurs | 105 |
| 5.7.2 | Recherche des instants de rendez-vous à partir de la séquence d'images | 106 |
| 5.7.3 | Conclusion | 108 |
| 5.8 | Résultats relatifs à la construction de mosaïques 2D | 109 |
| 5.8.1 | Influence de la correction d'intensité lumineuse sur la génération et le rendu de mosaïques | 109 |
| 5.8.1.1 | Correction radiométrique automatique | 110 |
| 5.8.1.2 | Filtrage homomorphique | 110 |
| 5.8.2 | Influence du fenêtrage de l'image lors de l'estimation des déplacements de la mosaïque | 113 |
| 5.8.2.1 | Résultats obtenus pour la méthode RMR | 113 |
| 5.8.2.2 | Résultats obtenus pour la méthode KLT | 114 |
| 5.8.2.3 | Conclusion | 114 |
| 5.8.3 | Comparaison des algorithmes de construction de mosaïque : KLT et RMR | 118 |
| 5.8.3.1 | Comparaison sur la mosaïque construite | 118 |
| 5.8.3.2 | Comparaison sur les déplacements estimés | 118 |
| 5.8.3.3 | Comparaison sur les trajectoires reconstruites | 123 |
| 5.8.3.4 | Conclusion | 123 |
| 5.8.4 | Réalisation de la localisation AR/I | 126 |
| 5.9 | Conclusion | 127 |

| | | |
|----------|---|------------|
| 6 | Faisabilité de la fusion entre localisation AR/I et localisation AR/(C,V) | 129 |
| 6.1 | Fusion de données | 129 |
| 6.2 | Référentiel commun aux localisation AR/(C,V) et AR/I | 130 |
| 6.2.1 | Modélisation d'une caméra | 130 |
| 6.2.2 | Changement de repère | 130 |
| 6.2.3 | Calibrage d'une caméra | 132 |
| 6.2.4 | Auto-calibrage en milieu sous-marin | 133 |
| 6.2.5 | Influence de la précision sur la distance (caméra, sol) | 134 |
| 6.3 | Fusion des localisations AR/(C,V) et AR/I | 136 |
| 6.4 | Expérimentations/résultats | 137 |
| 6.4.1 | Auto-calibrage de la caméra | 137 |
| 6.4.2 | Influence de la position de la caméra | 137 |
| 6.4.3 | Comparaison localisation A/(C,V) et localisation AR/(C,V) | 140 |
| 6.4.4 | Comparaison des localisations AR/(C,V) et AR/I | 143 |
| 6.4.5 | Fusion localisation AR/(C,V) et localisation AR/I | 149 |
| 6.5 | Conclusion | 149 |
| 7 | Alternative : déformation de la mosaïque initiale par localisation AR/(C,V) | 151 |
| 7.1 | Introduction à la déformation d'images | 151 |
| 7.2 | Quelques exemples d'utilisation du warping en traitement d'images | 152 |
| 7.3 | Transformations spatiales | 153 |
| 7.4 | La déformation d'images en pratique | 154 |
| 7.5 | Application du warping à la déformation de la mosaïque | 154 |
| 7.5.1 | Principe de base de la méthode | 154 |
| 7.5.2 | Extension à plusieurs couples de segments | 156 |
| 7.5.3 | Déformation de la mosaïque à partir de la localisation AR/(C,V) | 157 |
| 7.6 | Expérimentations | 158 |
| 7.6.1 | Un exemple de déformation de la mosaïque | 158 |
| 7.6.2 | Etude des paramètres de la déformation à partir de plusieurs paires de segments | 159 |
| 7.7 | Conclusion | 162 |
| 8 | Conclusion et perspectives | 163 |
| 8.1 | Récapitulatif : le système de correction de mosaïques | 163 |
| 8.2 | Discussion et perspectives | 165 |
| | Bibliographie | 167 |
| A | Détails du calcul des biais, variances et covariances | I |
| A.1 | Démonstration de la formule (3.36) de la partie 3.4.4 | I |
| A.2 | Détails du calcul des biais, variances, covariances dans la navigation à l'estime | I |
| A.2.1 | Calcul des biais | II |
| A.2.1.1 | Espérance de \hat{x}_k et biais | II |

| | | |
|----------|--|-------------|
| A.2.1.2 | Espérance de \hat{y}_k et biais | III |
| A.2.2 | Calcul des variances | III |
| A.2.2.1 | Variance de x | III |
| A.2.2.2 | Variance de y | IV |
| A.2.3 | Calcul des covariances | V |
| A.3 | Biais, variances et covariances de l'estime après réduction du biais | VII |
| A.3.1 | Réduction du biais | VIII |
| A.3.2 | Calcul des variances après correction du biais | VIII |
| A.3.3 | Calcul de la covariance après correction du biais | VIII |
| A.4 | Détails du calcul des biais, variances, covariances pour la trajectoire auto-estimée | IX |
| A.4.1 | Biais de l'estimateur | IX |
| A.4.2 | Matrice de variance-covariance | IX |
| A.4.3 | Exemple de calculs pour le cas d'une trajectoire en ligne droite | X |
| B | Critères de confiance de la PISA | XIII |
| B.1 | Critère de confiance λ_k sur un appariement k | XIII |
| B.2 | Critères de confiance pour la qualification de l'estimation | XIV |
| C | Le logiciel MATISSE® | XV |

Table des figures

| | | |
|------|--|----|
| 2.1 | Deux exemples d'engins sous-marins utilisés à l'Ifremer. | 7 |
| 2.2 | Navigation à l'estime | 7 |
| 2.3 | Principe d'une base longue | 12 |
| 2.4 | Principe d'une base ultra courte | 14 |
| 2.5 | Résultats obtenus par la méthode de Jouffroy [JO04] | 18 |
| 2.6 | Étapes du SLAM | 21 |
| 3.1 | Exemple de trajectoire lors du quadrillage d'une zone d'intérêt. | 26 |
| 3.2 | Exemples de trajectoires intégrées à partir des caps et vitesses bruités. | 26 |
| 3.3 | Problème apparaissant lors du calcul des instants de rendez-vous. | 28 |
| 3.4 | Trajectoires assimilées à des lignes polygônales. | 28 |
| 3.5 | Exemple de stratégie de parcours par l'extérieur sur la trajectoire quadrillée. | 39 |
| 3.6 | Exemple de stratégie de parcours de graphe sur la trajectoire quadrillée. | 42 |
| 3.7 | Exemple de stratégie de parcours par le plus court chemin sur la trajectoire "graphe". | 43 |
| 4.1 | Structure de la matrice de Fisher dans le cas d'une trajectoire à un seul instant de rendez-vous. | 48 |
| 4.2 | Trajectoires générées (en mètres). | 50 |
| 4.3 | Comparaison entre EKF aller-retour et lissage RTS. | 51 |
| 4.4 | Problème de "décrochage". | 52 |
| 4.5 | Exemple de trajectoire estimée dans le cas du "alpha". | 53 |
| 4.6 | Comparaison des résultats obtenus avec des initialisations différentes dans le cas du "alpha". | 53 |
| 4.7 | Estimation de la trajectoire "alpha" par les moindres carrés (1000 simulations). | 54 |
| 4.8 | Estimation de la trajectoire "quadrillée" par les moindres carrés (500 simulations). | 55 |
| 4.9 | Faisceau de trajectoires estimées. | 56 |
| 4.10 | Comparaison des biais théoriques (sans correction de biais) et simulés sans et avec réduction du biais pour la trajectoire "alpha". | 58 |
| 4.11 | Comparaison des variances théoriques (sans correction de biais) et simulées sans et avec réduction du biais pour la trajectoire "alpha". | 58 |
| 4.12 | Comparaison des variances théoriques (sans correction de biais) et de la BCR pour la trajectoire "alpha". | 58 |
| 4.13 | Comparaison des biais et variances selon la stratégie. | 59 |

| | | |
|------|--|-----|
| 4.14 | Comparaison des biais sans réduction de biais et avec réduction de biais, et variances obtenus avec une loi gaussienne et une loi uniforme, sur 5000 trajectoires "alpha". | 60 |
| 4.15 | Comparaison des biais sans réduction de biais et avec réduction de biais, et variances obtenus à partir d'une loi gaussienne et à partir d'un mélange de bruit, sur 5000 trajectoires "alpha". | 61 |
| 4.16 | Faisceau de trajectoires estimées (en mètres). La trajectoire vraie est en blanc. | 62 |
| 4.17 | Comparaison des biais théoriques (sans correction de biais) et empiriques avant et après réduction du biais pour la trajectoire quadrillée. | 63 |
| 4.18 | Comparaison des variances théoriques (avant correction de biais) et empiriques avant et après réduction du biais pour la trajectoire quadrillée. | 63 |
| 4.19 | Comparaison des variances théoriques (avant correction de biais) et de la BCR pour la trajectoire quadrillée. | 63 |
| 4.20 | Faisceau de trajectoires estimées. | 64 |
| 4.21 | Comparaison des biais théoriques (sans correction de biais) et empiriques avant et après réduction du biais pour la trajectoire "graphe". | 65 |
| 4.22 | Comparaison des variances théoriques (avant correction de biais) et empiriques avant et après réduction du biais pour la trajectoire "graphe". | 65 |
| 4.23 | Comparaison des variances théoriques (avant correction de biais) et de la BCR pour la trajectoire "graphe". | 65 |
| 5.1 | Caractéristiques de l'absorption de la lumière dans l'eau. | 72 |
| 5.2 | Image d'une épave d'avion, avec halo lumineux. | 73 |
| 5.3 | Phénomènes liés à l'objectif et à la source lumineuse. | 74 |
| 5.4 | Principe du filtrage homomorphique. | 76 |
| 5.5 | Appariement graphes. | 82 |
| 5.6 | Principe de la PISA. | 84 |
| 5.7 | Principe du suivi de points. | 92 |
| 5.8 | Principe de la construction d'une pyramide d'images. | 94 |
| 5.9 | Deux niveaux de la pyramide de l'image 3 de la séquence "Lucky Luke". | 95 |
| 5.10 | Le portique PISCATOR sur le bassin d'essai. | 101 |
| 5.11 | Trajectoires réalisées dans le bassin d'essais intérieur. | 102 |
| 5.12 | Zoom sur l'instant de rendez-vous de la trajectoire "alpha". | 103 |
| 5.13 | Images 3 et 191 de la séquence "Lucky Luke". | 103 |
| 5.14 | Image de produits d'entretien. | 105 |
| 5.15 | Principe de la séquence augmentée. | 110 |
| 5.16 | Correction radiométrique automatique avec image de référence acquise en pleine eau. | 111 |
| 5.17 | Filtrage homomorphique. | 112 |
| 5.18 | Méthode RMR : influence du fenêtrage des images sur la construction de mosaïque. | 115 |
| 5.19 | Méthode KLT : influence du fenêtrage des images sur la construction de mosaïque. | 116 |
| 5.20 | Influence du fenêtrage des images sur la construction de mosaïque RMR, au niveau de la trajectoire. | 117 |

| | | |
|------|--|-----|
| 5.21 | Influence du fenêtrage des images sur la construction de mosaïque KLT, au niveau de la trajectoire. | 117 |
| 5.22 | Comparaison des mosaïques construites à partir du RMR et du KLT pour la trajectoire "alpha". | 119 |
| 5.23 | Comparaison des mosaïques construites à partir du RMR et du KLT pour une trajectoire à deux boucles. | 120 |
| 5.24 | Comparaison des déplacements obtenus par les méthodes RMR (+) et KLT (\times), pour la trajectoire "alpha". | 121 |
| 5.25 | Comparaison des déplacements obtenus par les méthodes RMR (+) et KLT (\times), pour la trajectoire à deux boucles. | 122 |
| 5.26 | Comparaison des trajectoires obtenues par les méthodes RMR (en pointillés '–') et KLT (en pointillés '..'), pour la trajectoire "alpha". | 124 |
| 5.27 | Comparaison des trajectoires obtenues par les méthodes RMR (en pointillés '–') et KLT (en pointillés '..'), pour la trajectoire à deux boucles. | 125 |
| 5.28 | Localisation A/I (RMR), localisation AR/I, et trajectoire auto-estimée pour la trajectoire alpha. | 126 |
| 6.1 | Modèle de caméra sténopé. | 131 |
| 6.2 | Changement de repère. | 131 |
| 6.3 | Méthode d'auto-calibrage développée par N. Pessel [Pes03]. | 133 |
| 6.4 | Influence de t_z sur la transformation 3D \rightarrow 2D pour la localisation AR/(C,V). | 138 |
| 6.5 | Comparaison des mosaïques construites à partir de la trajectoire vraie 2D calculée pour trois altitudes différentes, pour la trajectoire "alpha". | 139 |
| 6.6 | Comparaison trajectoires bruitée et auto-estimée pour la trajectoire "alpha". | 141 |
| 6.7 | Comparaison trajectoires bruitée et auto-estimée pour la trajectoire à deux boucles. | 142 |
| 6.8 | Comparaison des localisations AR/I et AR/(C,V) pour la trajectoire "alpha". | 145 |
| 6.9 | Comparaison des localisations AR/I et AR/(C,V) pour la trajectoire à deux boucles. | 146 |
| 6.10 | Comparaison des mosaïques construites par la trajectoire vraie, le RMR et par les localisations AR/(C,V) et AR/I pour la trajectoire "alpha". | 147 |
| 6.11 | Comparaison des mosaïques construites par la trajectoire vraie, le RMR et par les localisations AR/(C,V) et AR/I pour la trajectoire à deux boucles. | 148 |
| 6.12 | Fusion des localisations AR/(C,V) et AR/I pour la trajectoire "alpha". | 149 |
| 7.1 | Exemples de déformation. | 152 |
| 7.2 | Principe du warping. | 155 |
| 7.3 | Exemples de déformations à partir d'une paire de segments. | 156 |
| 7.4 | Exemples de déformations à partir de plusieurs paires de segments. | 157 |
| 7.5 | Exemple de déformation sur une mosaïque de 7 images. | 158 |
| 7.6 | Image initiale, avec les paires de segments surexposés, utilisée pour l'étude du paramètre a . . | 160 |
| 7.7 | Influence du paramètre a sur la déformation d'une image, pour $p = 0$ et $b = 0.5$ | 160 |
| 7.8 | Image initiale, avec les paires de segments surexposés, utilisée pour l'étude des paramètres p et b | 161 |
| 7.9 | Influence des paramètres p et b sur la déformation d'une image, pour $a = 1$ | 161 |

| | | |
|-----|--|-----|
| 8.1 | Synoptique de l'amélioration de mosaïques d'images proposée. | 164 |
| C.1 | Un exemple d'utilisation de MATISSE. | XVI |

Notations et définitions

Notations adoptées

Divers

\mathbf{I} : image considérée

$\mathbf{I}(x, y)$: intensité en niveaux de gris, du pixel de position (x, y)

(I, J) : couple d'instant de rendez-vous

F_I : voisinage d'images autour de l'image d'indice I : \mathbf{I}_I .

$X_k = (x_k, y_k)^T$: position en mètres du véhicule à l'instant t_k

θ : paramètre à estimer

Localisation par mesures de cap et vitesse

c_k : cap vrai à l'instant t_k

v_k : vitesse vraie à l'instant t_k

\tilde{c}_k : cap mesuré à l'instant t_k

\tilde{v}_k : vitesse mesuré à l'instant t_k

z_k : signal mesuré à l'instant t_k (cap ou vitesse)

σ_c : écart-type du bruit blanc additif sur les caps

σ_v : écart-type du bruit blanc additif sur les vitesses

$\hat{X}_{k|k}^+$: position estimée à l'instant t_k , connaissant les $k - 1$ premières mesures de cap et de vitesse

$\hat{X}_{k|N-k}^-$: position estimée à l'instant t_k , connaissant les $N - k$ mesures de cap et de vitesse, en partant de l'instant final.

$\hat{X}_{k|N}$: position estimée à partir des $N - 1$ mesures de cap et vitesse.

$\tilde{X}_{k|k}^+$: position estimée à l'instant t_k , connaissant les $k - 1$ premières mesures de cap et de vitesse, après réduction de biais.

$\tilde{X}_{k|N-k}^-$: position estimée à l'instant t_k , connaissant les $N - k$ mesures de cap et de vitesse, en partant de l'instant final, après réduction de biais.

$\tilde{X}_{k|N}$: position estimée à partir des $N - 1$ mesures de cap et vitesse, après réduction de biais.

$I_\theta(z_1, \dots, z_n)$: information de Fisher pour les mesures (z_1, \dots, z_n) .

BCR_θ : Borne de Cramer-Rao

Correction de l'intensité lumineuse

Pour une matrice CCD :

g_{sens} : image représentant le gain modélisant les écarts de sensibilité,

\mathbf{I}_0 : image en l'absence d'excitation lumineuse,

\mathbf{I}_{transm} : image représentant un bruit de transmission entre la caméra et la carte d'acquisition,

g_{obj} : image représentant la non-uniformité spatiale,

\mathbf{I}_a : image acquise,

\mathbf{I}_{exacte} : image exacte, que l'on aurait acquis dans un cas idéal (sans les différents bruits et non-uniformités).

$i(x, y)$: terme d'illumination

$r(x, y)$: terme de réflectance

Estimation des instants de rendez-vous

V_{AD} : vecteur d'attributs différentiels

P_1 : point extrait dans l'image 1, P_2 : son correspondant dans l'image 2, P'_1 : le point correspondant à P_1 reprojété dans l'image 2

R_α : rotation d'angle α entre les deux images considérées, $T = (t_x, t_y)$: translation, ρ : changement de résolution (zoom).

MC : estimateur des moindres carrés

MCP : estimateur des moindres carrés pondérés

EKF : filtrage de Kalman étendu

EKFP : filtrage de Kalman étendu pondéré

MCL : estimateur des moindres carrés linéaires

MCLP : estimateur des moindres carrés linéaires pondérés

Construction de mosaïque d'images

RMR (Robust Multi-Resolution) : algorithme d'estimation pyramidale du mouvement 2D d'une image.

KLT (Kanade, Lucas, Tomasi) : algorithme d'estimation du mouvement 2D d'une image, par une méthode de suivi de points.

Etude de la faisabilité de la fusion localisation par image/ localisation par mesures de cap et vitesse

\mathbb{K} : matrice des paramètres intrinsèques d'une caméra.

\mathbb{P}_{t_z} : transformation 3D->2D, dans le cas d'une caméra se déplaçant à altitude constante t_z .

Δ_x : déplacement en x .

Annexe

$\mathbf{E}(X)$: espérance mathématique de la variable aléatoire X

$b_{\hat{X}}(k)$: biais de la position estimée \hat{X} à l'instant t_k .

$\text{Var}(\hat{x}_k)$: variance de l'abscisse estimée \hat{x}_k à l'instant t_k .

$\text{Cov}(\hat{x}_k, \hat{y}_k)$: covariance du couple de position estimé (\hat{x}_k, \hat{y}_k) à l'instant t_k .

$\text{P}(\hat{X}_k)$: matrice de variance-covariance de la position estimée \hat{X}_k à l'instant t_k .

Définitions

Trajectoire

Ensemble des positions du véhicule sous-marin (estimées par une méthode quelconque).

Trajectoire vraie

Ensemble des positions effectivement suivies par le véhicule. En particulier, ce terme de "trajectoire vraie" sera utilisé pour désigner la trajectoire qu'on aurait si les mesures de cap et de vitesse n'étaient pas bruitées.

Couple d'instants de rendez-vous (RdV)

Couple d'instants pour lesquels l'engin repasse au-dessus d'une même zone.

Localisation Aller (A)

Estimation de la position du véhicule à l'instant t_k par les mesures antérieures (images ou caps+vitesse).

Localisation Retour (R)

Estimation de la position du véhicule à l'instant t_k par les mesures postérieures (images ou caps+vitesse).

Localisation Aller/Retour (AR)

Estimation de la position du véhicule à l'instant t_k par toutes les mesures disponibles (images ou caps+vitesse), en tenant compte des instants de RdV.

Localisation A/(C,V)

Localisation aller par mesures de cap et vitesse et instants de rendez-vous estimés.

Localisation R/(C,V)

Localisation retour par mesures de cap et vitesse et instants de rendez-vous estimés.

Localisation AR/(C,V)

Localisation aller-retour par mesures de cap et vitesse et instants de rendez-vous estimés.

Localisation A/I

Localisation aller à partir de la séquence d'images et des instants de rendez-vous estimés.

Localisation R/I

Localisation retour à partir de la séquence d'images et des instants de rendez-vous estimés.

Localisation AR/I

Localisation aller-retour à partir de la séquence d'images et des instants de rendez-vous estimés.

Localisation AR/(C,V,I)

Localisation aller-retour à partir des mesures de cap et de vitesse, de la séquence d'images et des instants de rendez-vous estimés.

Trajectoire simple

Trajectoire estimée sans tenir compte des instants de rendez-vous.

Trajectoire complexe

Trajectoire estimée en tenant compte de la géométrie d'exploration de la zone, notamment des instants de rendez-vous.

Chapitre 1

Introduction

Depuis toujours, l'Homme cherche à repousser les limites de ses connaissances. Le domaine terrestre une fois exploré, il s'est tourné vers les vastes étendues d'eau et leurs mystères...

Les hommes ont pour la première fois traversé les mers pour le négoce : il s'agissait de trouver de nouvelles routes commerciales sur les mers pour éviter la longue traversée de pays inhospitaliers avant d'atteindre au but.

Puis un peu plus tard, pour des motifs un peu moins lucratifs et un peu plus scientifiques, l'Homme s'est tourné vers l'exploration des profondeurs. Quels vestiges d'une ancienne civilisation, quelles nouvelles formes de vie, quels trésors géologiques, quels secrets allait-on découvrir au fond des océans ?

Bien qu'une grande partie des profondeurs des océans reste encore à découvrir, les recherches sous-marines et les évolutions permanentes de la technologie permettent de repousser nos limites et de dévoiler petit à petit certains de ces mystères. Pendant longtemps, beaucoup d'efforts de recherche et d'étude ont été dirigés vers l'espace alors que la mer représente 80% de la surface terrestre. C'est une des raisons pour lesquelles les recherches sous-marines sont relativement jeunes.

L'exploration sous-marine peut être réalisée par l'intermédiaire de cartes visuelles sous-marines, ou mosaïques d'images sous-marines. Celles-ci permettent d'obtenir la vue globale d'une zone d'intérêt (par exemple une épave de bateau ou d'avion, un champ d'algues...) à partir d'une séquence d'images. Celle-ci est acquise par une caméra fixée sur un véhicule sous-marin explorant une zone. Trois spots lumineux au moins permettent d'éclairer la scène sous-marine (la lumière étant très atténuée avec la profondeur d'eau).

Ifremer a développé le logiciel MATISSE® qui permet de construire et visualiser des mosaïques d'images en temps réel à partir d'une séquence d'images seule.

Plusieurs problèmes surgissent lors de la construction d'une telle mosaïque. Le premier problème est relatif à la qualité de l'image. On peut notamment remarquer que l'éclairage artificiel de la scène sous-marine provoque un halo lumineux au centre des images acquises.

Le deuxième problème est lié à l'estimation du mouvement entre deux images successives. Cette estimation provoque une dérive de la mosaïque par rapport à la trajectoire effectivement suivie par le véhicule.

Ifremer a ressenti la nécessité de corriger ces problèmes inhérents à la construction de mosaïque. D'autant plus que ces travaux se replacent dans une continuité de recherches en imagerie sous-marine. Sistiaga [Sis00] a étudié la navigation référencée terrain et Pessel [Pes03] a développé une méthode d'auto-calibrage de caméra.

C'est pourquoi l'Ifremer a proposé d'initier des travaux de recherche afin d'**améliorer la fiabilité** et de **localiser au mieux de telles cartes visuelles**. Ces travaux ont été développés à l'Université du Sud Toulon-Var.

De façon plus pratique, les recherches ont été abordées en considérant que pour réaliser une couverture optimale de la zone d'exploration, le véhicule porteur de la caméra suivait une trajectoire de type quadrillage. Grossièrement, cette trajectoire est composée de colonnes et de lignes qui se recoupent par endroits (c'est ce qu'on appellera les *instants de rendez-vous*). Nous avons donc voulu tenir compte de ce fait pour proposer des solutions adaptées à cette particularité de parcours de la zone d'exploration.

D'autre part, il s'avère que l'engin transporte des capteurs autres qu'une caméra. En autres, il s'agit de capteurs de cap et de vitesse. Parallèlement aux positions estimées à partir de la séquence d'images seule, ceux-ci permettent d'obtenir par le biais d'une intégration une estimation des positions de l'engin. L'idée que nous proposons ici est d'améliorer la mosaïque en introduisant les mesures de cap et de vitesse.

Étant donné que le sujet de recherche était nouveau, des hypothèses simplificatrices du problème ont été données. Nous nous sommes placés dans un contexte d'exploration en deux dimensions (le relief n'a pas été pris en compte et on a supposé que la caméra se déplaçait à profondeur constante). D'autre part, les mesures image et de cap et vitesse sont supposées synchronisées.

Le **principe général** de la solution proposée est donc le suivant :

1. Estimer les instants de rendez-vous,
2. Réaliser une estimation de la position du véhicule à partir des mesures de cap et vitesse et des instants de rendez-vous estimés (Localisation AR/(C,V)),
3. Réaliser une estimation de la position du véhicule à partir de la séquence d'images et des instants de rendez-vous estimés (Localisation AR/I),
4. Réaliser la fusion des estimations par mesures de cap et vitesse et par séquence d'images (Localisation AR/(C,V,I)),
5. Reconstruire la mosaïque.

Les deux derniers points n'étant pas toujours réalisables (fusion impossible ou séquence d'images pas ou plus disponibles), nous avons proposé une alternative qui consiste à déformer la mosaïque déjà construite à partir de l'estimation de la position du véhicule par mesures de cap et vitesse.

La progression de ce mémoire est la suivante : tout d'abord, un état de l'art sur les capteurs et les techniques de positionnement des véhicules sous-marins et quelques exemples de fusion de données sont exposés dans le **chapitre 2**.

Puis dans le **chapitre 3**, quelques méthodes d'estimation de la position de l'engin à partir de mesures de cap et vitesse sont décrites dans un premier temps. En particulier, une méthode originale, l'auto-estimation, est proposée. Dans un second temps, les particularités de la trajectoire d'exploration de la zone sont exploitées pour réaliser la localisation AR/(C,V). Les instants de rendez-vous (supposés connus dans ce chapitre mais sont déterminés à partir des images et d'une méthode développée dans le chapitre 5) sont inclus dans l'estimation de la position du véhicule sous-marin [BRL03]. Le **chapitre 4** est un chapitre de résultats qui permet la qualification des méthodes présentées sur des simulations.

Le **chapitre 5** est décomposé en trois parties. Après un bref exposé des contraintes et problèmes liés à l'acquisition d'images sous-marines [BRL03], un état de l'art sur les méthodes classique pour la mise en correspondance d'images est présenté. A partir de ces techniques, est exposée la méthode proposée pour rechercher les *instants de rendez-vous* à partir des images. La deuxième partie définit ce qu'est une mosaïque d'images et expose quelques méthodes de construction de telles mosaïques. Une localisation AR/I est proposée, reposant sur un algorithme d'aller-retour. La troisième partie de ce chapitre est consacrée à l'étude expérimentale des méthodes proposées.

Le **chapitre 6** traite de la faisabilité de la fusion des deux localisations AR obtenues, l'une obtenue à partir de mesures de cap et vitesse et des instants de rendez-vous estimés, l'autre obtenue à partir de la séquence d'images et des instants de rendez-vous estimés. Après une introduction à la recherche d'un référentiel commun pour fusionner les localisations AR, des résultats comparant les deux trajectoires en position seront exposés. Enfin, une étude sera menée sur la fusion de celles-ci (localisation AR/(C,V,I)).

Le **chapitre 7** décrit l'alternative proposée dans le cas de l'impossibilité de réaliser la fusion des deux estimations de position (par caps et vitesses, par images). La méthode proposée consiste donc à déformer la carte visuelle, *i.e.* la mosaïque, à partir de la localisation AR/(C,V) (positions estimées par mesures de cap et vitesse et instants de rendez-vous estimés) [BJR04]. Un premier jeu de résultats est présenté.

Enfin, une conclusion et des perspectives sont présentées dans le **chapitre 8**.

Chapitre 2

État de l'art des méthodes de localisation sous-marines

La navigation est un ensemble de techniques qui permettent à un véhicule de se situer dans un environnement et de s'y déplacer. Nous nous pencherons plus particulièrement sur le cas de véhicules autonomes, c'est-à-dire de véhicules qui se déplacent sans l'intervention d'un opérateur dans des environnements connus ou inconnus et dans lesquels ils doivent effectuer des tâches. Ils sont utilisés par exemple pour l'exploration planétaire ou sous-marine ou dans le cadre d'une intervention à caractère industriel. Notons que nous ne traiterons ici que de localisation sous-marine civile, et non militaire.

La problématique de la navigation en environnement naturel inconnu peut se décomposer en quatre axes principaux : la perception de l'environnement, la localisation du robot dans son environnement, la génération de trajectoires permettant de contourner les obstacles, de franchir des zones accidentées ou d'atteindre un lieu donné, enfin, l'exécution de la tâche définie.

Dans ce chapitre, nous nous attacherons aux deux premières problématiques : la **perception** et surtout la **localisation** dans un environnement particulier qui est le milieu sous-marin.

Tout d'abord, nous présenterons brièvement le milieu sous-marin et les contraintes qui y sont liées. Puis les différents types de navigation en milieu sous-marin, ainsi que les capteurs qui y sont associés seront exposés. Enfin, nous donnerons un aperçu plus général de la navigation, en milieu terrestre et aérien.

2.1 Le milieu sous-marin et ses spécificités

En milieu sous-marin, les ondes électromagnétiques ne peuvent pas être utilisées car elles sont très rapidement atténuées après pénétration dans l'eau.

Le son et la lumière ont des comportements très différents, bien que tout deux puissent être considérés comme des ondes. La lumière, composée d'énergie électromagnétique, se propage mieux dans le vide, et en général de moins en moins bien dans les milieux de densité croissante. En revanche, le son est constitué d'énergie acoustique due aux vibrations du milieu parcouru. Ainsi, le son se propage mieux dans les solides

et les liquides que dans les gaz et pas du tout dans le vide. Plus particulièrement en milieu sous-marin, l'atténuation due à l'absorption et à la diffusion est beaucoup plus faible que pour les ondes électromagnétiques [Hil62]. Les ondes acoustiques constituent donc le meilleur moyen de transmission sous l'eau.

Les longueurs d'onde λ couramment rencontrées dans l'océan s'étendent du millimètre à environ 50 mètres. Or la célérité du son c dans l'eau est approximativement égale à $1500 \text{ m}\cdot\text{s}^{-1}$. D'après la relation $f = \frac{c}{\lambda}$, les fréquences des ondes acoustiques en milieu sous-marin s'étendent donc de 30 Hz à 1,5 MHz¹.

Remarque : Certains capteurs comme le GPS sont donc utilisés seulement lors de la remontée de l'engin en surface (voir une brève description dans la partie 2.3.1). Par grands fonds, ils sont inutilisables.

2.2 La navigation sous-marine

Dans la partie suivante, après avoir brièvement présenté quelques généralités, les méthodes de positionnement d'un engin sous-marin les plus courantes ainsi que les capteurs employés dans chaque cas seront exposés.

En milieu sous-marin, ce milieu sans cesse en mouvement, plein d'imprévus, une bonne connaissance de la localisation d'un véhicule est indispensable lorsqu'on veut imposer des tâches au véhicule.

Les véhicules sous-marins sont conçus en fonction de leurs domaines d'application. Les critères de choix pour définir le type, la forme et les dimensions d'un engin peuvent être par exemple la capacité de manipulation, la capacité d'emport d'une charge utile, d'appareils de prélèvement, l'autonomie énergétique, la profondeur du travail. Il existe trois grands types d'engins sous-marins :

- les ROV (Remotely Operated Vehicle) sont reliés à la surface par un câble les alimentant en énergie et transmettant des données. Ces engins sont en général équipés de bras manipulateurs et peuvent embarquer du matériel de mesure. Ils sont très présents lors de tâches d'inspection et de télémanipulation. Aussi, l'information visuelle est très importante pour le pilotage de tels véhicules. Un exemple de ROV est présenté figure 2.1(a).
- les AUV (Autonomous Underwater Vehicle) sont des engins autonomes, préprogrammés pour effectuer une mission. Ils embarquent l'énergie nécessaire à leur mission. Ces engins sont en général spécialisés pour des tâches simples comme la cartographie d'une zone.
- les engins habités sont des sous-marins de petite taille, pouvant embarquer deux à trois personnes. Ils ont comme l'AUV une autonomie énergétique, mais le pilote embarqué dans l'engin est chargé de la navigation. Un exemple est présenté figure 2.1(b).

L'ensemble des capteurs embarqués doit permettre de déterminer les grandeurs suivantes :

- La position : latitude, longitude, immersion,
- l'attitude : cap, roulis, tangage,
- les vitesses et accélérations du véhicule selon les trois axes,
- les vitesses et accélérations angulaires autour des trois axes.

Les différents types de navigation qui vont être exposés dans la suite permettent soit d'estimer les paramètres par rapport à l'instant de mesure précédent (*navigation relative* comme les navigations à l'estime ou inertielle), soit d'estimer les paramètres, indépendamment, à tout instant, par rapport à un repère fixe (*na-*

¹NB : la limite audible pour l'homme étant de 20 Khz



(a) Un exemple d'engin téléopéré : le ROV Victor 6000

(b) Un engin sous-marin habité : le Nautilus

FIG. 2.1 – Deux exemples d'engins sous-marins utilisés à l'Ifremer.

vigation absolue comme la navigation acoustique). Les trois méthodes de navigation principales en milieu sous-marin sont la navigation incrémentale (navigation à l'estime ou inertielle), la navigation acoustique et la navigation géophysique [LBSF98].

2.2.1 Navigation à l'estime

La navigation relative, dite à l'estime, est basée sur la connaissance de l'attitude du véhicule (cap, tangage, roulis) dans un repère de référence géographique ou géodésique lié à la terre et des déplacements relatifs du véhicule par intégration de mesures de vitesses.

Voici représenté sur la figure 2.2 le principe de calcul de la navigation à l'estime.

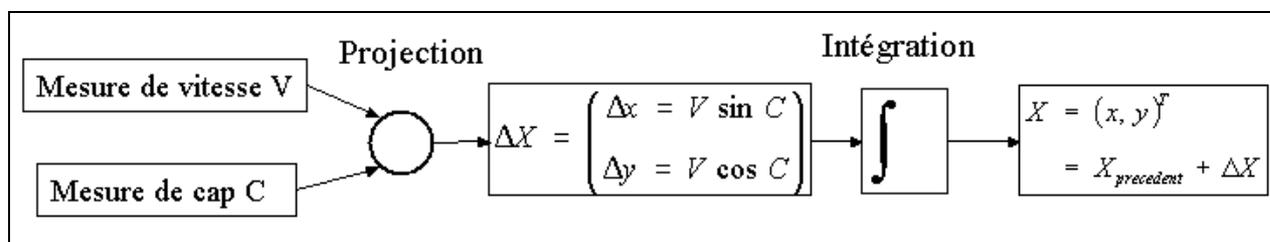


FIG. 2.2 – Navigation à l'estime

En navigation sous-marine, les mesures d'attitude et de vitesse sont acquises par resp. les gyroscopes ou gyrocompas et les lochs.

2.2.1.1 Capteurs d'attitude

Les capteurs d'attitude permettent d'estimer l'orientation, c.-à-d. le cap, le tangage et le roulis du véhicule.

Le compas magnétique : Cet appareil mesure la direction du champ magnétique local. On distingue les boussoles classiques et les compas flux gate. Ces derniers mesurent la différence de potentiel générée par le champ magnétique local aux bornes d'une bobine parcourue par un champ électrique. En théorie, ces compas mesurent la direction de la composante horizontale du champ magnétique terrestre, mais ils sont perturbés par les variations locales du champ magnétique d'origine naturelle ou issues des perturbations électromagnétiques de l'engin. Enfin, la mesure peut être perturbée par les accélérations de l'engin dans le cas des capteurs les plus simples. En général la résolution de ces instruments est du dixième de degré mais leur précision est rarement supérieure au degré.

Le gyrocompas classique : Cet appareil permet la mesure du cap par rapport au nord géodésique à l'aide d'un gyromètre à deux degrés de liberté, monté sur une plate forme pendulaire suspendue. Le gyrocompas chercheur de Nord est un système équivalent, qui par identification de la verticale locale à l'aide d'un inclinomètre, permet d'estimer le cap par rapport au Nord géographique. Les principaux paramètres affectant la mesure des gyrocompas sont la latitude, la déclinaison et la connaissance de la verticale locale.

2.2.1.2 Capteurs de vitesse

Les Lochs mécaniques et électromagnétiques : Les **lochs mécaniques** sont de simples roues à godets ou capteurs de pression. Les **lochs électromagnétiques** utilisent le principe de Faraday, qui énonce que le déplacement de l'eau de mer (assimilée à un conducteur électrique) à travers un champ magnétique génère un courant induit proportionnel à sa vitesse de déplacement. Les lochs mécaniques et électromagnétiques mesurent la *vitesse que par rapport à l'eau*.

Les Lochs acoustiques : Le loch Doppler mesure les *vitesse de l'engin par rapport au fond ou à la colonne d'eau*. Le principe consiste à éclairer le fond avec un écho-sondeur à une fréquence donnée, alors que le véhicule se déplace à la vitesse v . La différence de fréquence entre l'émission et la réception au niveau du sondeur est affectée par l'effet Doppler et elle est proportionnelle à la célérité du son dans l'eau et à la vitesse de déplacement suivant l'axe acoustique. On a $\Delta f = f_r - f_t \simeq \pm \frac{2v f_t}{c}$ où c est la vitesse du son dans l'eau, f_t est la fréquence transmise, f_r la fréquence reçue, v est la vitesse [Bro94].

Dans le cas de la configuration la plus classique dite de Janus, quatre écho-sondeurs, positionnés sur une couronne éclairant le fond ou les couches d'eau vont émettre simultanément un signal de même fréquence. En tenant compte de la mesure d'attitude du loch (donc de l'engin), il est possible de mesurer les vitesses de l'engin. Le loch Doppler fonctionne relativement bien sur terrain plat (de un à quelques % de la distance parcourue) mais ses performances se dégradent assez rapidement sur fonds tourmentés. Par ailleurs, la précision de la mesure dépend directement de la connaissance de la salinité et de l'attitude et sa résolution à faible vitesse n'est pas excellente.

Le loch Sonar à corrélation est constitué d'un émetteur Sonar dirigé vers le fond et d'une antenne matricielle d'hydrophones de réception. Le principe de mesure est que la structure du signal reçu par un élément avant de l'antenne se retrouve, lors de l'avancée du porteur, reçue par un élément arrière, avec un retard fonction de la vitesse de déplacement et de l'écartement des deux capteurs. La comparaison des pics de corrélation d'une image acquise par l'antenne à l'autre permet une estimation de ce retard, qui est utilisé pour le calcul de la vitesse. Le traitement de corrélation est en fait idéalement réalisé sur la transformée de Fourier rapide du signal de réverbération du fonds sous-marin. Ce système utilise des fréquences plus basses que les lochs Doppler décrits précédemment, ce qui autorise l'emploi d'un unique transducteur, avec une ouverture de faisceau et une portée plus importante. Les lochs à corrélation sont intrinsèquement plus précis que les lochs Doppler à faible vitesse mais il n'existe à ce jour aucun produit opérationnel embarquable sur robot mobile sous-marin. Enfin sur fonds très uniformes la corrélation est moins précise.

Les principaux avantages des lochs Sonar et Doppler comparés aux lochs conventionnels sont :

- La possibilité de travailler par rapport au fond (Bottom track) ou par rapport aux couches d'eau (Water track)
- Une meilleure précision sur la mesure de la vitesse (surtout lorsque celle-ci est mesurée par rapport au fond)
- La possibilité de mesurer les trois composantes de vitesse.

Les principaux inconvénients de ces systèmes sont :

- La sensibilité à des changements significatifs des pentes du fond marin
- La sensibilité à des changements soudains dans les courants locaux lorsque le loch travaille en water track
- Les problèmes de compatibilité acoustiques avec d'autres senseurs du robot, sonars et sondeurs.

2.2.1.3 Tableau récapitulatif des capteurs employés

Plusieurs types de capteurs sont nécessaires pour une navigation à l'estime. Dans le tableau 2.1 sont donnés quelques paramètres pour chaque capteur employé.

| Capteur | Type de mesures | Précision | Récurrence |
|---------------|--|-----------|------------|
| Loch-Doppler | Vitesses par rapport au fond marin (m/s) | 5mm/s | 0.5 s |
| Gyrocompas | Attitude de l'engin (degrés) | 0.1° | 0.1s |
| Profondimètre | Immersion de l'engin (m) | 1 mètre | 0.2 s |
| Altimètre | Altitude de l'engin (m) | 10 cm | 0.5 s |

TAB. 2.1 – Paramètres des capteurs employés dans la navigation à l'estime

2.2.1.4 Limitations de la navigation à l'estime

Par son principe, la navigation à l'estime intègre des mesures bruitées. A court terme, la méthode est très fiable. Mais malheureusement, ce phénomène induit un biais sur le calcul de position, qui croît avec le temps. Actuellement, une solution proposée à cette dérive de navigation est de recalculer l'estime à intervalles

réguliers, soit par une mesure GPS obtenue en remontant à la surface, soit par une mesure de position absolue (par exemple par mesures de base ultracourte - voir 2.2.3.3). Cependant, chaque solution présente un inconvénient. Dans le premier cas, la remontée à la surface représente un coût opérationnel non négligeable et n'est pas toujours possible ou sans danger. Dans le second cas, les mesures de position absolue sont assez souvent des données aberrantes. Il suffirait d'acquérir une donnée aberrante pour que le recalage soit effectué à un point encore plus éloigné de la position actuelle qu'avec la dérive de l'estime.

2.2.2 Navigation inertielle

Les systèmes inertiels sont basés sur l'association des capteurs d'attitude et de capteurs accélérométriques. Cette navigation est également une navigation relative, puisqu'elle repose sur la double intégration de caps et d'accélération. La navigation inertielle repose sur une combinaison de mesures provenant de gyroscopes, qui mesurent le taux de rotation, et d'accéléromètres qui mesurent l'accélération. Il existe également des systèmes intégrant tous ces capteurs : les centrales inertielles.

Plus généralement, les systèmes de navigation inertielle fonctionnent bien lors d'un régime de navigation à vitesse et accélération assez élevées, ce qui n'est pas souvent le cas, sauf dans le domaine aérien.

2.2.2.1 Capteurs d'accélération

L'accéléromètre consiste à mesurer la force massique appliquée à une masse pour la maintenir en place. L'effort consenti pour cet asservissement dépend de l'accélération suivant l'axe de mesure. Cette mesure est effectuée le long des trois axes du véhicule.

2.2.2.2 Capteurs de vitesse angulaire

Les gyroscopes, gyromètres et gyroscopes optiques permettent de mesurer une rotation instantanée selon l'un des trois axes.

Le gyroscope mécanique classique permet de mesurer les composantes d'un vecteur lié à l'axe d'une toupie en rotation dont l'extrémité est solidaire d'un boîtier soumis au mouvement de l'engin. L'angle de déviation de l'axe toupie par rapport aux axes du boîtier est proportionnel au mouvement angulaire dans le repère inertiel engin. Sur ce principe, un **gyromètre** mesure les **composantes de rotation instantanées**, en asservissant l'axe de la toupie à l'aide d'un moteur couple (par exemple) de manière à toujours ramener l'axe de la toupie à une position de repos par rapport au repère du boîtier. La mécanisation de ces capteurs, les contraintes de maintenance liées aux parties mobiles et leurs coûts importants sont des inconvénients, mais en général la précision de ces systèmes est très bonne (d'une dizaine à une centaine de degrés à 1σ).

La technologie des **gyroscopes optiques** a permis de réaliser de grands progrès en précision et en fiabilité. Le gyroscope de type ou **gyrolaser** est un capteur à composants liés (*i.e.* sans partie mobile). Il est constitué d'un jeu de prismes matérialisant un chemin optique pour un faisceau laser autour d'un axe de rotation matérialisé par le centre du chemin optique. Le principe consiste à mesurer, à l'aide d'une horloge et d'un récepteur performant, le temps de vol correspondant au chemin optique, qui est physiquement raccourci ou allongé suivant la *vitesse* et le centre de rotation autour de l'axe sensible. Ces capteurs sont très

performants mais très coûteux.

La démocratisation de la fibre optique est à l'origine du **gyromètre à fibre optique**. Dans ce cas, le chemin optique est matérialisé par une fibre optique enroulée autour de l'axe sensible de mesure. Cette fibre peut mesurer plusieurs centaines de mètres sur des rayons de quelques centimètres. Le principe est alors de mesurer le déphasage du signal optique à chaque extrémité de la fibre pour estimer la *vitesse de rotation* du gyroscope. Ces senseurs sont bon marché et permettent d'obtenir de bonnes précisions (inférieures au dixième de degré à 1σ).

Les capteurs d'accélération et de vitesse angulaire sont associés à un capteur qui mesure l'attitude du véhicule.

2.2.2.3 La centrale inertielle

Cet appareillage est constitué de trois accéléromètres et trois gyroscopes. Il peut se présenter de deux façons : la plate-forme compensée ou à composants liés. Le premier système permet d'isoler la centrale du véhicule, de façon à ce qu'elle puisse être orientée indépendamment de l'engin. Le deuxième système est fixé à l'engin, ce qui fait que les accélérations et vitesses angulaires sont calculées par rapport à la structure du véhicule. La centrale inertielle est un système complet, autonome, qui, si elle est fiable, peut suffire à la navigation d'un véhicule. Malheureusement, un tel système est très encombrant et très coûteux.

Deux inconvénients majeurs dans ces systèmes de navigation sont d'une part, la grande sensibilité à l'inclinaison du véhicule (donc à la régularité du sol), d'autre part, au fait que les accéléromètres fournissent un rapport signal sur bruit très faible dans les mouvements de faible accélération, qui constitue la majorité des cas. De plus, un facteur limitant est un coût encore très élevé pour ces systèmes de navigation. Cette méthode de navigation est de ce fait, encore peu utilisée.

Comme expliqué, les systèmes de navigation inertielle fonctionnent bien lors d'un régime de navigation à vitesse et accélération assez élevées. Malheureusement, ce n'est majoritairement pas le cas dans le milieu sous-marin.

Comme la navigation à l'estime, l'erreur sur l'estimation de la position augmente avec le temps. Il convient donc de recalculer de temps en temps les positions estimées.

Pour obtenir la position du mobile à partir d'une navigation inertielle, une ou deux intégrations sont nécessaires (une pour le taux de rotation, deux pour l'accélération). Elles sont la source d'une dérive sur la localisation du robot, puisque l'erreur sur les mesures est également intégrée.

D'autre part, ces systèmes sont très onéreux et consomment beaucoup d'énergie. Même si ces deux contraintes tendent à être réduites, les systèmes de navigation inertielle sont encore peu employés.

2.2.3 Navigation acoustique

La navigation acoustique est très largement répandue en milieu sous-marin. Elle permet de connaître à chaque instant la position absolue de l'engin sous-marin, sans avoir la contrainte de remonter à la surface.

Cette navigation repose sur la communication acoustique entre un réseau de balises dont les positions sont connues dans le repère géographique et le véhicule sous-marin. Les deux principaux types de navigation acoustique, la base longue (BL) et la base ultracourte (BUC), sont présentés.

2.2.3.1 Bases longues (BL)

Le principe de la base longue (voir figure 2.3) correspond au positionnement du robot sous-marin par rapport à un champ de transpondeurs (balises acoustiques), préalablement mouillés et dont la position est calibrée par rapport à la position du navire en surface. Le robot navigant au milieu de ce champ émet un signal d'interrogation acoustique. Chaque transpondeur répond sur une fréquence qui lui est propre. Connaissant la mesure du temps de vol écoulé entre l'émission et la réception du signal retour et la célérité de l'onde acoustique dans le milieu, il est possible d'évaluer la distance par rapport aux transpondeurs. Par multi-latération (intersection géométrique des cercles de distances centrées sur chaque balise), il est possible de déterminer précisément la position de l'engin. En général, une bonne multi-latération requiert au moins trois balises, on parle alors de trilatération. Il est cependant possible de ne déployer que deux balises si l'engin est équipé d'un profondimètre, pour lever l'ambiguïté de l'intersection de deux sphères de trilatération. La précision d'un système base longue dépend en particulier :

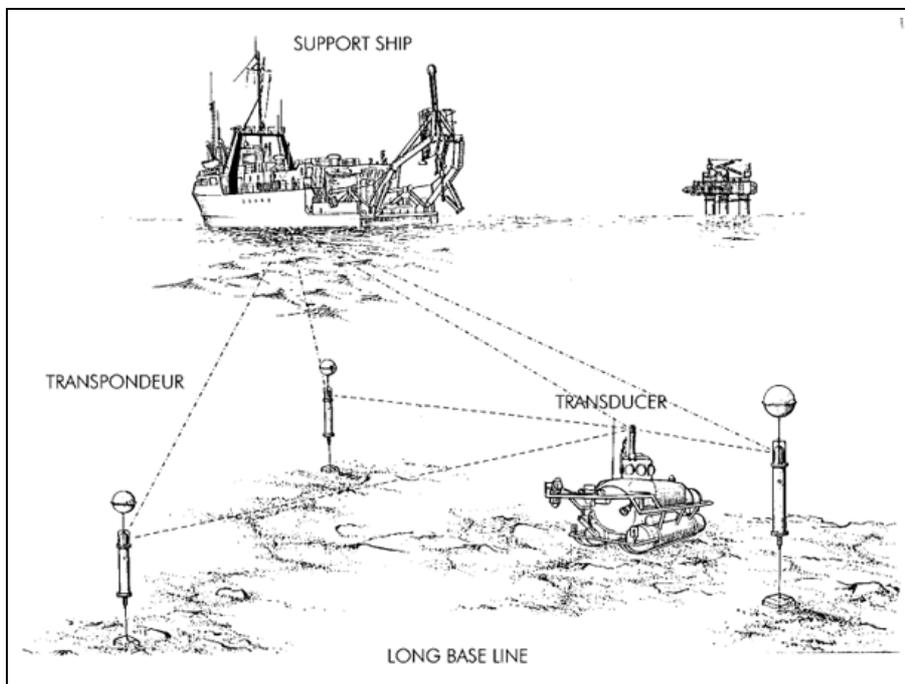


FIG. 2.3 – Principe d'une base longue

- du nombre de transpondeurs
- de la géométrie du champ de balises et des masquages éventuels par le relief
- de la qualité de calibration du champ
- de la connaissance des profils de célérité sur la colonne d'eau.

Suivant les fréquences d'interrogation et de réponse des transpondeurs, engin, navire ou balise, la portée en distancemétrie limite la taille du champ. Le principal problème opérationnel est la mise en place, la récupération et la calibration des balises. La position du robot peut être connue soit à bord du sous-marin, soit en surface via une télémétrie par le câble ou via une télétransmission acoustique suivant les types d'engin

(ROV-Remotely Operated Vehicle, SUV-Supervised Underwater Vehicle).

Il est assez courant pour des balises basses fréquences (de quelques dizaines de kHz) sur des champs de taille moyenne (1000m entre balises) d'obtenir un positionnement à 10 mètres près.

2.2.3.2 Système GIB

Le système GIB (GPS Intelligent Buoys) [BT02] repose également sur un déploiement de 4 à 12 bouées qui mesurent le temps de vol de signaux acoustiques émis par l'engin sous-marin.

A chaque cycle (typiquement 2 secondes), l'émetteur acoustique (pinger) émet 2 signaux successifs, l'un synchrone au temps GPS et l'autre qui sera retardé proportionnellement à la profondeur.

Contrairement aux bouées d'une base longue, qui sont calibrées par rapport à la position du bateau, chaque bouée du système GIB transmet au PC installé sur le bateau ces 2 signaux ainsi que sa propre position D-GPS. Comme dans le cas d'une base longue, un système de triangulation est alors mis en place pour déterminer avec précision la position de l'engin sous-marin.

Ce système a une précision métrique voire submétrique.

2.2.3.3 Bases ultracourtes (BUC)

La BUC est composée d'une antenne acoustique à plusieurs éléments, positionnée sous la coque du navire ainsi que d'un pinger équipant le robot.

Le principe de cette navigation (voir figure 2.4) réside dans la mesure de la différence de phase en réception d'un signal acoustique émis par le robot sous-marin sur les éléments de l'antenne acoustique. Pour ce faire, il est nécessaire de connaître l'attitude du navire avec une très grande précision. En mode pinger, la mesure de la différence de phase permet de connaître l'angle de la ligne engin/navire par rapport à une référence de verticale. En mode transpondeur, il est possible de mesurer la distance à la source et l'angle d'incidence de l'onde acoustique en estimant le temps de vol engin/navire. Quand un câble permet de dialoguer avec l'engin depuis la surface, l'interrogation de l'émetteur engin par câble permet d'estimer la distance à la source et l'angle d'incidence de l'onde acoustique. On parle alors de mode répondeur. Il est possible d'augmenter la précision des deux précédents modes en utilisant une mesure de profondeur par profondimètre (voir partie 2.2.3.6).

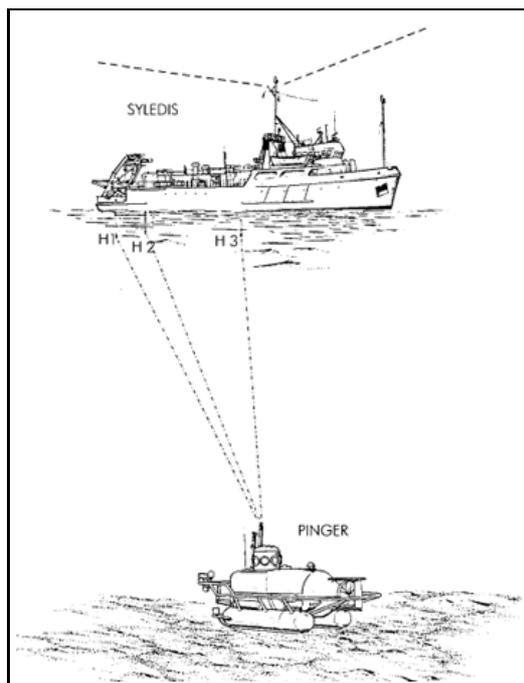


FIG. 2.4 – Principe d'une base ultra courte

2.2.3.4 Tableau récapitulatif

Sur le tableau 2.2 sont regroupés quelques paramètres de la navigation absolue en milieu marin.

| Méthode | Précision | Contraintes de déploiement en surface | Contraintes de déploiement sur le fond | Champ | Cadence |
|-------------------|--|---|--|--|--|
| Base longue | Répétabilité de 2 à 6 mètres à 1σ | Acoustique intégrable en poisson | 3 balises (éventuellement 2) | Égal à la hauteur d'eau | 10 à 40 s selon les modes et la profondeur |
| Base ultra-courte | Précision de 0.5% de la hauteur d'eau à 1σ dans un cône de $\pm 30\%$ en mode de base | Installation de l'antenne sous la coque. Disponibilité immédiate sur site | Sans balise en mode de base | Dans un angle de $\pm 45^\circ$ à la verticale du navire | 8 s |

TAB. 2.2 – Quelques paramètres pour les méthodes de navigation acoustique

2.2.3.5 Limitations du système

Des erreurs peuvent survenir à plusieurs niveaux. Tout d'abord, il faut avoir une connaissance du profil de célérité du son. D'autre part, les retards de réponse des balises doivent être connus. Enfin, les positions

des balises doivent être connues parfaitement.

L'utilisation de balises limite la zone de mesure, pour les BUC comme pour les BL (géométrie du champ de balises et portée des balises). D'autre part, dans le cas des BL, les contraintes opérationnelles lors du déploiement et de la récupération des balises ne sont pas négligeables.

Deux sources d'erreur existent en navigation acoustique. Il s'agit d'une part, d'une erreur de géométrie du système, d'autre part d'une erreur sur la célérité du son.

La première erreur peut être réduite en calibrant le système. Par exemple, Opderbecke propose dans [Opd97] une méthode de calibrage d'un système de base ultracourte. Il s'agit d'aligner au mieux les capteurs en utilisant un transpondeur fixe placé au fond de l'eau. Une trajectoire quelconque est effectuée au-dessus du transpondeur et enregistrée. Les non-alignements des émetteurs sont estimés par un filtre de Kalman.

La deuxième source d'erreur, sur la vitesse supposée du son, peut être réduite en accroissant la connaissance du profil de célérité du son. Deffenbaugh [DSB96] a mis au point une technique pour les bases longues, qui considère toutes les impulsions reçues (et non la première seulement) pour retrouver les variations de célérité du son.

2.2.3.6 Mesures d'immersion ou de profondeur

Dans la navigation à l'estime, d'autres capteurs permettent d'obtenir des mesures absolues de profondeur ou d'immersion. C'est le cas des profondimètres, qui mesurent l'immersion de l'engin, et des altimètres, qui mesurent l'altitude de l'engin par rapport au fond marin.

Ces capteurs reposent sur des jauges de contraintes ou des capteurs piézoélectriques, plus précis, qui mesurent la variation de vibration d'un quartz en fonction de la pression qui s'exerce sur celui-ci. Ces capteurs sont en général très précis (1 /100 à 1/1000 de la hauteur d'eau).

2.2.4 Comparatif des différentes navigations en milieu sous-marin

Les méthodes de navigation en milieu sous-marin ainsi que les principaux capteurs associés ont été présentés précédemment. Le tableau 2.3 regroupe les avantages et les inconvénients de chaque méthode.

| Méthode | Mesures | Avantages | Inconvénients |
|------------|---|---|---|
| Estime | Cap et vitesse | Fréquence d'acquisition importante, précis à court terme | Dérive qui augmente avec le temps, nécessité de recalibrer |
| Inertiel | Accélération, attitude, vitesse angulaire | Fréquence d'acquisition importante, bonne précision à court terme | Dérive, encombrant, coût élevé, mauvais fonctionnement à faibles vitesses et accélérations |
| Acoustique | Position | Position absolue | Calibrage important, données aberrantes, contraintes de déploiement, faible fréquence d'acquisition |

TAB. 2.3 – Tableau comparatif des méthodes de navigation en milieu sous-marin

Une hybridation entre estime et positionnement acoustique permet d'optimiser la localisation sous-marine. En général, le positionnement acoustique est connu en surface, avec une référence correspondant au meilleur positionnement (GPS) du navire. L'estime est un moyen de connaître sa position à court terme et à haute cadence. Dans le cas des engins libres, c'est le positionnement acoustique fond (relatif au champ de balises) qui est connu de l'engin et qui est hybridé avec le positionnement à l'estime. Dans le cas des engins à câble, l'hybridation entre estime et positionnement acoustique de surface est également possible.

La navigation intégrée est donc une façon de compenser la dérive obtenue par la navigation à l'estime en recalant ponctuellement la position grâce à la navigation absolue.

2.2.5 Navigation géophysique

Pour quelques applications des véhicules autonomes, la navigation par champ de balises ne peut pas être envisagée. Une alternative à cette navigation absolue est la navigation géophysique. Elle s'appuie sur un repérage dans une carte de l'environnement déjà définie. Ainsi, en acquérant des mesures de paramètres géophysiques, comme les relevés bathymétriques, du champ magnétique ou des anomalies de gravitation, on peut mettre en relation ces données avec la carte préalablement construite.

Les caméras embarquées sur un véhicule sous-marin sont soit fixées verticalement, soit fixées latéralement. Elles sont insérées dans des enceintes qui permettent de corriger les erreurs de distorsion (voir chapitre sur les images). Ces caméras sont de type CCD.

Dans [RRCC03], est décrit un système permettant à un robot sous-marin de naviguer dans un environnement connu à partir d'images acquises par une caméra verticale fixée sur l'engin. Un réservoir d'eau, cylindre de 4.5 mètres de diamètre, dans lequel l'engin se déplace, présente un fond composé de cercles de couleurs différentes, avec de temps en temps des trous dont la position absolue est connue. La position et la vitesse de l'engin sont estimées par un filtrage de Kalman étendu, qui repose sur la suivi de ces cercles et le recalage de la position lorsqu'un trou est détecté.

Les méthodes opérationnelles utilisent la mise en correspondance de cartes par corrélation pour éviter un coût algorithmique trop important.

Dans sa thèse [Sis00], Sistiaga a proposé une méthode complète et originale pour recalibrer une carte locale construite à partir d'images vidéo ou de profondeur (image de bathymétrie), sur une carte globale construite au préalable. Une étape d'extraction de points est réalisée, puis un algorithme itératif (reposant sur les moindres carrés pondérés) de mise en correspondance des points détectés est mis en jeu, pour trouver la transformation entre la carte locale et la carte globale. Cette méthode sera détaillée puis utilisée dans la partie 5.3.1.

Néanmoins, cette méthode repose sur la construction d'une carte globale a priori. Les deux problèmes principaux sont la difficulté à générer cette carte et à la stocker (car beaucoup de données), et à rechercher le pic de corrélation lors de la mise en correspondance de la carte locale avec la carte globale. Par exemple, Lucido et al. [LPOR96] ont simplifié le problème en étudiant la segmentation et la mise en correspondance de profils bathymétriques.

Cependant, dans la pratique, une carte globale incluant la zone d'exploration de l'engin est très rarement accessible. Ainsi, le même problème que pour le milieu terrestre revient : il s'agit du problème de construire

une carte et de se localiser simultanément (SLAM, voir partie 2.3.3.2).

Le SLAM s'est développé très récemment en navigation sous-marine. Par exemple, [NLR03] propose un SLAM basé sur des caractéristiques extraites d'images sonar (plus précisément, d'images acquises par un sonar à ouverture synthétique).

2.2.6 Fusion en milieu sous-marin

La fusion de plusieurs capteurs peut être nécessaire pour améliorer la qualité de la position de l'engin. Pour pouvoir fusionner les données de différents capteurs, une phase de calibration géométrique in situ et d'alignement des capteurs les uns par rapport aux autres est nécessaire. En effet, les instruments de mesure ne sont en général pas installés au même endroit sur l'engin. Par exemple, les auteurs de [KW02] proposent une méthode reposant sur une estimation des moindres carrés pour déterminer la géométrie d'un système BL, loch-Doppler et gyroscope. [GHM91] propose une méthode de calibration d'un système de navigation inertielle (combinaison de gyroscopes et accéléromètres), reposant sur un filtrage de Kalman.

Après avoir réalisé cette calibration, la fusion elle-même peut alors avoir lieu.

Whitcomb, dans [WYS99] propose une amélioration de la navigation basée sur une base longue seule ou un sonar Doppler (mesure de vitesse selon 4 faisceaux) seul, en fusionnant ces deux systèmes à partir d'un filtrage linéaire en fréquence (filtre passe-bas pour la base longue, filtre passe-haut pour les données de vitesse).

2.2.6.1 Fusion estime/BUC

Plusieurs algorithmes ont été développés pour localiser l'engin sous-marin, en combinant les avantages de la navigation acoustique et de la navigation à l'estime. Le principal avantage de la navigation acoustique est son positionnement absolu, en l'absence de GPS. Malheureusement, le fait même d'utiliser des ondes acoustiques induit des erreurs et des données aberrantes car les ondes peuvent être réfléchies plusieurs fois avant d'être captées. D'autre part, les données de position n'ont pas une fréquence très élevée. En ce qui concerne la navigation à l'estime, les données de cap et de vitesse sont relativement précises, comportent peu de points aberrants et surtout sont très fréquentes. Malheureusement, les bruits induisent tout de même un biais sur la trajectoire. Le fait de combiner la navigation acoustique à la navigation à l'estime permettrait donc d'avoir un positionnement de l'engin très fréquent et avec une précision améliorée. Dans l'article [VLB96], les auteurs proposent une méthode reposant sur le filtre de Kalman, qui permet de partir à l'estime et de réactualiser la position avec la navigation acoustique. D'autre part, ils rejettent les points aberrants.

Jouffroy [JO04] propose une méthode combinant les données provenant d'un système de navigation à l'estime et des mesures BUC, reposant sur l'équation de diffusion 2.1. Soit \hat{X} l'estimée de la trajectoire. Elle doit minimiser l'intégrale suivante (équation de diffusion) :

$$\int_{t_b}^{t_e} \left(\nabla \hat{X} - V_{dop} \right)^2 + k(t) \left(\hat{X} - X_{ac} \right)^2 dt \quad (2.1)$$

où V_{dop} est le vecteur vitesse du véhicule, donné par le système gyro-Doppler et X_{ac} est le vecteur position d'un système de mesure acoustique. $k(t)$ est une fonction ajustable. Cette intégrale signifie que la trajectoire

estimée doit être un compromis entre d'une part, la proximité entre \hat{X} et les positions mesurées d'une part, et d'autre part, la proximité entre la dérivée de \hat{X} et les vitesses mesurées. Ce critère est minimisé grâce à des techniques basées sur la PDE (Partial Differential Equations ou équations aux dérivées partielles), où l'intégrale est la solution obtenue par la méthode de descente du gradient suivante :

$$\frac{\partial \hat{X}}{\partial s} = \nabla \left(\nabla \hat{X} - V_{dop} \right) - k(t) \left(\hat{X} - X_{ac} \right) \quad (2.2)$$

Cette approche, pour laquelle des contraintes peuvent être ajoutées sous la forme de fonctions non-linéaires, donne de bons résultats sur des données réelles provenant d'une mission du ROV Victor 6000. La figure 2.5 illustre le résultat d'une fusion de données gyro-Doppler (en trait plein gris clair) et de mesures de position d'une base ultracourte (en pointillés). Le résultat est affiché en trait plein noir. Le filtre PDE montre un résultat présentant à la fois une bonne position absolue (données BUC) et une bonne précision incrémentale (données gyro-Doppler).

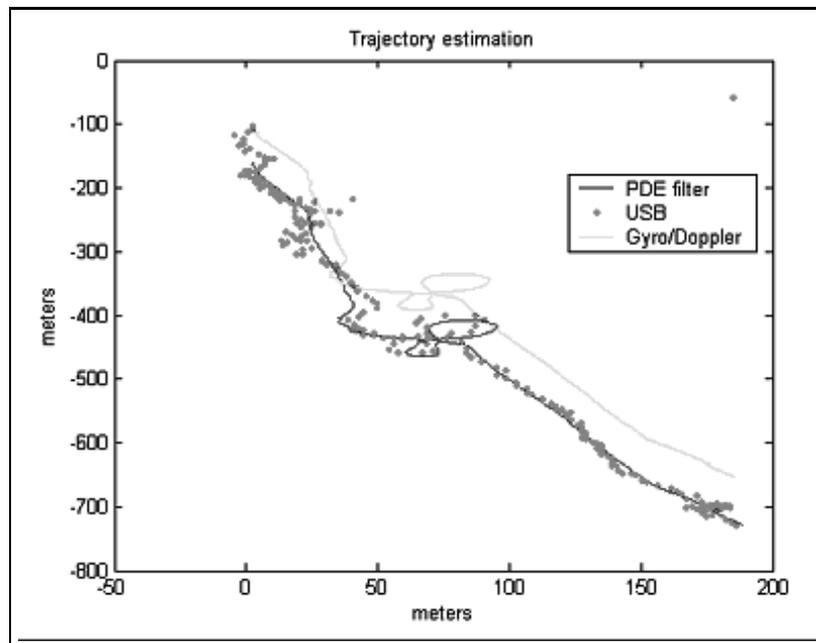


FIG. 2.5 – Résultats obtenus par la méthode de Jouffroy [JO04]

Dans notre problématique, la localisation sera effectuée à partir de données d'estime et d'images vidéo.

2.3 La problématique plus générale de localisation

La localisation d'un robot mobile permet de positionner ce robot dans son environnement, soit par rapport à un repère absolu, fixe dans le temps, soit par rapport à un repère relatif, mobile, lié au robot. On peut donc classer les capteurs en deux catégories : les capteurs extéroceptifs et les capteurs proprioceptifs. Un panorama des capteurs actuellement utilisés ou en phase d'études peut être trouvé dans [BEFW97] [Maj01]. Dans cette partie, le fonctionnement des capteurs les plus utilisés sera brièvement décrit. En milieu terrestre et aérien, des données sur l'environnement sont souvent déjà disponibles. D'autre part, ces environnements sont moins en mouvement que le milieu marin.

2.3.1 Capteurs extéroceptifs

Les capteurs extéroceptifs acquièrent des informations sur l'environnement du mobile et permettent une localisation absolue. Les appareils de mesure les plus connus sont le GPS, la caméra, le sonar ou le télémètre.

La localisation par champ de balises est essentiellement utilisée en navigation maritime et aérienne. Deux types de systèmes de balises existent : les balises actives et les balises passives. Parmi les balises passives, on retrouve des réflecteurs radar utilisés dans la navigation des cargos, ou des réflecteurs laser utilisés dans les systèmes de navigation en milieu intérieur. Parmi les balises actives, on peut trouver les transpondeurs radio comme le LORAN ou DECCA en environnement naturel ou les méthodes acoustiques en milieu intérieur ou sous-marin. Les balises actives les plus connues sont les GPS. Cette navigation utilise au moins quatre satellites qui délivrent des signaux qui permettent de déduire la position d'un objet à partir d'une triangulation.

Les capteurs de distance les plus employés sont le sonar (ultrasons), le lidar (infrarouges) ou le radar (hautes fréquences). Ces appareils mesurent le temps de vol t d'une impulsion qu'ils ont émise : $t = \frac{2d}{v}$ (d distance, v vitesse du son ou de la lumière, connue).

Parmi ces trois types de capteurs, le télémètre à ultrason est le plus utilisé. Il fonctionne selon le principe de l'écho. Le sonar émet un "pulse" qui est réfléchi par un obstacle situé quelque part dans le cône d'observation du capteur. L'intervalle de temps entre l'émission du pulse émis et la réception du pulse réfléchi permet d'estimer de façon précise la distance de objet-sonar. Cependant, il ne permet pas de déterminer à quel angle du cône le pulse a été réfléchi.

Le lidar et le radar peuvent également déduire la distance à partir du déphasage $\phi : \frac{\phi}{2\pi} = \frac{2d}{v}$.

Il existe d'autres types de télémètres qui ont des portées ou des précisions variables, notamment le télémètre à infrarouge qui mesure l'angle de réflexion d'une onde IR modulée grâce à une rangée de capteurs et le télémètre laser.

Les caméras vidéo sont employées dans grand nombre de problèmes et permettent de tirer des renseignements de l'environnement qui entoure le robot. Ces renseignements peuvent être des caractéristiques du milieu (droites, points d'intérêt) ou des informations de profondeur, si deux caméras sont montées en stéréo.

2.3.2 Capteurs proprioceptifs

Les capteurs proprioceptifs acquièrent des informations sur le robot lui-même, et permettent une localisation relative, dans un repère lié au robot. L'odométrie est une technique très répandue, qui estime la distance parcourue en comptant le nombre de tours de roue du véhicule. Elle permet une bonne précision de la localisation à court terme, fournit des données à une fréquence élevée et est très peu chère. Cependant, une telle technique comporte des erreurs : des erreurs systématiques, mesurables (imperfections du robot, par exemple, des diamètres de roues non égaux) et des erreurs non-systématiques, liées au terrain (sol glissant, bosses...). Ces erreurs sont intégrées dans le temps et donnent lieu à un biais sur l'estimation des positions. La navigation inertielle, détaillée dans le cas du milieu sous-marin, est également très utilisée en milieu terrestre ou aérien. Les capteurs de mesure sont les mêmes qu'en navigation sous-marine.

2.3.3 Quelques exemples de navigation en milieu non sous-marin

En milieu terrestre, les robots mobiles sont la plupart du temps équipés d'un système odométrique ainsi que de capteurs télémètres et de caméras. La caméra est très utilisée. En effet, il est bien plus fréquent d'observer des scènes structurées ou comportant des caractéristiques détectables (bâtiments ou routes en navigation aérienne, murs en navigation terrestre).

2.3.3.1 Navigation référencée terrain

La navigation référencée terrain consiste à repérer dans un environnement cartographié structuré (comportant des objets ou formes connus, de position connue), certains éléments identifiables. Les capteurs généralement utilisés sont des caméras vidéo ou des télémètres laser (mesurant la distance véhicule-objet).

Cette navigation est complémentaire des navigations à l'estime ou inertielle.

Son principe consiste à détecter dans l'environnement (souvent un milieu intérieur) des amers (murs, coins, segments 3D dans le cas d'un milieu intérieur) dont les positions sont connues. Ainsi, en les repérant sur une image de distances ou une image vidéo, on peut se positionner par rapport à eux.

Il est également possible de mettre en correspondance une "sous-carte" de profondeur (télémètre) ou visuelle avec une carte de l'environnement dans lequel évolue l'engin, et qui est déjà construite. Cette étape est réalisée en repérant des caractéristiques dans la sous-carte et en les mettant en correspondance avec la carte globale.

Dans [DCF⁺95], les auteurs proposent un système de navigation reposant sur des capteurs laser, caméras, odométriques et inertiels. Le robot doit construire une description de son environnement à partir de primitives géométriques (polygones, plans d'élévation, recherche d'horizons...) et planifier ses déplacements de façon itérative dans l'environnement qu'il a modélisé, pour remplir sa mission.

Afin de s'affranchir de la connaissance de la carte globale qui n'est pas toujours disponible, une autre technique, actuellement en plein développement, a été mise au point : le SLAM (Simultaneous Localisation and Map Building).

2.3.3.2 SLAM

Dans ce que nous avons présenté, les robots mobiles évoluaient dans un environnement connu ou du moins, structuré. Cependant, pour des applications récentes comme l'exploration de planètes, l'intervention dans des zones à risque, l'environnement est inconnu et non-structuré.

La cartographie de l'environnement peut être obtenue soit préalablement (et donc doit être mémorisée par le robot), soit construite au fur et à mesure des déplacements du robot. Ce dernier problème, généralement posé lors d'une navigation dans un environnement naturel, est plus connu sous le nom de SLAM (Simultaneous Localisation and Map Building). Il permet au robot à la fois de se localiser et de se déplacer, en temps réel, dans un environnement dont on ne connaît rien a priori. Le SLAM consiste à estimer conjointement la position du robot et les positions d'un ensemble de caractéristiques de l'environnement, supposés immobiles. Les différentes étapes qui composent le SLAM sont présentées figure 2.6.

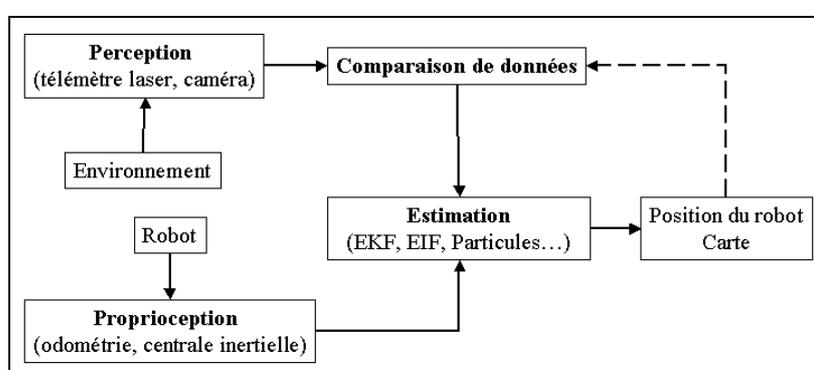


FIG. 2.6 – Étapes du SLAM

On en trouve la description suivante dans [Lem04] :

- La perception : les algorithmes de perception fournissent un ensemble d'observations des caractéristiques de l'environnement à partir des données laser, caméra, etc...
- L'association de données : prise de décision pour déterminer si une observation correspond à un amer déjà présent dans la carte, et si oui, lequel,
- Proprioception : estimation de la position du robot à partir de données proprioceptives (odométrie ou d'une centrale inertielle),
- Estimation : intégration des données de perception et de proprioception pour estimer la position du robot et des caractéristiques de l'environnement.

L'étape d'estimation met en jeu, en général, le filtrage de Kalman étendu [DPNDWC01] [Cha02] [DW02], dans lequel est créé un vecteur d'état "augmenté", c'est-à-dire un modèle où sont juxtaposés le modèle du véhicule (positions en fonctions des données odométriques) et un modèle pour chaque caractéristique du milieu. D'autres techniques d'estimation existent, comme le filtrage particulaire [MTKW03]. Néanmoins, celui-ci n'est pas appliqué pour toutes les dimensions du vecteur d'état pour des contraintes de coût algorithmique.

2.4 Fusion pour la navigation

Après avoir exposé quelques types de capteurs, nous allons décrire quelques techniques qui permettent au robot ou à l'aéronef de se repérer. Comme expliqué ci-dessus, chaque capteur a aussi bien des avantages que des inconvénients. D'autre part, il arrive très rarement qu'un seul capteur fournisse toutes les informations nécessaires à la localisation du robot et/ou à son pilotage automatique. Pour ces raisons, plusieurs capteurs sont en général combinés pour améliorer l'estimation de la position du véhicule ou donner des informations supplémentaires sur l'état du véhicule. Dans ce contexte, [VG99] fait la distinction entre trois types de fusion :

- Fusion complémentaire : les capteurs ne fournissent qu'une information partielle de l'état du système. Très souvent, un capteur proprioceptif (comme ceux utilisés dans l'odométrie) est combiné à un capteur extéroceptif (balises) [BEFW97].
- Fusion "compétitive" : les capteurs de type différent mesurent le même paramètre (exemple : un sonar et une caméra mesurant la distance d'un objet au véhicule).
- Fusion coopérative : l'un des capteurs ne mesure une quantité qu'après avoir reçu le résultat de mesure d'un autre capteur (exemple : deux caméras d'un montage stéréoscopique sont nécessaires pour estimer la dimension d'un objet situé dans leur champ commun).

2.4.1 Quelques exemples de fusion de données

En général, les robots mobiles sont conçus pour avoir un comportement autonome et réaliser des tâches [Cha95].

[VF01] propose une méthode qui permet de fusionner un système inertiel avec un système GPS, qui sont complémentaires. Une alternative à un filtre de Kalman étendu non-stable, pour le système non-linéaire (état et mesure) est décrite. La solution est stable de façon globale.

Dans [BS00], l'auteur décrit un système de navigation embarqué sur un robot terrestre, pour le pilotage de celui-ci. Deux types de fusion de capteurs sont réalisés. La première fusion consiste à intégrer des données de capteurs détectant des objets ou obstacles (montage stéréoscopique de caméras, scanner laser, radar). La deuxième fusion consiste à estimer l'état du véhicule à partir de capteurs de position ou de détection d'objets (DGPS, navigation inertielle, montage stéréoscopique + les capteurs précédents). On trouve dans son système aussi bien de la fusion compétitive que complémentaire. La première fusion est réalisée grâce au filtre de Kalman, dont le vecteur d'état regroupe la position, la vitesse, et les attributs des objets. La deuxième fusion est réalisée grâce à un algorithme minimisant un certain critère faisant intervenir les vitesses et orientations des objets détectés.

Dans [KHH95], la même thématique est reprise. Il s'agit d'estimer la position du robot pour générer un trajet et atteindre un objectif. Cependant, ici, l'environnement est connu. Les capteurs employés sont des capteurs odométriques ainsi que sonar. Les positions des objets contenus dans la scène sont connues et les objets décrits. La position du robot est calculée par une minimisation sur les positions mesurées des objets et leurs caractéristiques et répartition dans la scène, qui sont connues.

Les récentes avancées en navigation aérienne ont porté sur la reconstruction de cartes 3D du sol. Afin

d'avoir un géoréférencement² précis, plusieurs types de capteurs de position sont embarqués. Il s'agit la plupart du temps de GPS et de centrales inertielles qui sont fusionnés [GBLHT02] [KHE96] [BPDR04]. Jung et Lacroix [JL04] proposent d'établir un modèle numérique de terrain à partir de deux séquences d'images acquises par un système stéréoscopique, basé sur une mise en correspondance de points d'intérêt entre images successives.

La fusion de capteurs a été très développée et est maintenant très utilisée pour les milieux terrestre et aérien. En milieu sous-marin, la fusion n'en est pas à un stade aussi avancé, néanmoins, la recherche progresse.

²Géoréférencement = rattachement de données à des coordonnées géographiques

Chapitre 3

Localisation par mesures de cap et vitesse et instants de rendez-vous estimés

Après avoir présenté la problématique générale de navigation en environnement sous-marin ou plus généralement terrestre et aérien, nous allons décrire notre système, les mesures disponibles ainsi que les caractéristiques de notre champ d'exploration et ce qu'on peut en déduire.

Le problème de localisation du véhicule est divisé en deux parties, afin de décomposer le problème. Dans la première partie de ce chapitre, les instants de rendez-vous ne seront pas exploités. Ainsi, divers estimateurs existant dans la littérature seront présentés pour réaliser une estimation de la position du véhicule à partir de mesures de cap et vitesse seuls. Dans ce cadre, une contribution originale sera décrite. Puis dans la deuxième partie, la géométrie d'exploration de la zone (instants de rendez-vous) sera exploitée afin de réaliser la localisation de l'engin par mesures de cap et vitesse (notée dans la suite localisation AR/(C,V)). Soulignons ici qu'il s'agit d'une problématique de localisation d'un véhicule. Ses positions seules seront estimées, alors que ses vitesses ne le seront pas.

3.1 Modélisation du problème en 2 dimensions

Le contexte est le suivant : nous nous plaçons dans le cas de la cartographie d'une zone d'intérêt par un véhicule sous-marin. Cet engin (ROV téléopéré par câble¹ ou AUV autonome²) est équipé d'au moins une caméra verticale et de capteurs lui permettant de naviguer et de se positionner (Loch-Doppler, gyrocompas, base courte...). L'engin quadrille la zone de façon à ce qu'elle soit cartographiée globalement. Un exemple de trajectoire est présenté figure 3.1.

L'utilisation des données d'estime pour localiser l'engin consiste le plus souvent en une simple intégration des caps et vitesses bruités à partir d'une position donnée, ou dans le cadre inertiel, à la mise en place d'une double intégration et d'un estimateur de type Kalman. Dans toute la suite, nous ne considérerons que le cas de données de cap et de vitesse.

¹ROV = Remotely Operated Vehicle

²AUV = Autonomous Underwater Vehicle

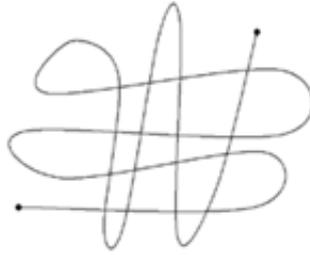


FIG. 3.1 – Exemple de trajectoire lors du quadrillage d'une zone d'intérêt.

Sur les figures 3.2(a) et 3.2(b) sont représentées les trajectoires en mètres vraie et intégrée à partir des caps et vitesses bruités (trajectoire dite "à l'estime"), pour deux trajectoires de plus de 1000 mesures. Comme exposé précédemment, dans le chapitre 2, partie 2.2.1, on remarque une dérive en position due à l'intégration des bruits.

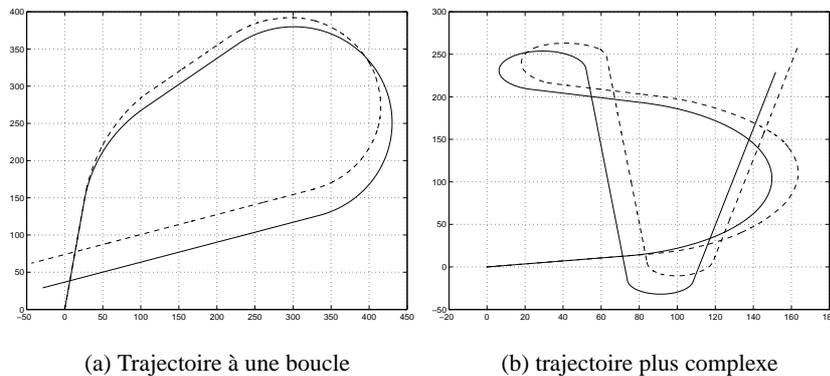


FIG. 3.2 – Exemples de trajectoires intégrées à partir des caps et vitesses bruités (trait pointillé). Les trajectoires vraies sont en train plein.

Notre objectif est de **localiser de façon précise le véhicule porteur de la caméra**. Pour ce faire, il est intéressant de tenir compte de la façon dont l'engin parcourt la zone d'intérêt. Ainsi, lors du quadrillage d'une zone d'intérêt, plusieurs types d'information supplémentaires nécessitent d'être pris en compte. Les données de cap et de vitesse acquises par respectivement un gyrocompas et un Loch-Doppler, sont entachées d'un bruit (voir modélisation en section 3.1.4). Les positions à l'instant initial et à l'instant final sont connues de façon certaine ou avec une précision absolue donnée.

Il est également intéressant d'utiliser une information sur la géométrie d'exploration de la zone. En effet, pour réaliser une couverture optimale de la zone d'exploration, l'engin suit une trajectoire de type "quadrillage", c'est-à-dire que l'engin repasse au-dessus de certains endroits. Ces instants dits "de rendez-vous" seront explicités dans la section 3.1.2 et une méthode pour les estimer sera décrite dans la partie 5.3.

La solution que nous proposons pour résoudre le problème d'estimation de la position du véhicule prend en compte tous ces types d'information.

3.1.1 Trajectoire simple : connaissance seule des points extrémaux

La position des points initial et final est supposée connue, soit en positionnant un amer artificiel sur le fond marin, soit en acquérant des données de capteurs de position absolue, connaissant leur précision, soit en connaissant parfaitement la position du bateau, par exemple dans le cas d'une mission en mer relativement courte, sachant que l'on part du port et que l'on y revient. Dans la suite, nous utiliserons le terme **trajectoire simple** pour le cas d'une trajectoire dont les seules informations disponibles sont les positions des points initial et final.

3.1.2 Trajectoire complexe : connaissance supplémentaire des instants de rendez-vous

3.1.2.1 Définition des instants de rendez-vous

Les instants de rendez-vous correspondent aux instants où l'engin repasse au-dessus d'un même lieu géométrique. Ils sont déterminés d'une part, grâce à la navigation à l'estime (localisation $A/(C,V)$), c'est-à-dire en intégrant les mesures de cap et de vitesse, et d'autre part, grâce aux images. Dans un premier temps, la navigation à l'estime permet de déterminer de façon approchée les deux instants I et J où l'engin passe sur une zone de rendez-vous. Dans un deuxième temps, deux ensembles d'images sont construits : l'ensemble F_I , à partir d'une fenêtre ΔI construite autour de l'image d'indice I , \mathbf{I}_I (les images correspondant aux instants $F_I = \{I - \Delta I; I + \Delta I\}$), et l'ensemble F_J à partir d'une fenêtre autour de l'image d'indice J , \mathbf{I}_J ($F_J = \{J - \Delta J; J + \Delta J\}$).

La taille de la fenêtre ΔI (comme celle de la fenêtre ΔJ) sera déterminée grâce à la connaissance des variances des positions intégrées à partir des mesures de cap et vitesse (localisation $A/(C,V)$).

Enfin, une recherche sur ces deux ensembles F_I et F_J permet de déterminer les instants de rendez-vous. Le couple d'instants sera déterminé par un appariement de deux images, appartenant respectivement à F_I et F_J . Les images seront appariées si elles contiennent le plus d'information en commun et sont les plus proches au sens géométrique. Cette méthode est présentée dans la section 5.3.3 et consiste à rechercher les deux images les plus proches à partir de deux ensembles d'images.

Dans la suite, nous utiliserons le terme **trajectoire complexe** lorsque nous parlerons d'une trajectoire dont on connaît les positions des points initial et final et les instants de rendez-vous.

3.1.2.2 Problème de discrétisation

Il existe néanmoins un problème qui réside dans la discrétisation de la trajectoire. Ce problème est illustré sur la figure 3.3. En effet, les deux positions X_I et X_J au point de rendez-vous ne se recouvrent pas exactement. L'erreur est due à la discrétisation des mesures.

Ce problème peut être résolu grâce à la détermination de la *transformation* entre les deux images aux instants I et J et donc à l'estimation de ce déplacement en mètres (pour le passage du plan image au plan 3D, voir section 6.2). L'information donnée par $X_I = X_J$ est équivalente à $X_I = X_J + distance(X_I, X_J)$. Aussi, dans ce qui suit, nous allons supposer que cette distance est nulle. Cependant, l'algorithme pourrait être facilement adapté pour tenir compte de cette distance.

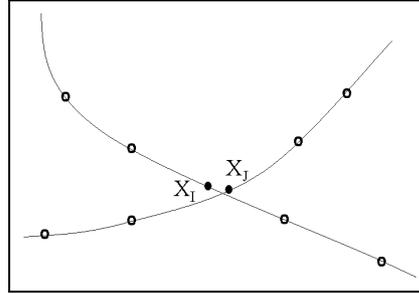


FIG. 3.3 – Problème apparaissant lors du calcul des instants de rendez-vous.

3.1.3 Modélisation proposée pour la trajectoire

La trajectoire est représentée par une suite de positions $\{X_k, k = 1..N\}$ avec $X_k = (x_k, y_k)^T$. Soit c_k et v_k les cap et vitesse vrais à l'instant de mesure t_k . Soit $X_k = (x_k, y_k)^T$ la position de l'engin à l'instant de mesure t_k . Les trajectoires sont assimilées à des lignes polygonales (segments de droite entre deux instants de mesure, voir figure 3.4) :

$$\begin{cases} x_k = x_{k-1} + (t_k - t_{k-1})v_{k-1} \sin c_{k-1} \\ y_k = y_{k-1} + (t_k - t_{k-1})v_{k-1} \cos c_{k-1} \end{cases} \quad (3.1)$$

Soit encore :

$$\begin{cases} c_k = \arctan \frac{x_{k+1} - x_k}{y_{k+1} - y_k} \\ v_k = \frac{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}}{T_k} \end{cases} \quad (3.2)$$

Dans la suite, il s'agira d'estimer le paramètre $\theta = [X_2^T \dots X_k^T \dots X_{N-1}^T]^T$ (X_1 et X_N étant connus ou préalablement estimés).

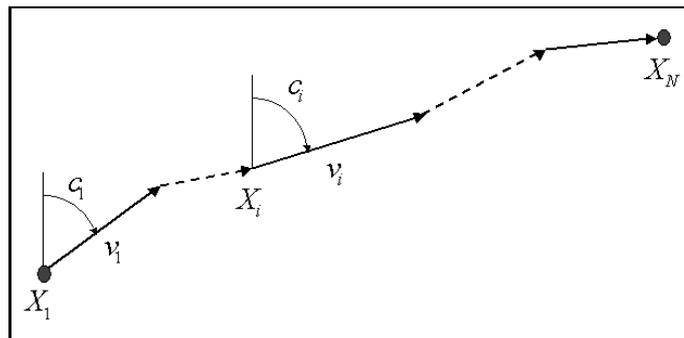


FIG. 3.4 – Trajectoires assimilées à des lignes polygonales.

3.1.4 Les mesures

Les instruments de mesure tels que le Loch-Doppler et le gyrocompas fournissent des mesures entachées de bruit, qui peuvent être modélisés par des bruits indépendants.

Les mesures de cap et de vitesse ont été modélisées par un bruit blanc centré additif, noté par respectivement ϵ_{c_k} , de variance σ_c constante et ϵ_{v_k} de variance σ_v constante. Pour mettre au point notre algorithme, nous avons modélisé les bruits de mesure par des bruits gaussiens centrés, de variance donnée. Cependant, afin de tester la robustesse de notre solution, nous avons étudié le cas où le bruit, toujours blanc, suit d'autres lois que la loi gaussienne.

Soit \tilde{c}_k et \tilde{v}_k les cap et vitesse mesurés à l'instant t_k . On a :

$$\begin{aligned}\tilde{c}_k &= c_k + \epsilon_{c_k} \\ \tilde{v}_k &= v_k + \epsilon_{v_k}\end{aligned}\tag{3.3}$$

Soit encore, d'après l'équation 3.2 :

$$\begin{aligned}\tilde{c}_k &= c_k(\theta) + \epsilon_{c_k} = \arctan\left(\frac{x_{k+1} - x_k}{y_{k+1} - y_k}\right) + \epsilon_{c_k} \\ \tilde{v}_k &= v_k(\theta) + \epsilon_{v_k} = \frac{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}}{T_k} + \epsilon_{v_k}\end{aligned}\tag{3.4}$$

Il arrive que les mesures soient interrompues. Dans ce cas, néanmoins peu fréquent en navigation à l'estime (et contrairement à la navigation acoustique), la solution à envisager est l'interpolation. En général, à moins que l'engin fasse une "embardée", les mesures sont relativement continues, ce qui est nécessaire à une bonne interpolation.

D'autre part, des points aberrants peuvent apparaître au fil des mesures. Ces points sont dits aberrants car ils sortent de la continuité des mesures. Contrairement à la navigation absolue (avec une base ultracourte par exemple), l'apparition de points aberrants est peu fréquente pour la navigation à l'estime. Cependant, un lissage préalable peut être nécessaire, pour les détecter et ainsi les retirer des mesures.

Après avoir présenté le problème et proposé une modélisation des mesures, nous allons décrire les algorithmes évalués dans le cadre de cette thèse. Nous présentons dans ce qui suit quelques méthodes permettant d'estimer des **trajectoires simples**. Puis nous décrirons leur adaptation dans le cas d'une **trajectoire complexe**.

3.2 Algorithmes de filtrage récursifs adaptés au problème

Dans cette partie, nous allons proposer la formalisation et l'évaluation de deux méthodes classiques qui permettent d'estimer les positions à partir des caps et vitesses. Ces méthodes dynamiques reposent sur le filtrage de Kalman.

Le filtrage de Kalman est une méthode récursive qui estime l'état d'un système à partir de mesures bruitées [Kal60][WB01][CC87].

Après en avoir rappelé quelques principes, notamment dans les cas linéaire et non-linéaire, nous présentons deux méthodes dérivées du filtrage de Kalman, qui peuvent être employées pour résoudre le problème d'estimation de la position de l'engin sous-marin.

3.2.1 Le filtrage de Kalman discret

Le filtrage de Kalman consiste à estimer l'état $X \in \mathbb{R}^n$ à l'instant t_k d'un système dynamique gouverné par une équation linéaire aux différences :

$$X_{k+1} = F_k X_k + G_k U_k + v_k \quad (3.5)$$

où

- F_k est la matrice d'état à l'instant t_k ,
- X_k est le vecteur d'état à l'instant t_k ,
- G_k est la matrice de commande à t_k ,
- U_k est la commande à t_k ,
- v_k est le bruit d'état à t_k .

et une équation de mesure à l'instant t_k :

$$Z_k = H_k X_k + w_k \quad (3.6)$$

où

- Z_k est la mesure à l'instant t_k ,
- H_k est la matrice de mesure à l'instant t_k ,
- w_k est le bruit de mesure à t_k .

Les matrices F_k , G_k , H_k sont supposées connues. Les bruits d'état w_k et de mesure v_k sont supposés blancs, indépendants l'un de l'autre, et gaussiens :

$$v_k \sim \mathbf{G}(0, Q_k)$$

$$w_k \sim \mathbf{G}(0, R_k)$$

où Q_k est la matrice de variance-covariance du bruit d'état et R_k la matrice de variance-covariance du bruit de mesure. Dans l'utilisation de ce filtrage, nous prendrons $Q_k = Q$ et $R_k = R$ constantes.

Ces matrices doivent être déterminées avant le filtrage. R est aisément identifiable si le bruit de mesure est correctement modélisé (dans notre cas $R = \begin{pmatrix} \sigma_c^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix}$). Dans le cas de Q , l'opérateur peut supposer que l'erreur sur l'état est petite et choisir une valeur relativement faible à Q .

Le principe général du filtre de Kalman consiste à estimer l'état d'un système à l'instant t_k en deux étapes successives. La première est l'étape de prédiction qui met en jeu l'estimation à l'instant t_k . La seconde est l'étape de correction qui met en jeu grâce à la mesure à l'instant t_{k+1} pour actualiser l'état prédit.

Les équations correspondant à ces deux étapes sont bien connues [BSL93][WB01]. Elles sont les suivantes :

Étape de Prédiction

| | |
|--|-------|
| Prédiction de l'état $\hat{X}_{k+1 k} = F_k \hat{X}_{k k} + G_k U_k$ | (3.7) |
| Prédiction de la mesure $\hat{Z}_{k+1 k} = H_{k+1} \hat{X}_{k+1 k}$ | |
| Prédiction de la covariance d'état $P_{k+1 k} = F_k P_{k k} F_k^T + Q_k$ | |
| Covariance de l'innovation $S_{k+1} = H_{k+1} P_{k+1 k} H_{k+1}^T + R_k$ | |
| Gain $W_{k+1} = P_{k+1 k} H_{k+1}^T S_{k+1}^{-1}$ | |

Étape de correction ou estimation

| | |
|---|-------|
| Mise à jour de l'état $\hat{X}_{k+1 k+1} = \hat{X}_{k+1 k} + W_{k+1} [Z_{k+1} - \hat{Z}_{k+1 k}]$ | (3.8) |
| Mise à jour de la covariance d'état $P_{k+1 k+1} = P_{k+1 k} - W_{k+1} S_{k+1} W_{k+1}^T$ | |

3.2.2 Lisseur par filtrage de Kalman Aller-Retour

Le filtrage de Kalman Aller-Retour consiste à filtrer une première fois les données mesurées en avant ("forward"), connaissant la première mesure X_1 , puis filtrer une nouvelle fois les mesures en arrière ("backward"), connaissant la dernière mesure X_N . Il suffit alors de fusionner les deux trajectoires obtenues par un estimateur du maximum de vraisemblance ou des moindres carrés.

Soit N le nombre de mesures à notre disposition, $\hat{X}_{k|k}^+$ et $P_{k|k}^+$ l'état et la covariance estimés par le filtrage forward (donc sur les k premières mesures) et $\hat{X}_{k|N-k}^-$ et $P_{k|N-k}^-$ l'état et la covariance calculés par le filtrage backward (donc calculées sur les $N - k$ dernières mesures). Ces deux états $\hat{X}_{k|k}^+$ et $\hat{X}_{k|N-k}^-$ suivent les lois gaussiennes lorsque les bruits ϵ_k sont eux-mêmes gaussiens.

$$\hat{X}_{k|k}^+ \sim \mathbf{G}(X_k, P_{k|k}^+)$$

$$\hat{X}_{k|N-k}^- \sim \mathbf{G}(X_k, P_{k|N-k}^-)$$

On peut supposer que $\hat{X}_{k|k}^+$ et $\hat{X}_{k|N-k}^-$ sont indépendants puisqu'ils ne sont pas calculés à partir des mêmes mesures. Ainsi, le vecteur ci-dessous suit la loi gaussienne (où ϵ est négligeable) :

$$\begin{pmatrix} \hat{X}_{k|k}^+ \\ \hat{X}_{k|N-k}^- \end{pmatrix} \sim \mathbf{G} \left[\begin{pmatrix} \hat{X}_k \\ \hat{X}_k \end{pmatrix}, \begin{pmatrix} P_{k|k}^+ & \epsilon \\ \epsilon & P_{k|N-k}^- \end{pmatrix} \right]$$

Calculons alors la vraisemblance du vecteur ci-dessus (pour plus de simplicité, nous avons choisi de retirer les indices) :

$$L = \text{cte} \exp \left(-\frac{1}{2} [(X^+ - X)^T (P^+)^{-1} (X^+ - X) + (X^- - X)^T (P^-)^{-1} (X^- - X)] \right) \quad (3.9)$$

Son maximum de vraisemblance est alors donné par

$$\min_X L = \min_X \exp \left(-\frac{1}{2} \left[(X^+ - X)^T P^{+-1} (X^+ - X) + (X^- - X)^T P^{-1} (X^- - X) \right] \right) \quad (3.10)$$

Soit en dérivant, on obtient l'estimé du filtrage de Kalman aller-retour $\hat{X}_{k|N}$:

$$\hat{X}_{k|N} = \left(P^{+-1} + P^{-1} \right)^{-1} \left(P^{+-1} X^+ + P^{-1} X^- \right) \quad (3.11)$$

3.2.3 Lissage RTS

Cette méthode a été conçue par Rauch, Tung et Striebel [RTS65]. Le principe du lissage consiste à estimer l'état à l'instant t_k connaissant toutes les mesures de t_1 à t_j pour $j > k$. En particulier, cette technique d'estimation est très employée lorsqu'on veut lisser une trajectoire alors que toutes les mesures relatives à une expérience sont connues.

En pratique, cette méthode revient à filtrer la trajectoire avec un filtre de Kalman aller et à faire un lissage récursif arrière (backward) [BSL93].

Supposons que grâce au filtrage de Kalman aller, nous ayons à notre disposition les $X_{k|k}^+$. Le lissage récursif revient alors à calculer les $\hat{X}_{k|N}$ (s'il y a N mesures), de $k = N$ à 1, selon :

$$\begin{cases} \hat{X}_{k|N} = X_{k|k}^+ + C_k [\hat{X}_{k+1|N} - F_k X_{k|k}^+] \\ C_k = P_{k|k} F_k P_{k+1|k}^{-1} \\ P_{k|N} = P_{k|k} + C_k [P_{k+1|N} - P_{k+1|k}] C_k^T \end{cases}$$

Nous prenons comme initialisations les dernières valeurs du filtre de Kalman aller.

3.2.4 Le filtrage de Kalman étendu (Extended Kalman Filter - EKF)

Le filtrage de Kalman consiste à rechercher l'estimée de l'état d'un processus gouverné par une équation linéaire. Cependant, dans notre cas, X_k représente la position à l'instant k et les mesures Z_k sont données par les caps et vitesses. Il n'existe donc pas de relation linéaire entre les caps et vitesses et les positions. A ce moment-là, nous pouvons rechercher une estimation sous-optimale de l'état (mais qui fonctionne tout de même dans bon nombre de cas) : le filtrage de Kalman étendu (EKF pour Extended Kalman Filter). Le principe est de linéariser les équations du système. Soit le système suivant :

$$\begin{cases} X_{k+1} = \mathbf{F}_k (X_k, U_k) + v_k \\ Z_{k+1} = \mathbf{H}_k (X_k) + w_k \end{cases}$$

Les équations du filtre de Kalman étendu deviennent :

Étape de Prédiction

| | |
|---|--------|
| Prédiction de l'état $\hat{X}_{k+1 k} = \mathbf{F}_k (X_{k k}, U_k)$ $F_k = \frac{d\mathbf{F}_k}{dX} (\hat{X}_{k k})$ Prédiction de la mesure $\hat{Z}_{k+1 k} = \mathbf{H}_{k+1} (\hat{X}_{k+1 k})$ $H_k = \frac{d\mathbf{H}_k}{dX} (X_{k k-1})$ Prédiction de la covariance d'état $P_{k+1 k} = F_k P_{k k} F_k^T + Q_k$ Covariance de l'innovation $S_{k+1} = H_{k+1} P_{k+1 k} H_{k+1}^T + R_k$ Gain $W_{k+1} = P_{k+1 k} H_{k+1}^T S_{k+1}^{-1}$ | (3.12) |
|---|--------|

Étape de correction ou estimation

| | |
|--|--------|
| Mise à jour de l'état $\hat{X}_{k+1 k+1} = \hat{X}_{k+1 k} + W_{k+1} [Z_{k+1} - \hat{Z}_{k+1 k}]$ Mise à jour de la covariance d'état $P_{k+1 k+1} = P_{k+1 k} - W_{k+1} S_{k+1} W_{k+1}^T$ | (3.13) |
|--|--------|

Dans notre cas, si $(x, y)^T$ est la position et $(\dot{x}, \dot{y})^T$ est la dérivée de la position, le vecteur d'état X est donné par :

$$X = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (3.14)$$

En supposant que la période d'échantillonnage des mesures est $T = 1$ s, la première équation de prédiction est :

$$X_{k+1|k} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{k|k} \quad (3.15)$$

Le vecteur de mesure est donné par

$$Z_k = \begin{bmatrix} c_k \\ v_k \end{bmatrix} \quad (3.16)$$

L'équation de mesure est donnée par

$$Z_k = \mathbf{H}_k(X_k) = \begin{bmatrix} \arctan \frac{\dot{x}_k}{\dot{y}_k} \\ \sqrt{\dot{x}_k^2 + \dot{y}_k^2} \end{bmatrix} \quad (3.17)$$

d'où

$$\frac{d\mathbf{H}_k}{dX} = \begin{bmatrix} 0 & 0 & \frac{\dot{y}_k}{\dot{x}_k^2 + \dot{y}_k^2} & -\frac{\dot{x}_k}{\dot{x}_k^2 + \dot{y}_k^2} \\ 0 & 0 & \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} & \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{bmatrix} \quad (3.18)$$

Les principes du filtrage de Kalman aller-retour ainsi que du lissage RTS restent les mêmes. Il suffit simplement de remplacer F_k par $\frac{dF_k}{dX}(\hat{X}_{k|k})$ et H_k par $\frac{dH_k}{dX}(X_{k|k-1})$ dans les équations.

3.3 Estimation par la méthode des moindres carrés

La méthode des moindres carrés, indépendamment découverte par Gauss et Legendre, permet de comparer des données expérimentales généralement entachées d'erreurs de mesure à un modèle mathématique décrivant ces données.

3.3.1 Position du problème

Reprenons les notations courantes en théorie de l'estimation. On pose $z = (z_1^T, \dots, z_n^T)^T$ réalisations de $Z = (Z_1^T, \dots, Z_n^T)^T$. Dans notre cas, $z_k = (c_k, v_k)^T$. θ est le paramètre à estimer (dans notre cas, les positions de l'engin inconnues). Les équations de mesure sont données par :

$$z_k = h_k(\theta) + \epsilon_k \quad (3.19)$$

avec $h_k(\theta) = [h_{k_1}(\theta), h_{k_2}(\theta)]^T = (c_k(\theta), v_k(\theta))^T$, donnés par l'équation 3.2.

Le critère des moindres carrés à minimiser est donné par

$$Q(\theta) = \frac{1}{2} [z - h(\theta)]^T R^{-1} \frac{1}{2} [z - h(\theta)] \quad (3.20)$$

où R est la matrice de variance-covariance des bruits de mesure (diagonale dans notre cas, puisque les bruits sont indépendants). Étant donné la forme particulière de R , l'expression du critère se simplifie en :

$$Q(\theta) = \frac{1}{2} \sum_{k=1}^{N-1} \left[\frac{\tilde{c}_k - c_k(\theta)}{\sigma_c} \right]^2 + \frac{1}{2} \sum_{k=1}^{N-1} \left[\frac{\tilde{v}_k - v_k(\theta)}{\sigma_v} \right]^2 \quad (3.21)$$

Remarque : Après estimation, le critère $Q(\hat{\theta})$ suit approximativement une loi du χ^2 à d degrés de liberté, d étant le nombre de mesures diminué de la dimension de θ [BSL93]. On aura alors $E[Q(\theta)] = \frac{1}{2}d$ et $Var[Q(\theta)] = \frac{1}{4}2d = \frac{1}{2}d$.

Dans le cas d'une trajectoire simple à N points, on a $N-1$ mesures de cap et vitesse et $N-2$ points (abscisse et ordonnée) à estimer (les deux points extrêmes sont connus). Ainsi, $d = 2(N-1) - 2(N-2) = 2$, le critère suit une loi du χ_2^2 .

3.3.2 Minimisation du critère $Q(\theta)$

Le cas qui nous est posé est celui d'une minimisation d'un critère non-linéaire. En effet, $Q(\theta)$ fait intervenir les fonctions non-linéaires arctan et $\sqrt{\cdot}$. Pour résoudre ce problème de minimisation non-linéaire, il convient d'appliquer une méthode de minimisation récursive classique comme celle de Gauss-Newton ou Levenberg-Marquardt.

La méthode de Gauss-Newton est une méthode itérative qui s'appuie sur le calcul du gradient G et du hessien H du critère $Q(\theta)$.

1. Initialisation : θ_0
2. Itérations : tant que $|\theta_k - \theta_{k-1}| > \epsilon$ ou $iter > max_{iter}$
 - $G = \nabla Q(\theta_k)$
 - $H = \nabla \nabla^T Q(\theta_k)$
 - $\theta_k = \theta_{k-1} - H^{-1}G$

Cette méthode peut être améliorée pour une meilleure convergence si le hessien n'est pas numériquement défini positif. C'est la méthode de Levenberg-Marquardt. D'autre part, la mise en place d'un pas permet d'accélérer la convergence vers la solution finale.

Les étapes de cette méthode de minimisation sont similaires à celles de la méthode de Gauss-Newton. Elles sont les suivantes :

1. Initialisation : θ_0
2. Itérations : tant que $|\theta_k - \theta_{k-1}| > \epsilon$ ou $iter > max_{iter}$
 - $G = \nabla Q(\theta_k)$
 - $H = \nabla \nabla^T Q(\theta_k)$
 - $\mathbb{H} = H + \alpha I$ si H n'est pas définie positive (α étant choisi de façon à ce que \mathbb{H} soit définie positive, par exemple $\alpha = -(\text{plus petite valeur propre}) + 10^{-6}$).
 - $\theta_k = \theta_{k-1} - p_k \mathbb{H}^{-1}G$ où p_k est choisi dans $\{0.1, 0.2, \dots, 1\}$ de façon à ce que $Q(\theta_k)$ soit minimal.

Le gradient du critère est donné par :

$$\nabla_{\theta} Q(\theta) = - \sum_{k=1}^{N-1} \left[\frac{\tilde{c}_k - c_k(\theta)}{\sigma_c^2} \right] \nabla_{\theta} c_k(\theta) - \sum_{k=1}^{N-1} \left[\frac{\tilde{v}_k - v_k(\theta)}{\sigma_v^2} \right] \nabla_{\theta} v_k(\theta) \quad (3.22)$$

Le Hessien est donné par :

$$H = \sum_{k=1}^{N-1} \frac{\nabla_{\theta} c_k \nabla_{\theta}^T c_k}{\sigma_c^2} + \sum_{k=1}^{N-1} \frac{\nabla_{\theta} v_k \nabla_{\theta}^T v_k}{\sigma_v^2} \quad (3.23)$$

$$\nabla_{\theta} c_k(\theta) = \begin{bmatrix} \frac{\partial c_k}{\partial \theta_1} \\ \frac{\partial c_k}{\partial \theta_2} \\ \dots \\ \frac{\partial c_k}{\partial \theta_{N-1}} \end{bmatrix} \quad (3.24)$$

Soit pour $k = 2..N - 2$, on calcule le vecteur à $2(N - 3) = 2N - 6$ dimensions, qui a seulement quatre termes non-nuls, puisque c_k ne dépend que de x_k, x_{k+1}, y_k et y_{k+1} selon l'équation 3.2 :

$$\nabla_{\theta} c_k(\theta) = \left[0 \quad \dots \quad 0 \quad \frac{\partial c_k}{\partial x_k} \quad \frac{\partial c_k}{\partial y_k} \quad \frac{\partial c_k}{\partial x_{k+1}} \quad \frac{\partial c_k}{\partial y_{k+1}} \quad 0 \quad \dots \quad 0 \right]^T \quad (3.25)$$

Pour $k = 1$

$$\nabla_{\theta} c_1(\theta) = \left[\frac{\partial c_1}{\partial x_2} \quad \frac{\partial c_1}{\partial y_2} \quad 0 \quad \dots \quad 0 \right]^T \quad (3.26)$$

et pour $k = N - 1$

$$\nabla_{\theta} c_{N-1}(\theta) = \left[0 \quad \dots \quad 0 \quad \frac{\partial c_{N-1}}{\partial x_{N-1}} \quad \frac{\partial c_{N-1}}{\partial y_{N-1}} \right]^T \quad (3.27)$$

D'autre part, pour tout $k \in [2, N - 1]$, on obtient en développant les calculs :

$$\left\{ \begin{array}{l} \frac{\partial c_k}{\partial x_k} = \frac{-(y_{k+1} - y_k)}{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \\ \frac{\partial c_k}{\partial y_k} = \frac{x_{k+1} - x_k}{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \\ \frac{\partial x_{k+1}}{\partial c_k} = \frac{y_{k+1} - y_k}{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \\ \frac{\partial y_{k+1}}{\partial c_k} = \frac{-(x_{k+1} - x_k)}{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \end{array} \right. \quad (3.28)$$

$$\left\{ \begin{array}{l} \frac{\partial v_k}{\partial x_k} = \frac{-(x_{k+1} - x_k)}{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}} \\ \frac{\partial v_k}{\partial y_k} = \frac{-(y_{k+1} - y_k)}{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}} \\ \frac{\partial x_{k+1}}{\partial v_k} = \frac{x_{k+1} - x_k}{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}} \\ \frac{\partial y_{k+1}}{\partial v_k} = \frac{y_{k+1} - y_k}{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}} \end{array} \right. \quad (3.29)$$

Notons que pour l'une ou l'autre des méthodes, lors de la convergence, le Hessien du critère est proche de l'information de Fisher (voir définition dans la partie 4.1.1).

La dimension de θ est très grande (plus de 1000 positions à estimer pour une trajectoire donnée). Aussi, la minimisation de critère dans un espace de grande dimension est difficilement réalisable en pratique. D'autre part, les coûts calculatoires sont très importants. En pratique, la méthode d'estimation par les moindres carrés est difficilement envisageable. Néanmoins, quelques résultats sur des trajectoires comportant peu de points seront proposés dans le paragraphe 4.3.2.

Remarque : il existe toute une panoplie de méthodes qui permettent de résoudre le problème de minimisation [CZ96].

3.4 Une nouvelle méthode : l'auto-estimation

La méthode originale que nous proposons pour résoudre notre problème d'estimation des positions s'inspire des méthodes que nous avons évaluées précédemment, comme l'EKF aller-retour [CC87][WB01] et le lissage RTS [RTS65]. En effet, le temps réel n'étant pas requis, nous avons pu mettre au point une méthode utilisant toutes les données antérieures et postérieures à un instant t pour estimer la position $X = (x, y)^T$ à l'instant t . L'auto-estimation a été présentée dans [BJR03].

Implicitement, il s'agit d'estimer le paramètre :

$$\theta = [X_2^T \quad \dots \quad X_k^T \quad \dots \quad X_{N-1}^T]^T \quad (3.30)$$

(X_1 et X_N étant connus ou préalablement estimés), à partir des mesures \tilde{q}_k et \tilde{v}_k définies pour $k = 1..N - 1$ par les équations 3.2 et 3.3.

Le principe de la méthode proposée est le suivant : étant données les deux positions extrêmes (positions

initiale et finale), l'idée de base est de fusionner une trajectoire à l'estime "aller" et une trajectoire à l'estime "retour" (c'est-à-dire en parcourant les caps et vitesses en sens rétrograde).

Dans ce qui suit, nous présentons l'algorithme développé dans le cas où l'on ne tient pas compte des instants de rendez-vous.

3.4.1 Estimation aller

Soit $\hat{X}_{k|k}^+ = (\hat{x}_{k|k}^+, \hat{y}_{k|k}^+)^T$ la position estimée à l'instant t_k , connaissant les $(k - 1)$ premiers caps et vitesses. Les positions estimées sont calculées grâce à l'équation (3.1), en remplaçant φ_{-1} et v_{l-1} par respectivement \tilde{c}_{l-1} et \tilde{v}_{l-1} .

$$\begin{aligned}\hat{x}_{k|k}^+ &\triangleq x_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \sin \tilde{c}_{l-1} \\ \hat{y}_{k|k}^+ &\triangleq y_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \cos \tilde{c}_{l-1}\end{aligned}\quad (3.31)$$

3.4.2 Estimation retour

Soit $\hat{X}_{k|N-k}^- = (\hat{x}_{k|N-k}^-, \hat{y}_{k|N-k}^-)^T$ la position estimée à l'instant t_k , connaissant les $(N - k)$ caps et vitesses en partant du point final. En utilisant une fois de plus l'équation (3.1), on obtient :

$$\begin{aligned}\hat{x}_{k|N-k}^- &\triangleq x_N - \sum_{l=N}^{l=k+1} (t_l - t_{l-1}) \tilde{v}_{l-1} \sin \tilde{c}_{l-1} \\ \hat{y}_{k|N-k}^- &\triangleq y_N - \sum_{l=N}^{l=k+1} (t_l - t_{l-1}) \tilde{v}_{l-1} \cos \tilde{c}_{l-1}\end{aligned}\quad (3.32)$$

3.4.3 Aller-retour

L'idée de départ de la méthode proposée ici est de fusionner l'estimé aller $\hat{X}_{k|k}^+$ avec l'estimé retour $\hat{X}_{k|N-k}^-$.

À un chaque instant, les caps et vitesses mesurés précédant cet instant et les caps et vitesses suivant cet instant sont indépendants. Ainsi, les deux estimés, l'aller $\hat{X}_{k|k}^+$ et le retour $\hat{X}_{k|N-k}^-$ sont indépendants. Nous

pouvons dire que le vecteur $\begin{pmatrix} \hat{X}_{k|k}^+ \\ \hat{X}_{k|N-k}^- \end{pmatrix}$ a pour espérance

$$\mathbf{E} \left[\begin{pmatrix} \hat{X}_{k|k}^+ \\ \hat{X}_{k|N-k}^- \end{pmatrix} \right] = \begin{pmatrix} \hat{X}_{k|N}^+ \\ \hat{X}_{k|N}^- \end{pmatrix}\quad (3.33)$$

et pour matrice de variance-covariance :

$$\mathbf{P} \left[\begin{pmatrix} \hat{X}_{k|k}^+ \\ \hat{X}_{k|N-k}^- \end{pmatrix} \right] = \begin{pmatrix} \mathbf{P} [\hat{X}_{k|k}^+] & 0 \\ 0 & \mathbf{P} [\hat{X}_{k|N-k}^-] \end{pmatrix}\quad (3.34)$$

La fusion de ces deux estimateurs au sens des moindres carrés est donnée par :

$$\hat{X}_{k|N} \triangleq \left(\mathbf{P} \left[\hat{X}_{k|k}^+ \right]^{-1} + \mathbf{P} \left[\hat{X}_{k|N-k}^- \right]^{-1} \right)^{-1} \cdot \left(\mathbf{P} \left[\hat{X}_{k|k}^+ \right]^{-1} \hat{X}_{k|k}^+ + \mathbf{P} \left[\hat{X}_{k|N-k}^- \right]^{-1} \hat{X}_{k|N-k}^- \right) \quad (3.35)$$

3.4.4 Calcul des biais et variances

Les biais et les variances théoriques des navigations à l'estime "aller" et "retour" sont facilement calculables. Nous donnons ci-dessous les formules pour les positions aller (les formules pour les positions retour sont de la même forme). Le détail des calculs est disponible dans l'annexe A. Néanmoins, tous les calculs reposent sur la formule suivante, qui est valable pour des bruits gaussiens :

$$\mathbf{E} [\sin \tilde{c}_{l-1}] = e^{-\frac{\sigma_c^2}{2}} \sin c_{l-1} \quad (3.36)$$

Biais :

$$\begin{aligned} b_x(k) &= \sum_{l=2}^k (t_l - t_{l-1}) v_{l-1} \sin c_{l-1} [1 - e^{-\sigma_c^2/2}] \\ b_y(k) &= \sum_{l=2}^k (t_l - t_{l-1}) v_{l-1} \cos c_{l-1} [1 - e^{-\sigma_c^2/2}] \end{aligned} \quad (3.37)$$

Variances :

$$\begin{aligned} \mathbf{Var} [\hat{x}_k] &= \sum_{l=2}^k \left[\frac{v_{l-1}^2}{2} \left(1 - e^{-\sigma_c^2} + 2 \cos 2c_{l-1} \sinh \frac{\sigma_c^2}{2} e^{-3\sigma_c^2/2} \right) + \frac{\sigma_v^2}{2} \left(1 - \cos 2c_{l-1} e^{-2\sigma_c^2} \right) \right] (t_l - t_{l-1})^2 \\ \mathbf{Var} [\hat{y}_k] &= \sum_{l=2}^k \left[\frac{v_{l-1}^2}{2} \left(1 - e^{-\sigma_c^2} - 2 \cos 2c_{l-1} \sinh \frac{\sigma_c^2}{2} e^{-3\sigma_c^2/2} \right) + \frac{\sigma_v^2}{2} \left(1 + \cos 2c_{l-1} e^{-2\sigma_c^2} \right) \right] (t_l - t_{l-1})^2 \end{aligned} \quad (3.38)$$

3.4.5 Réduction du biais

Les termes aller $\hat{X}_{k|k}^+$ et retour $\hat{X}_{k|N-k}^-$ présentent tous deux un biais qui peut être réduit avant l'étape de fusion, en effectuant la fusion sur les termes aller $\tilde{X}_{k|k}^+$ et retour $\tilde{X}_{k|N-k}^-$. Nous donnons ici la transformation à effectuer sur $\tilde{X}_{k|k}^+$:

$$\tilde{X}_{k|k}^+ \triangleq e^{\frac{\sigma_c^2}{2}} \hat{X}_{k|k}^+ + (1 - e^{\frac{\sigma_c^2}{2}}) X_1 \quad (3.39)$$

Les détails des calculs sont présentés dans l'annexe A, partie A.3.1.

Remarque : en effectuant le débiaisage, les variances et covariances sont multipliées par le terme $e^{\sigma_c^2}$ (voir A.3.2 et A.3.3).

3.5 Généralisation aux trajectoires complexes

Rappelons que pour une trajectoire complexe, nous tenons compte des positions initiale et finale ainsi que des instants de rendez-vous estimés. Pour pouvoir mettre en oeuvre les méthodes proposées précédem-

ment, nous allons décomposer une trajectoire complexe en un ensemble de trajectoires simples connexes. Ces trajectoires simples seront alors estimées dans un ordre précis par la méthode choisie (estimateur des moindres carrés, filtrage de Kalman, auto-estimation, etc...).

Dans cette section, nous allons exposer deux méthodes de décomposition de cette trajectoire, l'une propageant le traitement de l'extérieur vers l'intérieur, l'autre à partir de la notion de *plus court chemin* de la théorie des graphes.

NB : En réalisant cette propagation, l'erreur (biais, variances) sur les positions estimées aux instants de rendez-vous va se propager. Cependant, cette erreur est connue et aisément calculable.

3.5.1 Stratégie de parcours par l'extérieur

L'algorithme de parcours que nous proposons ici consiste à propager le traitement de l'extérieur vers l'intérieur de la trajectoire.

On définit l'enveloppe de la trajectoire de l'engin par la frontière du plus petit pavé de \mathbb{R}^2 , à laquelle on adjoint les segments extrémaux. On définit alors le *chemin extérieur* de la trajectoire par le parcours le plus court, inclus dans l'enveloppe de la trajectoire, qui joint les points extrémaux de la trajectoire (supposés connus).

Le traitement d'une trajectoire contenant plusieurs boucles consiste à :

1. Connaissant les deux points extrémaux, déterminer pour la trajectoire, le chemin extérieur le plus court entre ces deux points,
2. Réaliser l'estimation de ce chemin (filtrage de Kalman aller-retour, méthode d'auto-estimation, etc...),
3. Considérer les points de rendez-vous appartenant au chemin estimé comme connus,
4. Retirer la trajectoire extérieure et reconstituer un nouveau graphe,
5. Estimer le nouveau chemin extérieur à partir des points de position déjà estimée,
6. Revenir à l'étape (3).

Les figures 3.5(a) à 3.5(d) montrent un exemple de parcours de trajectoire par l'extérieur.

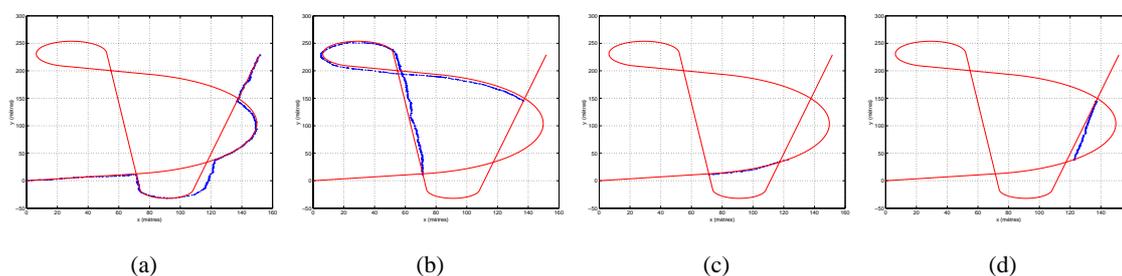


FIG. 3.5 – Exemple de stratégie de parcours par l'extérieur sur la trajectoire quadrillée.

Remarque : Cette méthode ne semble pas la plus judicieuse lorsque la trajectoire extérieure est très longue. Celle-ci, une fois estimée, présente une erreur croissante jusqu'au milieu de l'enveloppe, puis l'erreur décroît. Plus l'enveloppe correspondant à la trajectoire extérieure est longue, plus cette erreur sera importante. Or, cette erreur est transmise aux sous-trajectoires suivantes. Nous avons donc mis au point une autre façon de parcourir la trajectoire suivie par l'engin sous-marin, qui nous semble plus judicieuse a priori en matière de performances.

D'autre part, les boucles ne sont pas bien gérées car la méthode ne prend pas en compte une information d'instant de rendez-vous. On peut voir notamment sur la figure 3.5(b) que le chemin extérieur part de deux positions de rendez-vous estimées et inclut la boucle. Il ne considère donc pas qu'il y a un couple d'instants de rendez-vous au niveau du point où l'arc est rebouclé.

3.5.2 Stratégie de parcours par utilisation de la théorie des graphes

Notre trajectoire est composée d'arcs et de noeuds. Nous allons donc brièvement introduire dans cette partie la notion de graphe qui permet de modéliser la trajectoire. Pour plus de détails sur la théorie des graphes, voir le livre de Berge très complet sur le sujet [Ber83].

Après avoir défini la notion de plus court chemin dans un graphe, nous exposerons l'algorithme que nous avons choisi pour parcourir cette trajectoire.

3.5.2.1 Définitions

Graphe orienté Un graphe orienté $G = (\chi, A)$ est défini par l'ensemble $\chi = \{X_1, X_2, \dots, X_N\}$ de ses sommets et l'ensemble $A = \{a_1, a_2, \dots, a_m\}$ de ses arcs. Un arc de A est défini par un couple de sommets (également appelés "noeuds") distincts X et Y , X étant appelé l'origine et Y l'extrémité de l'arc.

Chemin On appelle chemin une suite $X_1 a_1 X_2 a_2 \dots a_{n-1} X_N$ de sommets X_i et d'arcs a_i , reliant X_i et X_{i+1} .

Graphe valué Un graphe valué est un graphe orienté $G = (\chi, A)$ muni d'une fonction $\gamma : A \rightarrow \mathbb{R}$, appelée fonction de coût. Dans la grande majorité des problèmes, les graphes valués ont des arêtes de coût (ou poids) positif. Un exemple simple d'un graphe valué est un réseau aérien dont les noeuds sont les villes reliées entre elles par avion. La fonction de coût d'un noeud à un autre est la durée de parcours en avion entre ces deux points.

Matrice de coûts d'un graphe Le coût d'un chemin est la somme des coûts des arcs de ce chemin. On peut définir la matrice des coûts du graphe, notée M_c , selon :

$$M_c(i, j) = \begin{cases} 0 & \text{si } i = j \\ \infty & \text{si } i \neq j \text{ et } (X_i, X_j) \notin A \\ \gamma(X_i, X_j) & \text{si } i \neq j \text{ et } (X_i, X_j) \in A \end{cases} \quad (3.40)$$

Matrice de coût minimum Soit $G = (\chi, A, \gamma)$ un graphe valué, et X et Y deux éléments de χ . Un chemin c de G de X à Y est dit minimum lorsque pour tout chemin c' de G allant de X à Y , on a $\gamma(c') \geq \gamma(c)$. On définit la matrice de coût minimum M_{cm} par :

$$M_{cm}(i, j) = \begin{cases} 0 & \text{si } i = j \\ \infty & \text{s'il n'existe pas de chemin de } X \text{ à } Y \\ \min\{\gamma(c) \mid c \text{ chemin de } X \text{ à } Y\} & \end{cases} \quad (3.41)$$

3.5.2.2 Recherche du plus court chemin

Trois problèmes et leurs solutions sont exposés dans la littérature [CGH96][Ber83][vL90] :

1. Étant donnés deux sommets X et Y , trouver un chemin de poids minimum, ou plus court chemin, de X à Y .
2. Étant donné un sommet X , trouver un plus court chemin de X à tous les sommets du graphe.
3. Trouver, pour toute paire de sommets X et Y , un plus court chemin de X à Y .

Notons que résoudre le problème 2 permet de résoudre les problèmes 1 et 3. Le problème 1 est celui qui nous intéresse. Les algorithmes mis en oeuvre sont différents selon les caractéristiques du graphe. Par exemple, lorsque tous les poids des arcs sont positifs ou nuls, on utilise l'algorithme de Dijkstra [Dij59]. Lorsque le graphe est sans circuit, on utilise l'algorithme de Bellman. Dans un cas plus général, on utilise l'algorithme de Ford ou celui de Ford-Dantzig [CGH96]. Dans notre cas, les arcs ont tous des poids positifs, il convient donc, pour des raisons de complexité algorithmique, d'utiliser l'algorithme de Dijkstra.

3.5.2.3 Algorithme de Dijkstra - Principe

Les plus courtes distances sont calculées de proche en proche, par ajustements successifs, à partir d'un sommet donné, pour tous les autres sommets du réseau.

Rappelons que les sommets du graphe G sont numérotés de 1 à N : $\chi = (X_1, \dots, X_N)$. L'algorithme de Dijkstra détermine le plus court chemin du sommet X_1 à tous les autres sommets du graphe (il donnera donc la première ligne de la matrice de coût minimum). On construit un vecteur $\pi = (\pi(1), \pi(2), \dots, \pi(N))$ ayant n composantes telles que $\pi(i)$ soit égal à la longueur du plus court chemin du sommet 1 au sommet i . On initialise ce vecteur à la première ligne de la matrice de coût du graphe M_c . On considère ensuite deux ensembles de sommets, S initialisé à $\{1\}$ et \bar{S} son complémentaire dans χ . A chaque pas de l'algorithme, on ajoute à S des sommets jusqu'à ce que $S = \chi$, de telle sorte que le vecteur π donne à chaque étape le coût minimal des chemins de X_1 aux sommets de S .

L'algorithme est le suivant :

Initialisations

$$\pi = (M_c(1, i)) \text{ pour } i \in \{1, 2, \dots, N\}$$

$$S = \{1\}$$

$$\bar{S} = \{2, 3, \dots, N\}$$

Itérations. Tant que $\bar{S} \neq \emptyset$

Choisir i dans \bar{S} tel que $\pi(i)$ est minimum

Retirer i de \bar{S} et l'ajouter à S

Pour chaque successeur j de i dans \bar{S} : $\pi(j) = \min\{\pi(j), \pi(i) + \gamma(i, j)\}$

L'algorithme de Dijkstra a une complexité en $O(N^2)$.

3.5.2.4 Stratégie de parcours de graphe proposée

A partir de la trajectoire à estimer, on construit un graphe, dont les sommets sont les points extrémaux ainsi que les points de rendez-vous de la trajectoire. Les arcs relient deux sommets s'il y a lieu de le faire. Ils sont orientés et la fonction de coût associée à chaque arc le nombre de mesures obtenues sur cet arc. Dans l'algorithme que nous avons mis au point, les boucles (arcs qui partent et reviennent au même sommet) sont traitées à la fin. D'autre part, il peut arriver qu'il y ait deux (ou plus) arcs de même orientation qui relient les deux mêmes sommets. Nous avons choisi d'inclure dans l'algorithme de parcours de graphe, le chemin le plus court parmi ces "doublons". Les chemins restants sont traités, comme les boucles, à la fin de l'algorithme.

L'algorithme est le suivant :

1. Construire la matrice de coûts du graphe
2. Rechercher le plus court chemin entre les extrémités connues grâce à l'algorithme de Dijkstra,
3. Réaliser l'estimation de ce plus court-chemin (méthode d'auto-estimation, filtrage de Kalman aller-retour, etc...),
4. Considérer comme certains les points de rendez-vous estimés à l'itération précédente,
5. Chercher les plus courts chemins à partir de ces points pris deux à deux,
6. Estimer les plus courts chemins ainsi déterminés,
7. Revenir à l'étape (4).

NB : La phase (6) d'estimation peut être réalisée grâce à plusieurs techniques : le filtrage de Kalman Aller-Retour (section 3.2.2), le lissage RTS (section 3.2.3), les moindres carrés (section 3.3), ou l'estimation que nous avons proposée (section 3.4).

Sur l'ensemble des figures 3.6(a) à 3.6(e) et 3.7(a) à 3.7(i), sont montrés deux exemples de parcours de trajectoire par la stratégie reposant sur le plus court chemin.

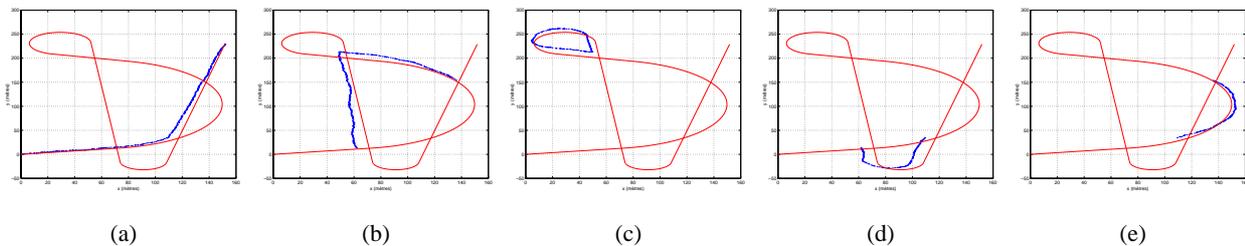


FIG. 3.6 – Exemple de stratégie de parcours de graphe sur la trajectoire quadrillée.

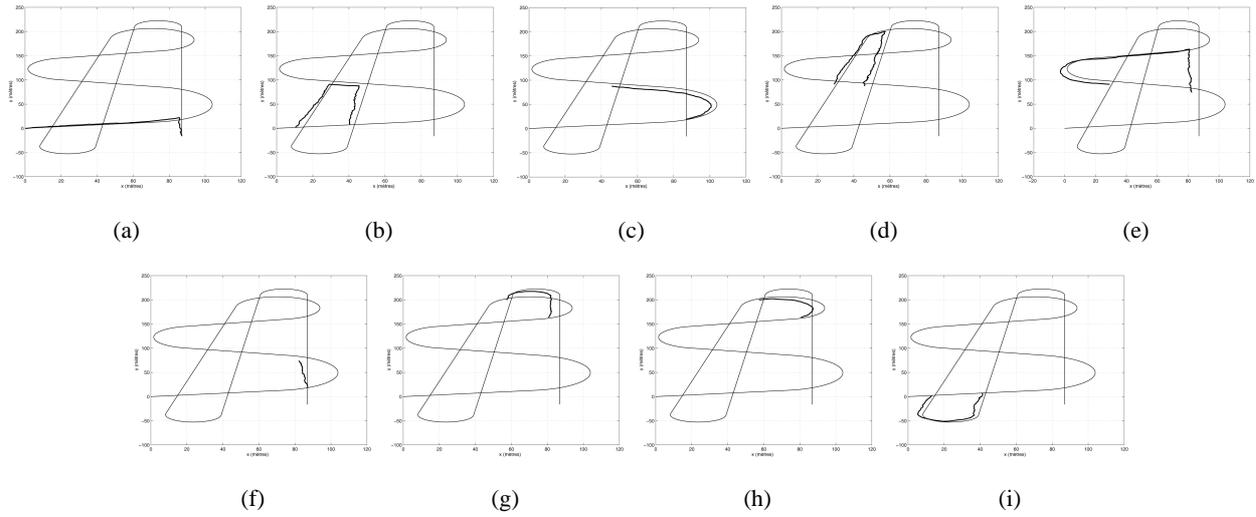


FIG. 3.7 – Exemple de stratégie de parcours par le plus court chemin sur la trajectoire "graphe".

3.5.3 Cas particulier de la méthode des moindres carrés

Pour le cas de l'estimation d'une trajectoire complexe par les moindres carrés, il existe deux solutions au traitement. La première consiste à utiliser une des deux stratégies proposées et à appliquer la méthode des moindres carrés sur chaque sous-trajectoire simple. La seconde est plus globale (donc théoriquement optimale). Elle consiste à estimer $\theta = (X_2, \dots, X_{N-1})$ en excluant les points de rendez-vous. Supposons la présence d'un seul instant de rendez-vous : on sait qu'aux instants I et J , $X_I = X_J$. Alors $\theta = (X_2, \dots, X_I, \dots, X_{J-1}, X_{J+1}, \dots, X_{N-1})$. Pour calculer les gradients, il convient de remarquer que

$$\begin{cases} c_{J-1} = \arctan\left(\frac{x_J - x_{J-1}}{y_J - y_{J-1}}\right) \\ c_J = \arctan\left(\frac{x_{J+1} - x_J}{y_{J+1} - y_J}\right) \end{cases} \quad (3.42)$$

En supposant que les positions aux instants I et J sont identiques, on a :

$$\begin{cases} c_{J-1} = \arctan\left(\frac{x_I - x_{J-1}}{y_I - y_{J-1}}\right) \\ c_J = \arctan\left(\frac{x_{J+1} - x_I}{y_{J+1} - y_I}\right) \end{cases} \quad (3.43)$$

Ainsi, les gradients deviendront :

$$\nabla_{\theta} c_{J-1}(\theta) = \left[0 \quad \dots \quad \frac{\partial c_{J-1}}{\partial x_I} \quad \frac{\partial c_{J-1}}{\partial y_I} \quad 0 \quad \dots \quad 0 \quad \frac{\partial c_{J-1}}{\partial x_{J-1}} \quad \frac{\partial c_{J-1}}{\partial y_{J-1}} \quad \dots \quad 0 \right]^T \quad (3.44)$$

$$\nabla_{\theta} c_J(\theta) = \left[0 \quad \dots \quad \frac{\partial c_J}{\partial x_I} \quad \frac{\partial c_J}{\partial y_I} \quad 0 \quad \dots \quad 0 \quad \frac{\partial c_J}{\partial x_{J+1}} \quad \frac{\partial c_J}{\partial y_{J+1}} \quad \dots \quad 0 \right]^T \quad (3.45)$$

Remarque : Étant donné que nous n'avons plus que $N - 3$ paramètres à estimer, il y aura un décalage de 2 à partir de l'indice J par rapport au cas précédent (sans instant de rendez-vous).

Le principe est le même dans le cas de plusieurs instants de rendez-vous.

3.6 Conclusion

Dans ce chapitre, nous avons décomposé le problème de localisation de l'engin à partir de mesures de cap et vitesse et d'estimations des instants de rendez-vous en deux sous-problèmes pour plus de simplicité. Nous avons tout d'abord considéré une trajectoire simple, c'est-à-dire sans tenir compte des instants de rendez-vous estimés. Pour réaliser l'estimation des positions de l'engin sous-marin, nous avons tout d'abord proposé des méthodes d'estimation classiques, comme le filtrage de Kalman ou le lissage RTS. Nous avons proposé une méthode originale : l'auto-estimation.

Dans un deuxième temps, nous avons considéré une trajectoire complexe, qui représente la trajectoire de l'engin estimée en tenant compte des instants de rendez-vous estimés. Pour cela, nous avons proposé de décomposer la trajectoire complexe en une liste de trajectoires simples connexes. Cet ensemble de trajectoires simples est trié. Nous avons pour cela proposé deux méthodes : l'une reposant sur un parcours "par l'extérieur", l'autre reposant sur une stratégie de parcours de graphe (plus court chemin). Enfin, l'estimation de ces trajectoires par l'une ou l'autre des méthodes d'estimation proposées dans la première partie peut être réalisée.

Dans le chapitre suivant, nous allons qualifier les méthodes proposées ici.

Chapitre 4

Etude des performances et résultats relatifs à la localisation AR/(C,V)

Ce chapitre présente les critères de qualification retenus pour l'évaluation des méthodes d'estimation de trajectoires que nous avons proposées dans le chapitre précédent. Dans un premier temps, une étude des performances sera réalisée, puis les résultats relatifs aux méthodes employées seront exposés. Enfin, ceux-ci permettront de justifier nos choix de méthodes.

4.1 Outils d'étude des performances

Après avoir présenté un outil qui permet d'évaluer les performances d'un estimateur en terme de variance, la Borne de Cramer-Rao (notée BCR), celui-ci sera évalué dans le cas d'une trajectoire simple, dans le cas d'une trajectoire complexe comportant un seul instant de rendez-vous, puis sera généralisé à une trajectoire complexe quelconque.

4.1.1 Matrice d'information de Fisher

La matrice d'information de Fisher permet de quantifier la quantité d'information pour une mesure z_k dans l'estimation du paramètre θ . Soit θ le paramètre à estimer. Soit z_k le signal mesuré à l'instant k , $h_k(\theta)$ étant le signal "vrai", non bruité, et ϵ_k l'erreur perturbant le système :

$$z_k = h_k(\theta) + \epsilon_k \quad (4.1)$$

Dans le cas de mesures gaussiennes, l'information de Fisher apportée par la mesure z_k est donnée par :

$$I_\theta(z_k) = \frac{\nabla_\theta h_k(\theta) \cdot \nabla_\theta^T h_k(\theta)}{\sigma_k^2} \quad (4.2)$$

Plus généralement, lorsque les mesures z_k sont indépendantes dans leur ensemble, l'information de Fisher globale est donnée par :

$$I_\theta(z_1, \dots, z_n) = \sum_{k=1}^n I_\theta(z_k) \quad (4.3)$$

Cette matrice carrée a la propriété d'être symétrique et définie positive (sauf si θ n'est pas observable).

4.1.2 Borne de Cramer-Rao

La borne de Cramer-Rao (BCR) permet de donner une borne inférieure à la matrice de variance-covariance de tout estimateur [BSL93]. Pour les estimateurs sans biais, elle est définie par :

$$BCR_{\theta} = I_{\theta}^{-1}(z_1, \dots, z_n) \quad (4.4)$$

Ainsi, la détermination de la BCR permettra d'évaluer les meilleures performances possibles de notre méthode : les éléments diagonaux de la BCR représentent les variances optimales obtenues lors d'une estimation.

4.1.3 Calcul dans le cas d'une trajectoire simple

Dans cette partie, nous allons calculer théoriquement la BCR dans le cas d'une trajectoire simple. Soit $X_k = (x_k, y_k)^T$ la position de l'engin à l'instant de mesure t_k . Soit $(\tilde{c}_k)_{k=1:N-1}$ et $(\tilde{v}_k)_{k=1:N-1}$ les mesures bruitées en cap et vitesse et $N - 1$ le nombre de mesures disponibles. Ici, le paramètre θ à estimer est le suivant :

$$\theta = [X_2 \quad \dots \quad X_k \quad \dots \quad X_{N-1}]^T \quad (X_1 \text{ et } X_N \text{ sont connus}) \quad (4.5)$$

Les mesures z_k définies dans l'équation (4.1) sont de deux types : les caps (c_1, \dots, c_{N-1}) et les vitesses (v_1, \dots, v_{N-1}) , dont les relations avec θ sont données dans l'équation (3.4). Nous allons noter $(z_1, \dots, z_{2N-2}) = (z_{c,1}, \dots, z_{c,N-1}) \cup (z_{v,1}, \dots, z_{v,N-1})$ avec $(z_{c,1}, \dots, z_{c,N-1}) = (c_1, \dots, c_{N-1})$ et $(z_{v,1}, \dots, z_{v,N-1}) = (v_1, \dots, v_{N-1})$. Posons

$$z_{i,k} = h_{i,k}(\theta) + \epsilon_{i,k} \quad i \in \{c, v\} \quad (4.6)$$

Les mesures \tilde{c}_k et \tilde{v}_k sont indépendantes dans leur ensemble (car les bruits sont indépendants), donc on peut calculer la matrice d'information de Fisher de la façon suivante :

$$I_{\theta}(z_1, \dots, z_{2N-2}) = \sum_{k=1}^{2N-2} I_{\theta}(z_k) = \sum_{k=1}^{N-1} \left[\frac{1}{\sigma_c^2} \nabla_{\theta} c_k(\theta) \nabla_{\theta}^T c_k(\theta) + \frac{1}{\sigma_v^2} \nabla_{\theta} v_k(\theta) \nabla_{\theta}^T v_k(\theta) \right] \quad (4.7)$$

La matrice d'information de Fisher est une matrice carrée, dont le nombre de lignes (ou de colonnes) est donné par la dimension du paramètre θ , c'est-à-dire $2(N - 2) = 2N - 4$ ($N - 2$ positions X soit $(N - 2)x$ et $(N - 2)y$). Pour déterminer l'information de Fisher globale, penchons-nous tout d'abord sur le calcul de l'information de Fisher "partielle", pour une mesure z_k , ici nous allons faire les calculs pour c_k . Le gradient a été calculé dans la partie 3.3.2.

On peut donc calculer pour $k \in \{2, \dots, N - 2\}$, la contribution de c_k dans la matrice d'information de

Fisher. Pour cela, par souci de simplicité, posons :

$$\left\{ \begin{array}{l} \frac{\partial c_k}{\partial x_k} = a \\ \frac{\partial c_k}{\partial y_k} = b \\ \frac{\partial x_{k+1}}{\partial c_k} = c \\ \frac{\partial y_{k+1}}{\partial c_k} = d \end{array} \right. \quad (4.8)$$

a, b, c, d étant donnés par les équations (3.28). L'expression de la contribution de c_k dans la matrice d'information de Fisher est la suivante :

$$\nabla_{\theta} c_k(\theta) \nabla_{\theta}^T c_k(\theta) = \left[\begin{array}{cccccccc|c} 1 & \dots & 2k-3 & 2k-2 & 2k-1 & 2k & \dots & N-1 & \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & a^2 & ab & ac & ad & 0 & 0 & 2k-3 \\ 0 & 0 & ab & b^2 & bc & cd & 0 & 0 & 2k-2 \\ 0 & 0 & ac & bc & c^2 & cd & 0 & 0 & 2k-1 \\ 0 & 0 & ad & bd & cd & d^2 & 0 & 0 & 2k \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & N-1 \end{array} \right] \quad (4.9)$$

Pour chaque $k \in \{2, \dots, N-2\}$, la contribution de c_k (et de v_k aussi bien car son gradient est de la même forme) est une sous-matrice bloc diagonale carrée symétrique de taille 4×4 .

Il reste maintenant à déterminer les contributions des mesures aux instants $k = 1$ et $k = N-1$. Posons

$$\nabla_{\theta} c_1(\theta) = (a, b, 0, \dots, 0)^T \text{ et } \nabla_{\theta} c_{N-1}(\theta) = (0, \dots, 0, c, d)^T$$

$$\nabla_{\theta} c_1(\theta) \nabla_{\theta}^T c_1(\theta) = \left[\begin{array}{cccccc} a^2 & ab & 0 & \dots & 0 \\ ab & b^2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \quad (4.10)$$

$$\nabla_{\theta} c_{N-1}(\theta) \nabla_{\theta}^T c_{N-1}(\theta) = \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & c^2 & cd \\ 0 & \dots & 0 & cd & d^2 \end{array} \right] \quad (4.11)$$

Ces deux matrices $(2N-4) \times (2N-4)$ sont également composées de sous-matrices blocs diagonales carrées symétriques de taille 2×2 , dont les termes diagonaux sont positifs.

Ainsi, la contribution des mesures $(z_{c,1}, \dots, z_{c,N-1})$ dans la matrice d'information de Fisher est une matrice

bloc diagonale symétrique, dont les termes diagonaux sont positifs.

Les calculs sont identiques pour les mesures $(z_{v,1}, \dots, z_{v,N-1})$ (en l'occurrence, les vitesses). Ainsi, la forme de la matrice de Fisher est une matrice bloc diagonale symétrique dont les termes diagonaux sont positifs, dans le cas d'une trajectoire simple.

4.1.4 Calcul dans le cas d'une trajectoire complexe à un seul instant de rendez-vous

Nous allons exposer l'idée pour un seul instant de rendez-vous, puis nous généraliserons dans le paragraphe suivant.

Soient I et J les indices pour lesquels l'engin repasse par la même position. Il suffira alors d'estimer seulement le point X_I (étant donné que $x_J = x_I$ et $y_J = y_I$). En effet, on aura donc $N - 3$ paramètres à estimer :

$$\theta = [X_2 \quad \dots \quad X_I \quad \dots \quad X_{J-1} \quad X_{J+1} \quad \dots \quad X_{N-1}]^T \quad (4.12)$$

La matrice d'information de Fisher aura donc une dimension inférieure, c'est-à-dire $2(N - 3) \times 2(N - 3)$.

A partir du calcul de gradients vu en partie 3.5.3, on forme la matrice d'information de Fisher.

On peut remarquer que le paramètre θ à estimer comportant un terme de moins que précédemment (X_J), la contribution de $\nabla_{\theta} c_{J-1}(\theta)$ et $\nabla_{\theta} c_J(\theta)$ met en jeu les paramètres X_I , X_{J-1} et X_{J+1} . Ainsi, sa contribution dans la matrice d'information de Fisher se manifeste par la présence de deux blocs extra-diagonaux. La structure est présentée sur la figure 4.1.

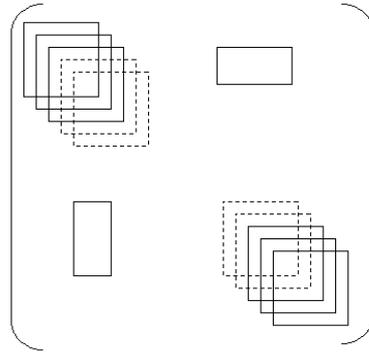


FIG. 4.1 – Structure de la matrice de Fisher dans le cas d'une trajectoire à un seul instant de rendez-vous.

4.1.5 Généralisation

Comme dit précédemment, la taille de la matrice de Fisher dépend de la taille du paramètre θ à estimer. Ainsi, si l'on note N_{RV} le nombre d'instant de rendez-vous, la dimension de la matrice de Fisher sera $2(N - 2) - 2N_{RV}$. Cette matrice sera constituée de blocs diagonaux ainsi que de blocs répartis symétriquement, de part et d'autre de la diagonale. Nous allons à présent décrire comment remplir cette matrice. Nous avons remarqué dans la partie précédente que dès qu'un indice de rendez-vous était dépassé, il y avait un décalage. Le remplissage va donc être effectué de façon séquentielle. Soit i l'indice courant. Soit I et J les instants de

rendez-vous courants.

Pour $i = 1$ à $N - 1$

- si $i = J - 1$ calculer $\nabla_{\theta} c_{J-1}$, en fonction de X_I et X_{J-1} .
- si $i = J$ calculer $\nabla_{\theta} c_J$, en fonction de X_I et X_{J+1} . Changer de J . Incrémenter le nombre d'instant de rendez-vous courant.
- sinon, calculer $\nabla_{\theta} c_i$, en tenant compte du nombre d'instant de rendez-vous courant.

Ensuite, la matrice d'information de Fisher est obtenue à partir des gradients en c et v et des variances α et σ_v (voir équation 4.7). La borne de Cramer-Rao est calculée en inversant la matrice de Fisher.

4.1.6 Performances désirées d'un estimateur

Les performances espérées d'un estimateur sont un **biais nul** (ou négligeable) et une **matrice de variance-covariance** la plus **proche possible de la BCR**.

Cependant, d'autres critères tels qu'un coût calculatoire faible et l'automatisation seront également pris en compte.

4.2 Contexte d'évaluation des méthodes

Des trajectoires en cap et vitesse ont été simulées. Pour générer les mesures de cap et vitesse, un bruit blanc a été ajouté à des données de cap et vitesse "vraies". Puis, grâce aux estimateurs proposés dans le chapitre précédent (chap. 3), des estimations de trajectoires en position (localisations AR/(C,V)) ont été obtenues. Ces méthodes d'estimation ont été évaluées afin de déterminer un estimateur qui présenterait les performances de biais négligeables et de variances proches de celles déterminées par la Borne de Cramer-Rao.

Plusieurs jeux de résultats vont être présentés. Tout d'abord, nous évaluerons chaque estimateur proposé sur une ou plusieurs trajectoires fixées, au regard des critères présentés (biais négligeables, variances proches de celles déterminées par la BCR). Un estimateur sera retenu. Puis nous évaluerons la sensibilité de l'estimateur choisi à la modélisation du bruit (gaussien ou autre). Nous confronterons ensuite les deux stratégies de parcours de trajectoire complexe. Enfin, une étude de l'estimateur proposé sera effectuée sur les différents types de trajectoire simulées.

Nous avons fait le choix de ne pas présenter de résultats sur données réelles (données gyro-doppler acquises pendant une plongée) car la comparaison avec la trajectoire véritablement suivie était impossible. La qualification d'une méthode n'aurait pas pu être réalisée.

4.2.1 Hypothèses sur le bruit

Bruit gaussien

Les algorithmes d'estimation proposés reposant sur l'hypothèse de gaussiennité du bruit, nous avons tout d'abord voulu appliquer un bruit gaussien sur nos caps et vitesses simulés.

Bruit uniforme

Dans le but d'évaluer la robustesse de l'algorithme proposé à d'autres types de bruit, des simulations ont été effectuées en entachant les caps et vitesses d'un bruit blanc uniforme au lieu du bruit blanc gaussien, hypothèse de base de l'estimateur. Le bruit uniforme est à support compact contrairement au bruit gaussien. De ce point de vue, ces deux types de bruit sont très "éloignés", à l'opposé l'un de l'autre. Si l'algorithme proposé fonctionne pour des simulations utilisant ce type de bruit, on pourra raisonnablement conclure qu'il fonctionnera pour différents autres types de bruit blanc, à distribution symétrique et unimodale.

Mélange de bruit

Un mélange de bruit blanc gaussien et uniforme a été également ajouté afin de tester les performances de notre algorithme.

4.2.2 Présentation des trajectoires simulées

Les trajectoires que nous avons estimées à partir des caps et vitesses simulés sont de trois types. Les trajectoires "vraies", c'est-à-dire sans ajout de bruit sur les caps et vitesses, sont présentées sur les figures 4.2(a), 4.2(b) et 4.2(c). La trajectoire notée "alpha" est trajectoire complexe de 1277 points comportant 1 instant de rendez-vous, la trajectoire notée "quadrillée" est une trajectoire complexe de 1036 points, à 4 instants de rendez-vous et la trajectoire notée "graphe" est trajectoire complexe de 1246 points, à 11 instants de rendez-vous.

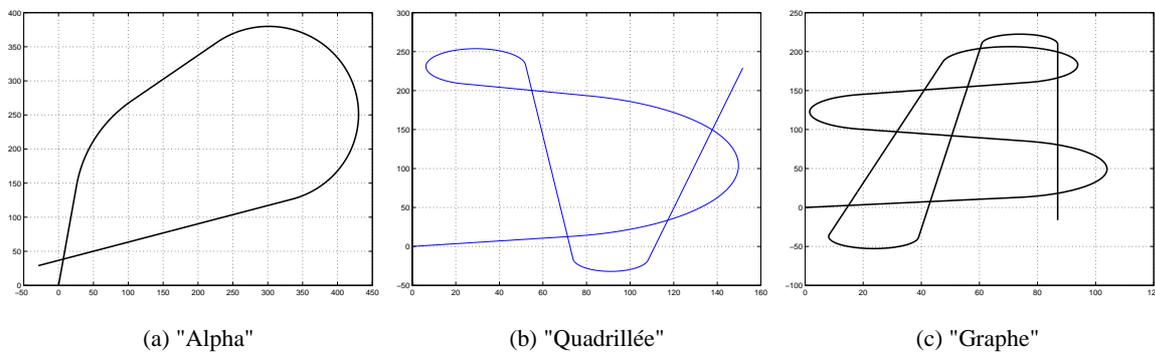


FIG. 4.2 – Trajectoires générées (en mètres).

4.3 Comparaison des algorithmes proposés

4.3.1 Kalman aller-retour et lissage RTS

L'état considéré est $X = (x, y, \dot{x}, \dot{y})^T$. Les initialisations sont les suivantes :

$$- R = \begin{pmatrix} \sigma_c^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix}$$

- État : filtre aller $X_{1|1}$ = position initiale et dérivées initiales (calculées par les premiers cap et vitesse),
filtre retour $X_{N|N}$ = position finale et dérivées finales,

- Matrice de variance-covariance : filtre aller $P_{1|1} = 10^{-6} I_4$, filtre retour : $P_{N|N} = 10^{-6} I_4$.

Les résultats sur les figures 4.3(a) et 4.3(b) montrent la trajectoire "vraie" en trait plein, la trajectoire obtenue à l'estime (après intégration des mesures bruitées) en pointillés '.', la trajectoire estimée par le filtrage de Kalman aller-retour en pointillés '-' et enfin la trajectoire estimée par le lissage RTS en pointillés '-.-'. Pour le cas de la trajectoire "alpha", notons que l'instant de rendez-vous n'a pas été pris en compte ici. La trajectoire est considérée comme simple.

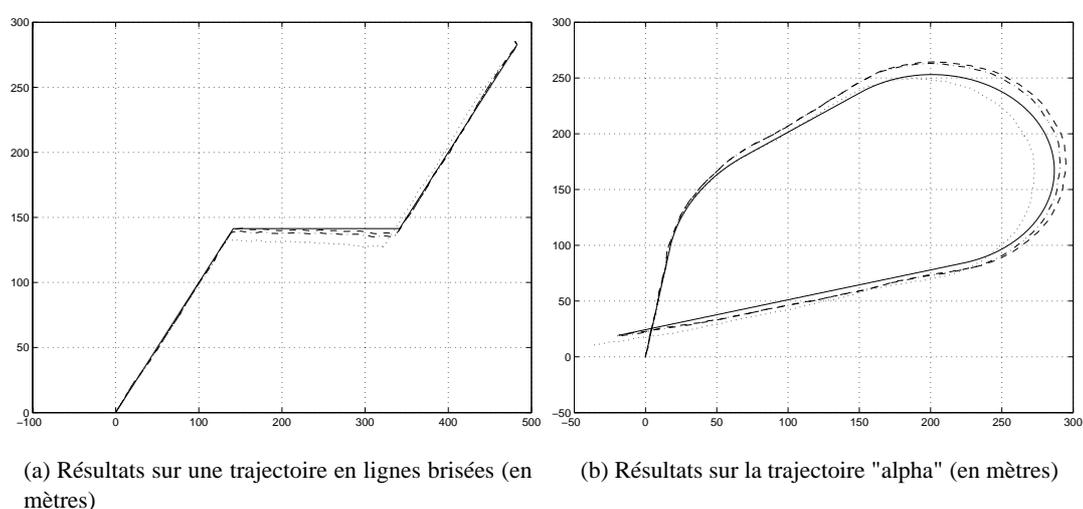


FIG. 4.3 – Comparaison entre EKF aller-retour et lissage RTS.

On remarque que les deux techniques rendent des résultats comparables. Sur la trajectoire en lignes brisées, on peut constater un petit décrochage au niveau des cassures de lignes. Dans l'ensemble, l'estimation des trajectoires est bonne.

Cependant, lors des expérimentations, la matrice de variance-covariance du bruit d'état a été réglée par l'opérateur afin que le filtre converge. En effet, sur l'exemple de la trajectoire "alpha", on peut clairement constater que la convergence (ou non) dépend du terme Q . Sur la figure (4.4), la trajectoire "vraie" est en trait plein, la trajectoire estimée par le filtre de Kalman étendu, en prenant $Q = 10^{-6}$ est en trait '-.-', la trajectoire estimée par un filtre de Kalman étendu, avec $Q = 10^{-2}$ est en pointillés. On remarque que la trajectoire estimée avec $Q = 10^{-6}$ "décroche" à un moment donné, tandis que celle estimée en prenant $Q = 10^{-2}$ suit bien la trajectoire vraie. Ainsi, l'intervention d'un opérateur pour régler Q est nécessaire. C'est un inconvénient non négligeable car le filtre n'est pas automatisable en pratique. La méthode a donc été abandonnée et les recherches de performance en terme de biais et variances n'ont pas été menées.

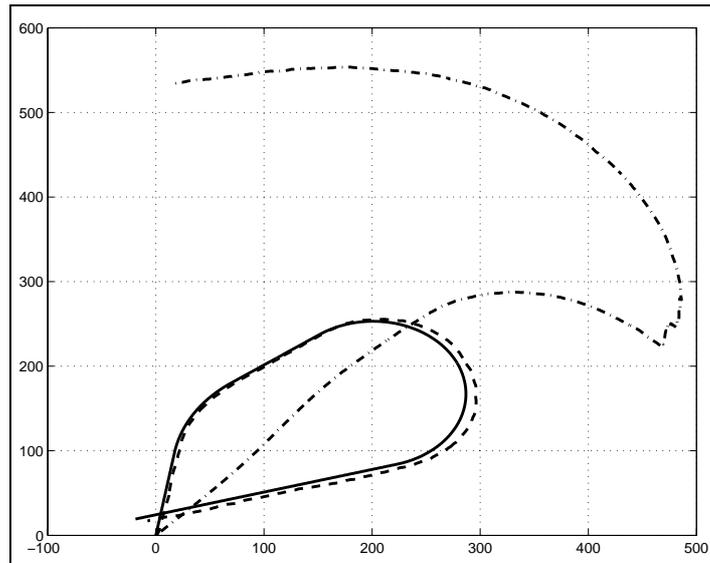


FIG. 4.4 – Problème de "décrochage".

4.3.2 Méthode des moindres carrés

Les résultats qui vont être présentés ont été obtenus sur deux types de trajectoires. L'une, "alpha", l'autre "quadrillée". L'étude a été réalisée volontairement sur des trajectoires comportant peu de points (85 pour alpha, 100 pour quadrillée) afin de favoriser la convergence. En effet, un trop grand nombre de mesures augmente la dimension de l'espace de recherche du minimum du critère des moindres carrés. D'autre part, notons que la méthode des moindres carrés gérait les instants de rendez-vous de façon directe, ceux-ci ont été pris en compte dans le traitement.

L'estimation a été réalisée en utilisant la méthode de Levenberg-Marquardt, associée au choix d'un pas. Le critère d'arrêt des itérations est soit la convergence de l'estimation, soit le dépassement du maximum fixé empiriquement à 20 itérations. Il convient de noter qu'un maximum d'itérations supérieur à 20 donne des résultats équivalents : des tests ont été effectués pour 30, 40 et 50. Le pas est choisi dans l'ensemble $\{0.1, 0.1 + \delta p, 0.1 + 2\delta p, \dots, 1\}$, avec $\delta p = 0.1$. Des résultats similaires ont été obtenus avec un δp inférieur, par exemple $\delta p = 0.01$.

Sur la figure 4.5 est représenté un exemple d'estimation de trajectoire, dans le cas d'un instant de rendez-vous. La courbe en pointillés représente l'initialisation θ_0 , la courbe en trait plein représente la trajectoire vraie (obtenue à partir des caps et vitesses non bruités) et le résultat de l'estimation en '-.-'. On peut remarquer une bonne convergence de la trajectoire estimée vers la trajectoire vraie.

Sur la figure 4.6, sont représentées : la trajectoire vraie en trait plein, la trajectoire bruitée en trait pointillés '-.', la trajectoire auto-estimée à partir des mesures bruitées en '+.-' et enfin, les trajectoires estimées par la méthode des moindres carrés, en '-.-'. Que l'on initialise la trajectoire par la trajectoire bruitée ou par la trajectoire auto-estimée, on obtient quasiment le même résultat (en '-.-'). La différence provient du

nombre d'itérations avant convergence (6 avec la trajectoire auto-estimée, 7 avec la trajectoire bruitée). Le critère est sensiblement le même (5.04 et 5.11).

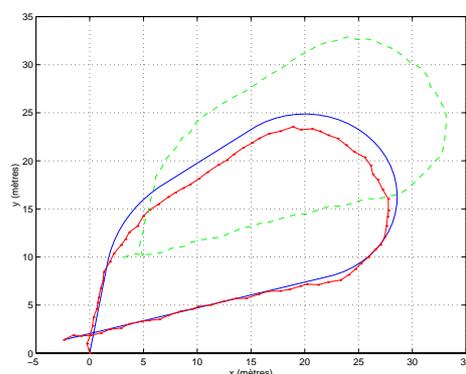


FIG. 4.5 – Exemple de trajectoire estimée dans le cas du "alpha".

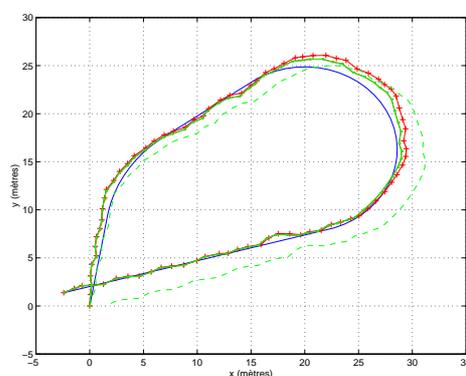
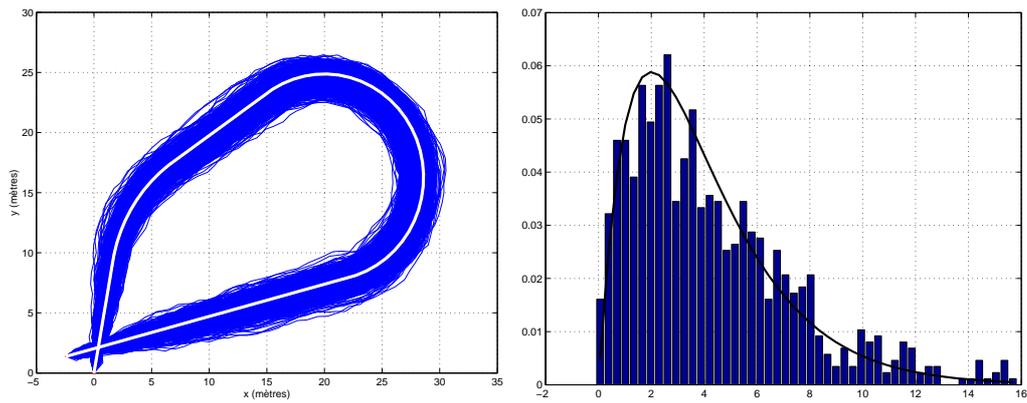


FIG. 4.6 – Comparaison des résultats obtenus avec des initialisations différentes dans le cas du "alpha".

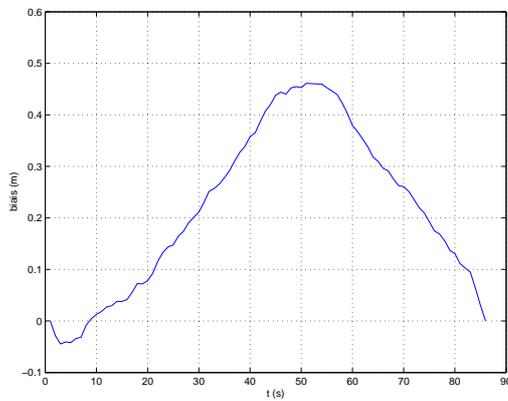
1000 estimations ont été effectuées à partir de 1000 jeux mesures de cap et vitesse. Les résultats sont consignés dans les figures 4.7. Pour cette trajectoire, l'algorithme de Levenberg-Marquardt présente 87% de taux de convergence. Ces 1000 estimations ont été réalisées en 1 h 25 min, ce qui revient à un temps de traitement moyen d'un peu plus de 5s par trajectoire. 500 trajectoires "quadrillée" ont été simulées. Les résultats obtenus sont montrés figure 4.8. Pour cette trajectoire, on observe seulement 26% de taux de convergence. D'autre part, les 500 simulations ont été réalisées en 1 heure 45, ce qui revient à un temps de traitement moyen de 12s par trajectoire.

Du point de vue des performances, les variances sont proches de la BCR et le critère $Q(\theta)$ suit la loi d'un χ^2 , ce qui confirme la théorie. Du point de vue des biais, ils ne vérifient pas le critère de biais négligeable. Ceci s'explique car la trajectoire "alpha" (il en va de même pour la trajectoire "quadrillée") étant "pauvrement" discrétisée, le phénomène présenté en partie 3.1.2.2 apparaît, c'est-à-dire que les positions X_I et X_J sur la trajectoire vraie n'étaient pas exactement superposés, alors que c'est une contrainte que nous avons imposée dans notre algorithme.

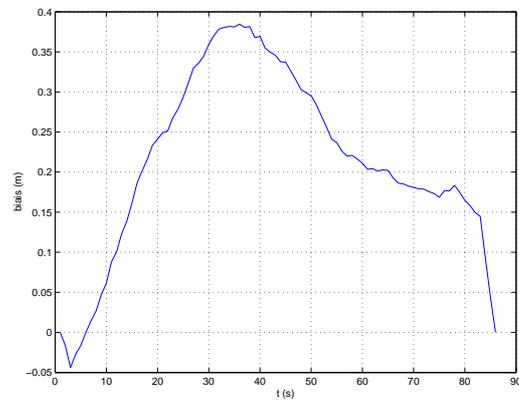
En revanche, le temps de calcul pour une estimation (supérieur à 5 secondes pour alpha, supérieur à 12 se-



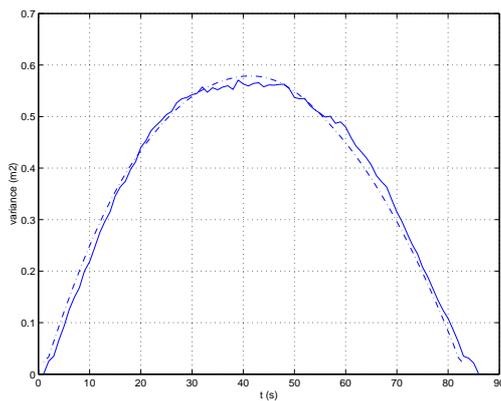
(a) Quelques exemples de résultats d'estimation.

(b) Comparaison de la loi du χ_4^2 (trait plein) à la statistique du critère sur 1000 trajectoires simulées (barres).

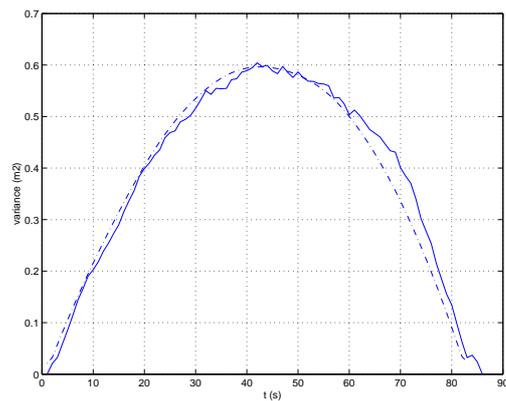
(c) Biais obtenus en x



(d) Biais obtenus en y

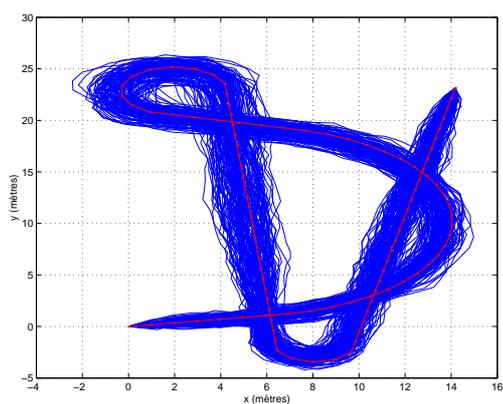


(e) Variances obtenues en x (trait plein) et BCR (en pointillés).

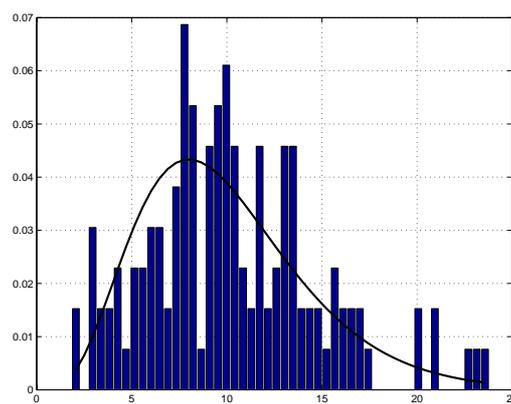


(f) Variances obtenues en y (trait plein) et BCR (en pointillés).

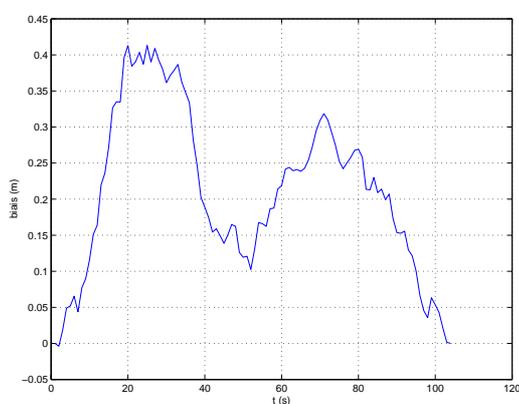
FIG. 4.7 – Estimation de la trajectoire "alpha" par les moindres carrés (1000 simulations).



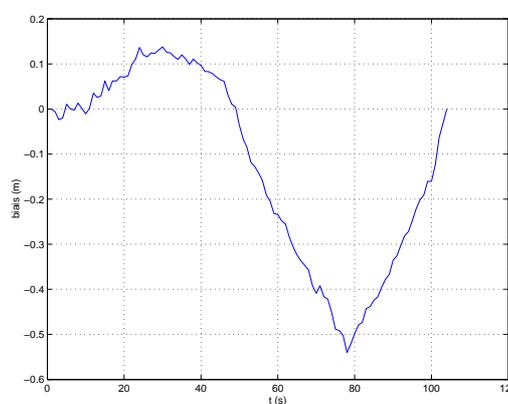
(a) Quelques exemples de résultats d'estimation.



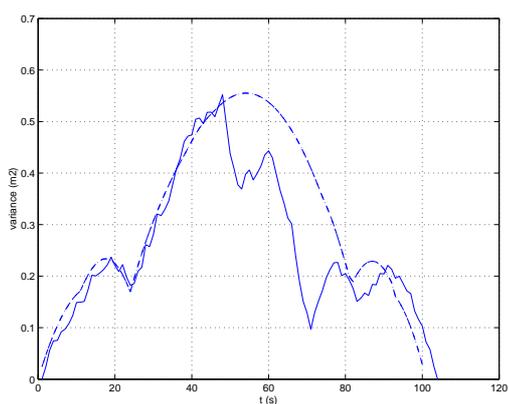
(b) Comparaison de la loi du χ^2_{10} (trait plein) à la statistique du critère sur 500 trajectoires simulées (barres).



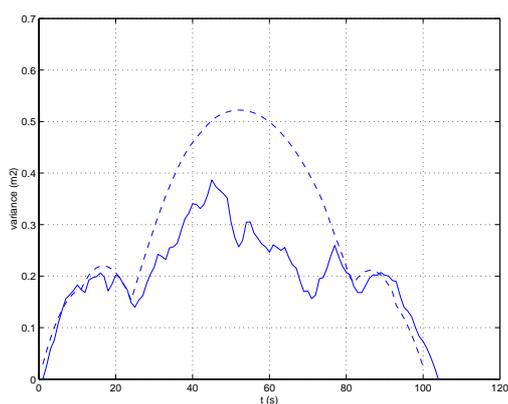
(c) Biais obtenus en x



(d) Biais obtenus en y



(e) Variances obtenues en x (trait plein) et BCR (en pointillés).



(f) Variances obtenues en y (trait plein) et BCR (en pointillés).

FIG. 4.8 – Estimation de la trajectoire "quadrillée" par les moindres carrés (500 simulations).

condes pour quadrillée) est assez élevé pour l'estimation de seulement 85-2(extrémités)-1(point de rendez-vous) = 83 points.

En terme de performance, des résultats équivalents ont été obtenus pour la trajectoire "quadrillée", comportant 4 instants de rendez-vous (variances proches de la BCR, critère suit la loi d'un χ^2). Cependant, le taux de convergence est beaucoup moins élevé que dans le cas d'un seul instant de rendez-vous (environ 30%), malgré un nombre de points à estimer à peine supérieur (94 au lieu de 83). La dimension de l'espace de recherche joue un rôle important dans la convergence de l'algorithme de Levenberg-Marquardt.

En pratique, les trajectoires sont très longues (de l'ordre de quelques milliers de mesures), contrairement à celles que nous avons utilisées ici, ce qui augmente la dimension de l'espace de minimisation du critère $Q(\theta)$. Ceci induit automatiquement un temps de calcul beaucoup plus important et une convergence loin d'être systématique dans un espace aussi grand. C'est pour ces raisons que la méthode, bien qu'étudiée et mise au point, a été abandonnée.

4.3.3 Méthode d'auto-estimation

L'estimation a été réalisée sur la trajectoire "alpha" est composée de 1277 points, dont un point de rendez-vous ($I = 40, J = 1240$), grâce à la méthode d'auto-estimation présentée dans la partie 3.4 et la stratégie de parcours de graphe, exposée section 3.5.2 (ce choix a été fait car pour cette trajectoire, les résultats avec les deux stratégies de parcours sont sensiblement les mêmes).

Les simulations sont réalisées avec les paramètres suivants pour la génération de bruits sur les caps et vitesses : $\sigma_c = 10^\circ$, $\sigma_v = 0.7\text{m/s}$, ce qui est beaucoup plus élevé que les valeurs typiques que l'on retrouve sur les capteurs en mer. Un faisceau de trajectoires estimées est présenté sur la figure 4.9.

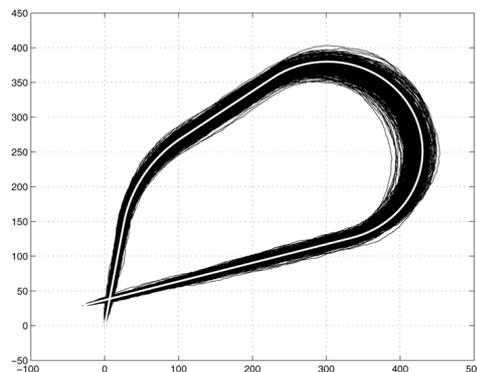


FIG. 4.9 – Faisceau de trajectoires estimées (en mètres). La trajectoire vraie est en blanc.

Les résultats sur l'efficacité de notre correcteur de biais sont présentés sur les figures 4.10(a) et 4.10(b). La fig.4.10(a) présente trois courbes : $\mathbf{E}[\tilde{x}_k|N] - x_k$, $\mathbf{E}[\hat{x}_k|N] - x_k$ et $b_x(k)$ en fonction de k . Sur cette figure, l'axe des abscisses est gradué de 0 à 1400s par pas de 200s et celui des ordonnées de -1 à 7m par pas de 1m. Avant débiaisage, on remarque une courbe en cloche qui culmine à 6.5m et qui est très proche de la courbe du biais théorique (en trait continu). Après débiaisage, le biais oscille entre $\pm 0.2\text{m}$. Les résultats sur les biais en y sont similaires (cf. fig. 4.10(b)).

Les variances sont illustrées sur les figures 4.11(a) et 4.11(b). L'axe des abscisses est gradué de 0 à 1400s par pas de 200s et celui des ordonnées de resp. 0 à 120m² par pas de 20m² (fig. 4.11(a)) et de 0 à 80m² par pas de 10m² (fig. 4.11(b)). Nous pouvons remarquer que les variances empiriques calculées avant réduction de biais dans l'estimation (en trait pointillés - sur la figure) et les variances calculées après réduction de biais dans l'estimation (en trait pointillés -) sont assez proches l'une de l'autre. Elles sont également très proches de la variance théorique (en trait plein) calculée dans le cas avant réduction de biais (voir annexe A.2.2). Enfin, les dernières figures 4.12(a) et 4.12(b) représentent la Borne de Cramer-Rao calculée dans le cas de la trajectoire "alpha" (trait plein) et la variance théorique (en trait pointillés) calculée sans réduction de biais. Dans le cas des variances en x , l'axe des abscisses est gradué de 0 à 1400, l'axe des ordonnées de 0 à 100m². On remarque que les variances théoriques sont très proches de la borne inférieure calculée grâce à la BCR ($\sigma_x < 10m$ et $\sigma_y < 8m$).

En résumé :

- Tout d'abord, le biais obtenu à partir des 5000 trajectoires simulées, estimées avant réduction de biais est proche du biais théorique.
- La variance obtenue par les 5000 simulations est proche de la variance théorique.
- La variance après correction de biais est à peu près égale à $e^{\sigma^2} = 1.0309$ la variance avant correction de biais (1.0310 pour x et 1.0313 pour y) (voir annexe A.3.2). Ainsi, la théorie et la pratique se rejoignent.
- D'autre part, le biais obtenu par les 5000 trajectoires estimées avec réduction de biais, est bien proche de 0. C'est une propriété de l'estimateur désiré.
- Enfin, la variance théorique (et donc la variance simulée) est proche de la BCR.

Ainsi, pour cette trajectoire, notre estimateur sans biais et dont les performances sont proches de de la BCR peut donc être considéré comme quasi-efficace.

4.3.4 Conclusion

Les méthodes Kalman aller-retour et de lissage RTS, pour des problèmes de stabilité, ainsi que la méthode des moindres carrés pour son faible taux de convergence et son fort coût algorithmique, ont été abandonnées. L'auto-estimation est simple et rapide d'utilisation. D'autre part, cette méthode satisfait aux contraintes de performances désirées (biais quasi-nul, variances proches de la borne de Cramer-Rao). Dans ce qui suit, nous allons donc présenter les résultats obtenus par l'auto-estimation.

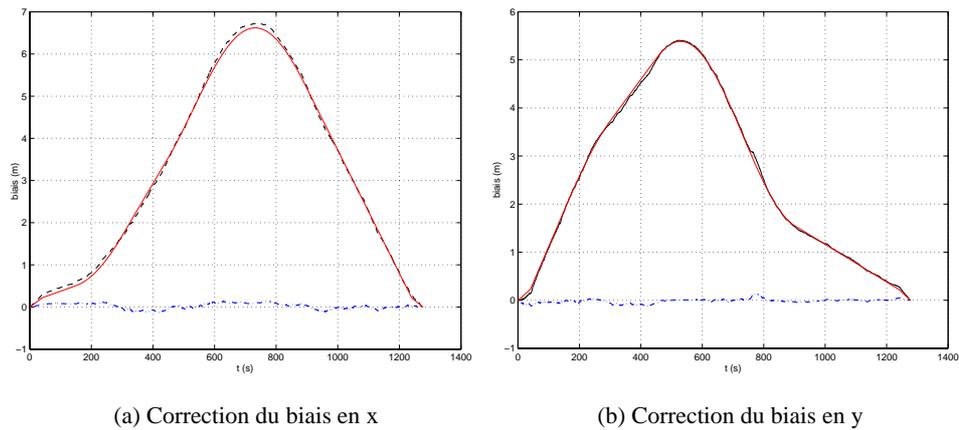


FIG. 4.10 – Comparaison des biais théoriques (sans correction de biais) et simulés sans et avec réduction du biais pour la trajectoire "alpha". En trait plein, le biais théorique, en pointillés, le biais non réduit, en "-.", le biais après réduction.

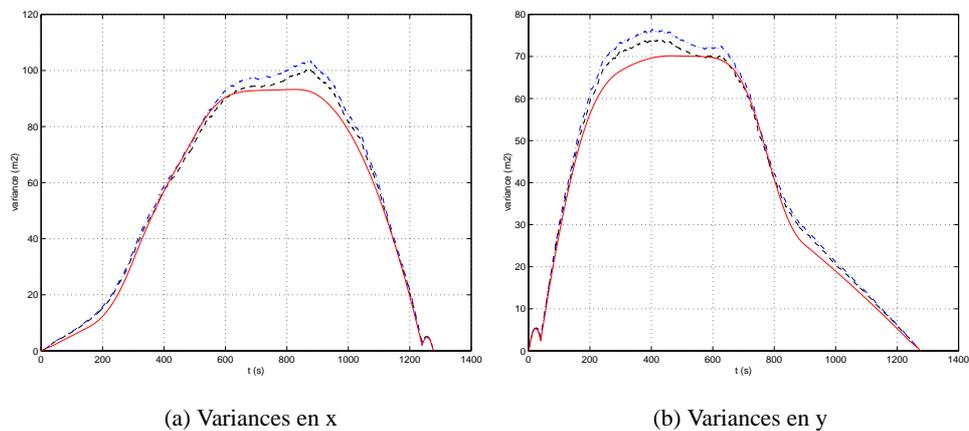


FIG. 4.11 – Comparaison des variances théoriques (sans correction de biais) et simulées sans et avec réduction du biais pour la trajectoire "alpha". En trait plein, la variance théorique, en pointillés, la variance sans réduction de biais, en "-.", la variance après réduction.

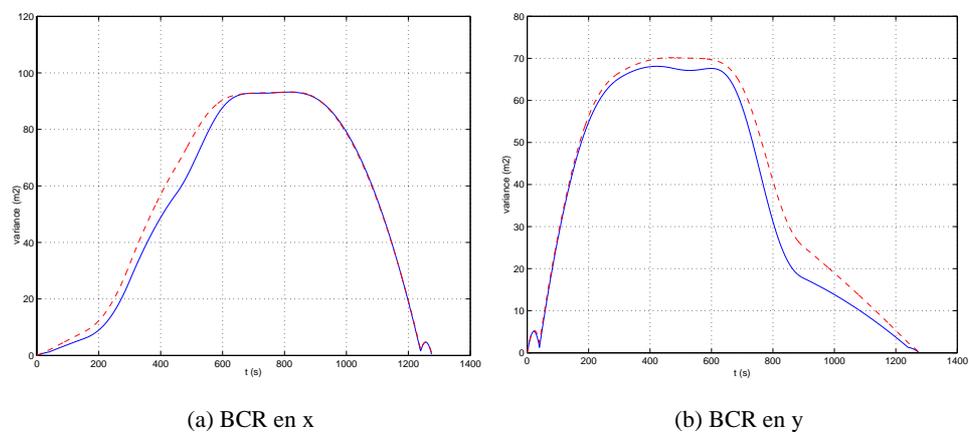


FIG. 4.12 – Comparaison des variances théoriques (sans correction de biais) et de la BCR pour la trajectoire "alpha". En trait plein, la BCR, en pointillés, la variance théorique.

4.4 Comparaison des résultats selon le type de stratégie employée

Nous avons choisi de faire les simulations sur la trajectoire quadrillée car elle est à la fois assez simple et comporte plus d'un instant de rendez-vous. Nous avons conservé les mêmes types de bruit que pour les simulations précédentes, c'est-à-dire des bruits gaussiens centrés, d'écart-type $\sigma_c = 10^\circ$ pour les caps et $\sigma_v = 0.7m/s$ pour les vitesses. La trajectoire quadrillée comporte 1036 points et 4 instants de rendez-vous : $I = \{73\ 124\ 252\ 353\}$, $J = \{707\ 837\ 955\ 518\}$.

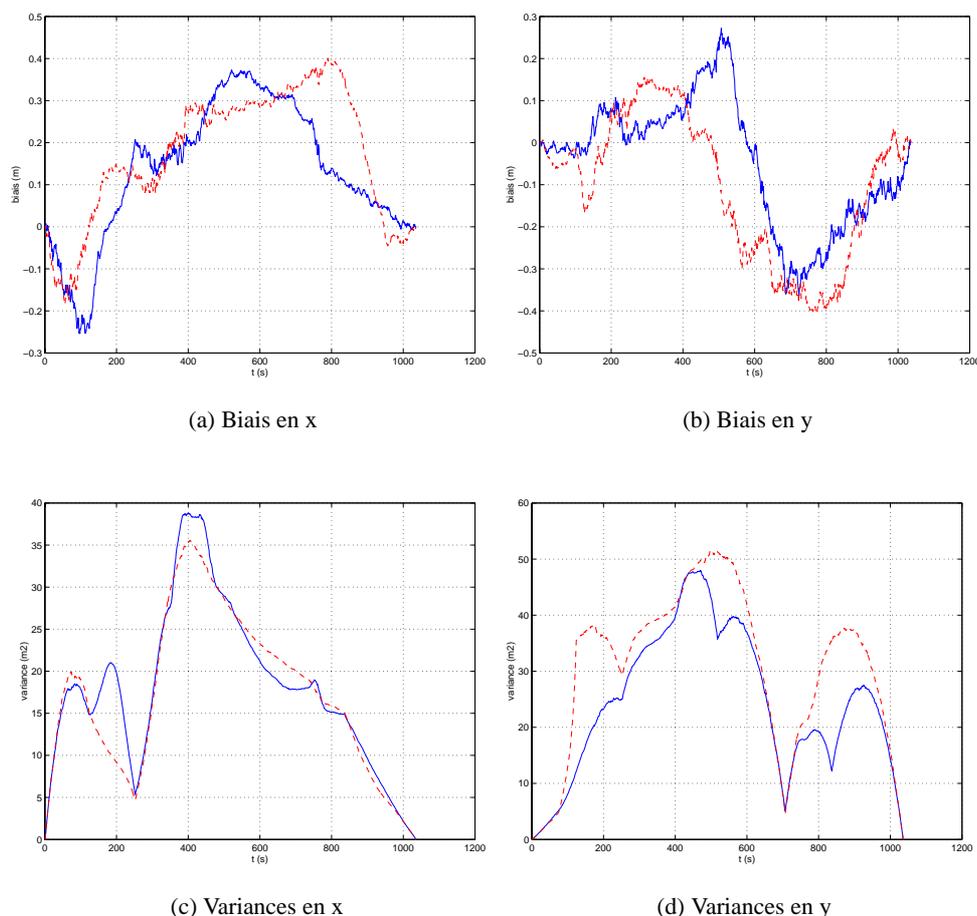


FIG. 4.13 – Comparaison des biais et variances selon la stratégie (extérieur : pointillés, graphe : trait plein).

Sur les figures 4.13(a) et 4.13(b) sont présentés les biais obtenus pour 5000 trajectoires simulées, en utilisant la stratégie de parcours par l'extérieur (en trait plein) et la stratégie de parcours par le plus court chemin (en trait pointillés). Nous pouvons remarquer que les biais calculés à partir des deux stratégies sont à peu près équivalents, et proches de 0. En revanche, on observe une différence notable sur les variances présentées figures 4.13(c) et 4.13(d). En effet, si les variances en x sont à peu près équivalentes (4.3% de différence), les variances en y montrent une différence de 25.4% : la stratégie reposant sur l'extérieur a une variance nettement plus élevée. Ainsi, avons-nous choisi dans la suite de conserver la stratégie de parcours de plus court chemin.

La stratégie par l'extérieur de la trajectoire commence par prendre l'enveloppe extérieure de la trajectoire pour l'auto-estimer. Or, cette trajectoire est censée être longue comparativement à la deuxième méthode que nous proposons (plus court chemin). Ainsi, l'erreur faite sur la première estimation -et qui se propagera puisque les positions des points de recalage ne seront plus réestimées par la suite- sera plus importante. D'autre part, cette stratégie ne gère pas bien les boucles puisque le recalage n'est pas pris en compte. Nous perdons donc de l'information pour la boucle.

4.5 Influence de la modélisation des bruits de mesure

L'estimateur que nous proposons repose sur l'hypothèse de gaussiennité des bruits additifs sur les caps et vitesses. Dans ce qui suit, nous allons donc comparer aux résultats obtenus pour la trajectoire "alpha" avec des bruits gaussiens, les résultats obtenus avec un bruit blanc uniforme et un mélange de bruits de lois gaussienne et uniforme respectivement.

4.5.1 Comparaison des résultats obtenus avec un bruit blanc uniforme

Les simulations ont été effectuées sur 5000 trajectoires en cap et vitesse. Le bruit uniforme appliqué sur les caps et vitesses vrais est centré et de même variance que le bruit gaussien. Sur la figure 4.14(a) sont présentés les biais sur la composante x , obtenus avant et après réduction de biais lors de l'estimation. Sur la figure 4.14(b), sont représentées les variances en x , calculées sans correction de biais (les résultats pour la composante y et pour les variances après correction de biais ne sont pas présentés ici car les conclusions sont identiques). Nous pouvons remarquer que les biais aussi bien que les variances sont quasiment superposables. Ainsi, notre estimateur, basé sur l'hypothèse d'un bruit additif gaussien, est robuste à l'emploi d'un bruit additif de loi uniforme.

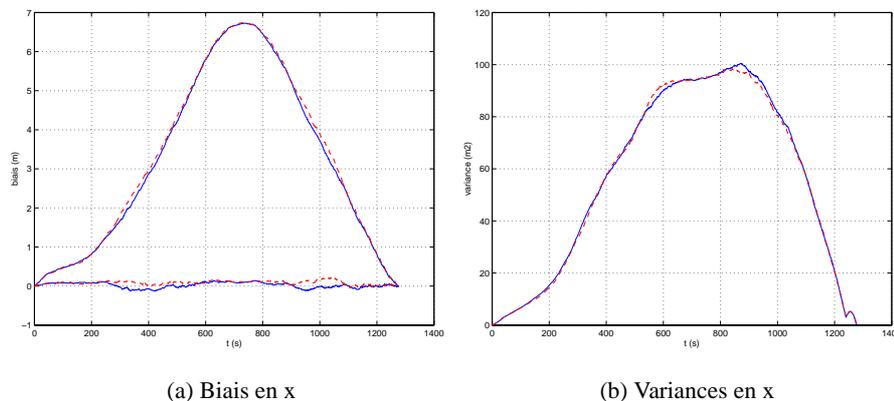


FIG. 4.14 – Comparaison des biais sans réduction de biais (courbes en cloche) et avec réduction de biais (proches de 0), et variances obtenus avec une loi gaussienne (trait plein) et une loi uniforme (trait pointillés), sur 5000 trajectoires "alpha".

4.5.2 Comparaison des résultats obtenus avec un mélange de bruits de lois différentes

Le mélange de bruit a été généré à partir d'un bruit de loi gaussienne et d'un bruit de loi uniforme. Soit N la longueur des caps et des vitesses. Nous avons procédé de la façon suivante pour les caps (resp. pour les vitesses) :

- Génération d'un bruit blanc quelconque, de taille N . Prenons l'exemple suivant, pour $N = 10$:

$$v = [-0.4940 \quad 0.8448 \quad -0.2251 \quad 1.6783 \quad 1.5202 \quad -0.9131 \quad 0.4086 \quad 1.1817 \quad -0.5409 \quad -1.3623]$$

- Tri de ce bruit blanc. On obtient les indices initiaux des éléments du vecteur, triés dans l'ordre croissant. Ex :

$$ind = [10 \quad 6 \quad 9 \quad 1 \quad 3 \quad 7 \quad 2 \quad 8 \quad 5 \quad 4]$$

- A partir des indices triés, on génère un vecteur de bruit. Si l'indice est inférieur à $\frac{N}{2}$, on génère le tirage d'un bruit gaussien, sinon, le tirage d'un bruit uniforme (de mêmes biais et variance). Dans le cas de notre exemple pour $N = 10$, si l'indice était inférieur à 5, on générerait un tirage gaussien.

Comme dans le cas précédent, les résultats sont présentés après les calculs sur 5000 trajectoires, sur les figures 4.15(a) (biais en x) et 4.15(b) (variances en x). Nous pouvons remarquer que l'ajout d'un bruit, mélange d'un bruit gaussien et d'un bruit uniforme, au lieu d'un bruit gaussien, sur les caps et les vitesses ne change pas les résultats obtenus sur les biais et variances. Ainsi, l'estimateur proposé est robuste au bruit ainsi construit.

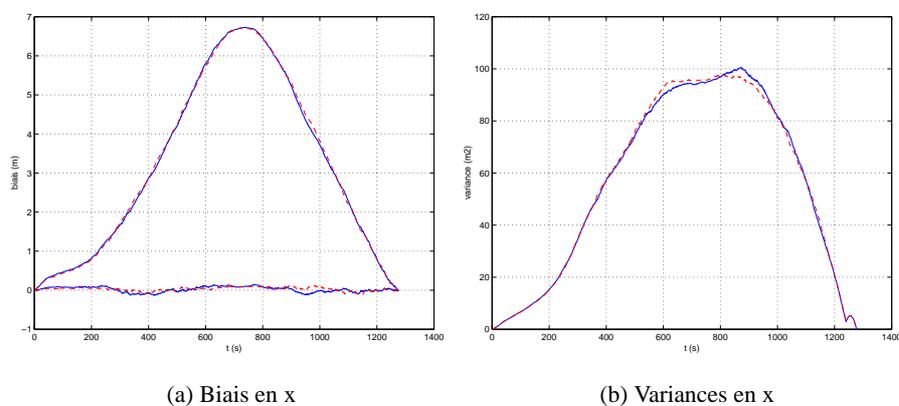


FIG. 4.15 – Comparaison des biais sans réduction de biais (courbes en cloche) et avec réduction de biais (proches de 0), et variances obtenus à partir d'une loi gaussienne (trait plein) et à partir d'un mélange de bruit (trait pointillés), sur 5000 trajectoires "alpha".

4.6 Résultats obtenus sur différentes trajectoires

Dans cette partie, nous montrons les limitations de l'auto-estimation. Pour cela, nous allons exposer les résultats obtenus à partir de 5000 trajectoires générées, sur les trajectoires "quadrillée" et "graphe".

4.6.1 Trajectoire quadrillée

La deuxième trajectoire que nous proposons pour les simulations est composée de 1036 points et de 4 instants de rendez-vous (les I sont 73, 124, 252, 353, les J correspondants sont 707, 837, 955 et 518). Un faisceau de trajectoires estimées est présenté sur la figure 4.16.

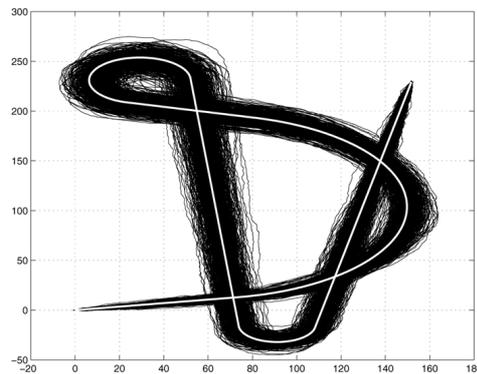


FIG. 4.16 – Faisceau de trajectoires estimées (en mètres). La trajectoire vraie est en blanc.

Les résultats sont présentés sur les figures 4.17(a) à 4.19(a). Comme pour la trajectoire "alpha", on peut remarquer sur les courbes 4.17(a) et 4.17(b) que les biais obtenus sur les simulations après réduction de biais oscillent autour de 0. D'autre part, sur les figures 4.18(a) et 4.18(b), les variances empiriques sont très semblables aux variances théoriques. Enfin, sur les figures 4.19(a) et 4.19(b), on remarque que les variances théoriques ont la même allure que les limites données par la borne de Cramer-Rao. Cependant, les variances théoriques sont plus élevées.

Sur les figures 4.18(a) à 4.19(b), nous avons représenté des couples de segments aux instants de rendez-vous (I, J) . Pour le premier instant de rendez-vous (73-707), le couples de segments est représenté par des +, pour le deuxième (124, 837), les segments sont représentées par des losanges \diamond , pour le troisième (252-955), par des croix \times , pour le quatrième (353-518), par des cercles \bullet . On peut remarquer que l'on vérifie bien que les variances théoriques, empiriques ou tirées de la BCR sont égales pour un couple d'instants de rendez-vous $(I, J) : Var_I = Var_J$.

D'après les résultats exposés, pour cette trajectoire, l'estimateur, bien que très acceptable, se révèle moins efficace que dans le cas de la trajectoire "alpha".

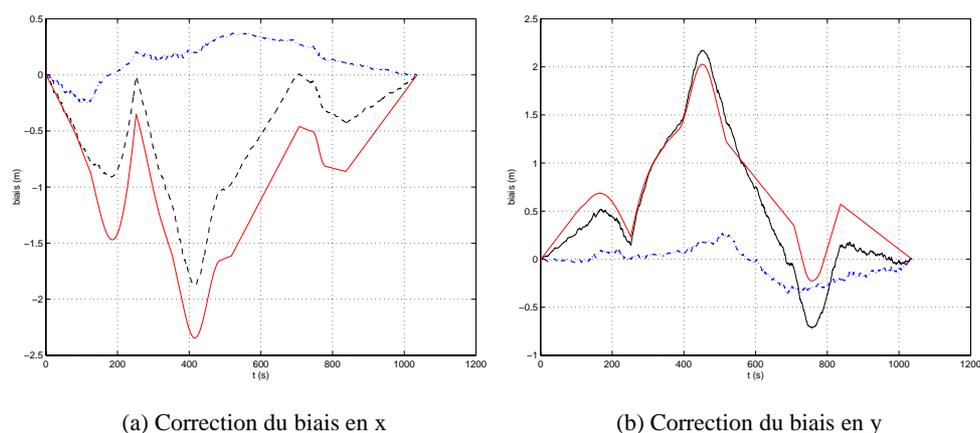


FIG. 4.17 – Comparaison des biais théoriques (sans correction de biais) et empiriques avant et après réduction du biais pour la trajectoire quadrillée. En trait plein, le biais théorique, en pointillés, le biais non réduit, en "-.", le biais après réduction.

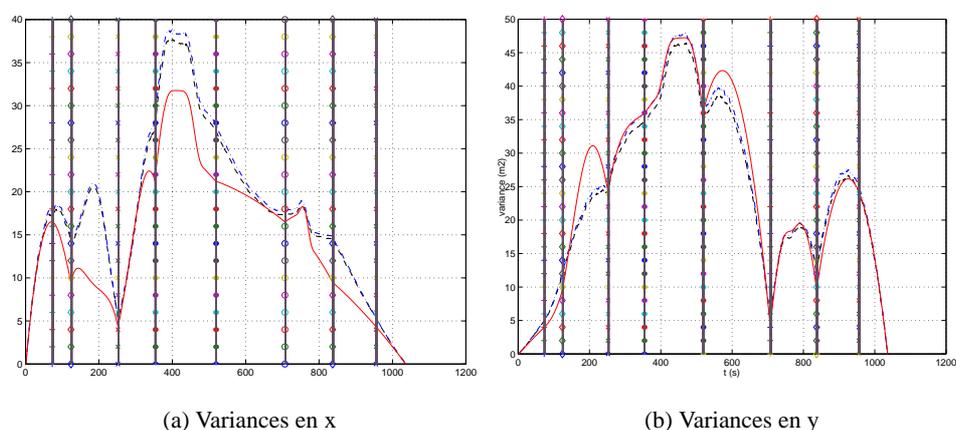


FIG. 4.18 – Comparaison des variances théoriques (avant correction de biais) et empiriques avant et après réduction du biais pour la trajectoire quadrillée. En trait plein, la variance théorique, en pointillés, la variance avant réduction de biais, en "-.", la variance après réduction.

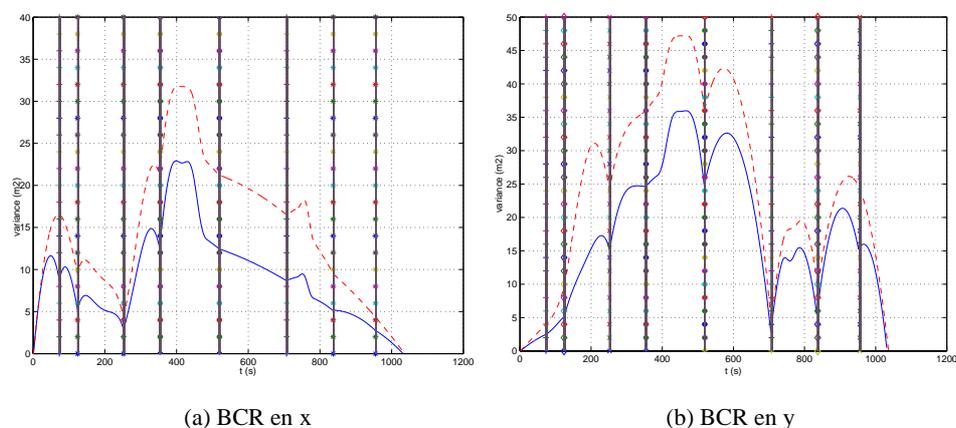


FIG. 4.19 – Comparaison des variances théoriques (avant correction de biais) et de la BCR pour la trajectoire quadrillée. En trait plein, la BCR, en pointillés, la variance théorique.

4.6.2 Trajectoire "graphe"

Cette trajectoire est composée de 1246 points, et 11 instants de rendez-vous (I : 16, 45, 90, 164, 202, 222, 319, 334, 367, 404, 433 , J correspondants : 643, 777, 1212, 1149, 862, 547, 492, 925, 1064, 1031, 974). Un faisceau de trajectoires estimées est présenté sur la figure 4.20.

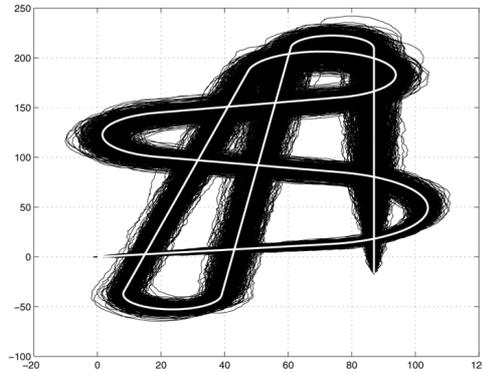


FIG. 4.20 – Faisceau de trajectoires estimées (en mètres). La trajectoire vraie est en blanc.

Les résultats sont présentés sur les figures 4.21(a) à 4.23(a). Par souci de lisibilité et de clarté des figures, nous n'avons pas représenté les segments correspondant aux couples d'instant de rendez-vous sur les figures obtenues pour la trajectoire "graphe", qui comporte 11 instants de rendez-vous (il y aurait 22 segments).

On peut remarquer sur les courbes 4.21(a) et 4.21(b) que les biais obtenus sur les simulations après réduction de biais oscillent autour de 0. D'autre part, sur les figures 4.22(a) et 4.22(b), les variances empiriques sont très semblables aux variances théoriques. Enfin, sur les figures 4.23(a) et 4.23(b), on remarque comme précédemment que les variances théoriques ont la même allure que les limites données par la borne de Cramer-Rao, bien que les variances théoriques soient plus élevées. Ainsi, pour cette trajectoire, l'estimateur, bien que très acceptable, se révèle moins efficace que dans le cas de la trajectoire "alpha".

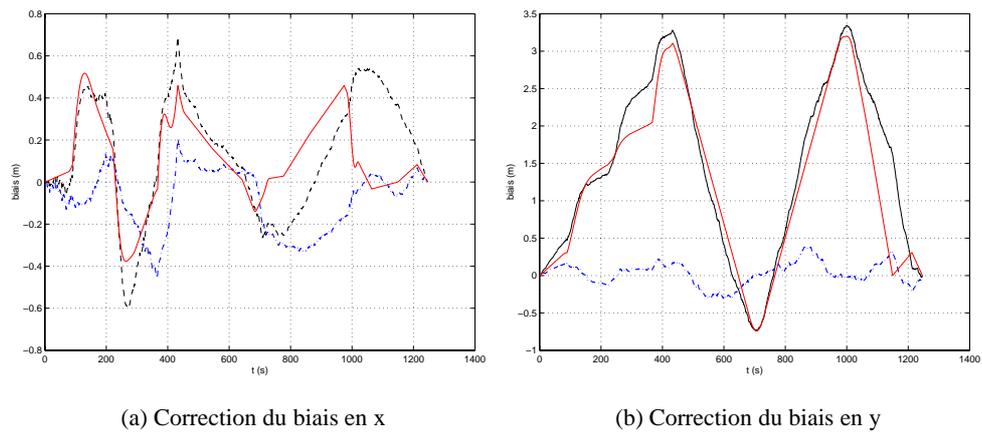


FIG. 4.21 – Comparaison des biais théoriques (sans correction de biais) et empiriques avant et après réduction du biais pour la trajectoire "graphe". En trait plein, le biais théorique, en pointillés, le biais non réduit, en "-", le biais après réduction.

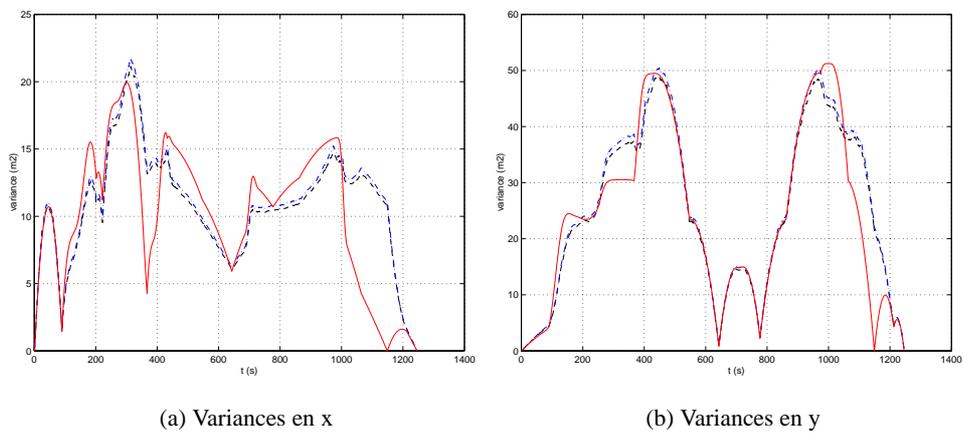


FIG. 4.22 – Comparaison des variances théoriques (avant correction de biais) et empiriques avant et après réduction du biais pour la trajectoire "graphe". En trait plein, la variance théorique, en pointillés, la variance avant réduction de biais, en "-", la variance après réduction.

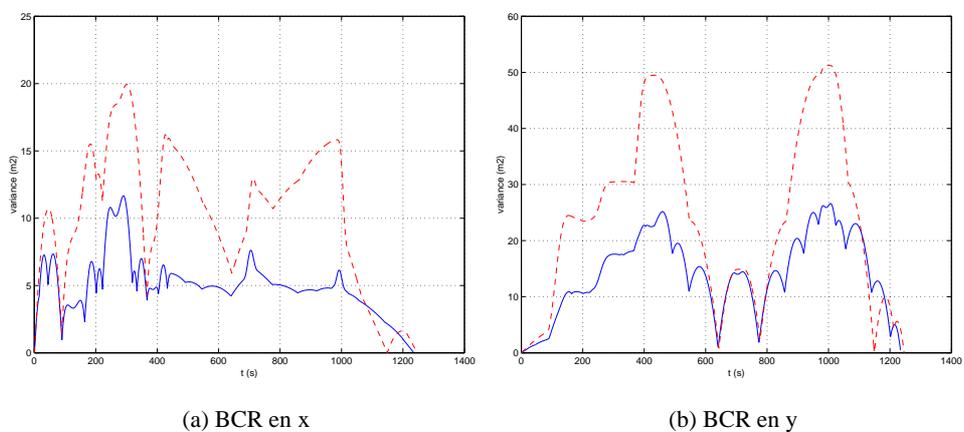


FIG. 4.23 – Comparaison des variances théoriques (avant correction de biais) et de la BCR pour la trajectoire "graphe". En trait plein, la BCR, en pointillés, la variance théorique.

4.7 Résumé et conclusions

Quelques résultats sont résumés sur les tableaux 4.1 et 4.2. Le tableau 4.1 présente les rapports moyens des variances calculées à partir des estimations sans réduction de biais et des variances calculées à partir des estimations avec réduction de biais. Soit N le nombre de points, $Var(\tilde{x}(k))$ la variance empirique de l'abscisse estimée après correction de biais, $Var(\hat{x}(k))$ la variance empirique de l'abscisse estimée avant correction de biais, le rapport, pour l'abscisse (la formule est identique pour les ordonnées), est donné par la formule :

$$Rapport_x = \frac{1}{N} \sum_{k=1}^N \frac{Var(\tilde{x}(k))}{Var(\hat{x}(k))} \quad (4.13)$$

D'après le calcul théorique (voir annexe A.3.3), ce rapport est égal à $e^{\sigma_c^2}$, ce qui donne dans nos simulations, en prenant $\sigma_c = 10^\circ$, $e^{\sigma_c^2} = 1.0309$. Les résultats consignés dans le tableau montrent donc que la théorie et la pratique se rejoignent.

| type de trajectoire | rapport sur x | rapport sur y |
|---------------------|-----------------|-----------------|
| alpha | 1.0310 | 1.0313 |
| quadrillée | 1.0247 | 1.0288 |
| graphe | 1.0301 | 1.0295 |

TAB. 4.1 – Rapport moyen entre variances obtenues avant et après réduction de biais.

Sur le tableau 4.2, sont exposés les résultats obtenus après le calcul d'une distance entre la limite donnée par la Borne de Cramer-Rao (notée BCR) et les variances théoriques (notées Var). Cette distance, notée L_2 , est définie de la façon suivante :

$$L_2 = \sqrt{\frac{1}{N} \sum_{k=1}^N \left(\frac{BCR(k) - Var(k)}{BCR(k)} \right)} \quad (4.14)$$

Ainsi, comme on avait pu le constater sur les courbes présentées dans les parties précédentes, plus on introduit d'instant de rendez-vous, moins l'estimateur est efficace.

| type de trajectoire | distance sur x | distance sur y |
|---------------------|------------------|------------------|
| alpha | 0.94 | 1.73 |
| quadrillée | 7.84 | 6.29 |
| graphe | 16.02 | 13.32 |

TAB. 4.2 – Distance L_2 entre BCR et la variance théorique.

Parmi les algorithmes de traitement d'une trajectoire simple, le filtrage de Kalman aller-retour et le lissage RTS sont des méthodes qui demandent un ajustement de certains paramètres par l'utilisateur, selon le système, ce qui est peu opérationnel. C'est pourquoi elles ont été abandonnées. En ce qui concerne la méthode des moindres carrés, elle serait une alternative intéressante si la dimension du paramètre à estimer

n'était pas aussi importante. Aussi, le coût algorithmique a été limitatif.

L'algorithme d'auto-estimation que nous avons proposé, en revanche, n'est pas coûteux en temps de calcul et ne nécessite pas d'ajustement de paramètre malgré une hypothèse de base de l'auto-estimation qui est la gaussiennité du bruit sur les caps et vitesses. En effet, les résultats que nous avons obtenus montrent que l'algorithme est robuste lors de l'introduction d'un autre type de bruit. Le tableau 4.3 présente une comparaison des différentes méthodes d'estimation évaluées.

Afin de traiter une trajectoire complexe, deux méthodes ont été proposées pour parcourir les trajectoires. Parmi elles, la plus performante (celle qui a été retenue) est celle qui utilise un algorithme du plus court chemin. Son coût algorithmique est relativement faible.

Dans le cas d'une trajectoire complexe, nous avons remarqué que l'efficacité de l'auto-estimateur dépend du nombre de recalages.

| | MC | Kalman A/R ou RTS | Auto-estimation |
|---------------|--|--|--|
| Performances | Variances proches de la BCR, biais résultant de l'échantillonnage | Non calculées | Biais quasi-nul, variances proches de la BCR |
| Avantages | Gestion directe des instants de recalage, sans mise en oeuvre de la stratégie de parcours : estimation "globale" | Rapide | Rapide, simplicité de la théorie, facilité de mise en oeuvre |
| Inconvénients | Lent, inapplicable en pratique | Gestion des matrices de variance-covariance d'état par l'opérateur | Choix d'une stratégie de parcours |

TAB. 4.3 – Comparaison des différents traitements.

Chapitre 5

Estimation des instants de rendez-vous et localisation à partir d'une séquence d'images

Dans les chapitres précédents ont été présentées les méthodes développées pour estimer la trajectoire en position de l'engin sous-marin, en 2 dimensions à partir de données de cap et de vitesse et d'instant de rendez-vous (localisation AR/(C,V)). Les instants de rendez-vous étaient supposés connus (estimés à partir des images). Nous allons donc tout d'abord présenter dans ce chapitre les méthodes utilisées pour estimer ces instants de rendez-vous. Puis nous nous intéresserons à la réalisation de la localisation de l'engin sous-marin à partir de la séquence d'images acquise lors de ses déplacements (localisation AR/I). Pour cela, nous présenterons quelques méthodes de construction de mosaïque utiles à la localisation AR/I.

Le plan de ce chapitre est le suivant : après une introduction aux problèmes inhérents à l'acquisition d'images sous-marines, quelques techniques de base permettant de déterminer la transformation entre deux images seront exposées. Elles sont utiles aussi bien à la recherche de couples d'images (correspondant aux couples d'instant de rendez-vous) qu'à la construction de mosaïques (utiles à la localisation AR/I).

Nous détaillerons ensuite l'algorithme que nous avons mis au point, permettant de rechercher les instants de rendez-vous à partir d'une séquence d'images, ainsi que nos contributions à l'intégration d'une méthode originale dans le logiciel que nous avons utilisé.

Puis quelques méthodes classiques de construction de mosaïque en 2 dimensions, à partir des images seules, seront exposées. Nous présenterons ensuite une étude pour réaliser la localisation AR/I.

Pour clore ce chapitre, des résultats relatifs à l'influence des bords de l'image et de la répartition de l'intensité lumineuse seront étudiés et exposés. Puis des résultats seront présentés dans le cas de la recherche d'instant de rendez-vous. Une comparaison des méthodes de construction de mosaïques permettra ensuite de sélectionner un algorithme pour réaliser la localisation AR/I et la fusionner (dans le chapitre suivant) avec la localisation AR/(C,V). Enfin, des premiers résultats au sujet de la localisation AR/I seront présentés.

5.1 Les images acquises en milieu sous-marin

Une image est la représentation discrète d'une scène 3D continue, par l'intermédiaire d'une matrice bidimensionnelle I qui à chaque pixel, associe une intensité. Par définition, un *pixel* (en anglais, "picture element") est le plus petit élément constituant la matrice d'intensité, obtenu par décomposition d'une image en lignes et en colonnes. La *résolution* d'une image décrit la précision atteinte lors de la retranscription du phénomène observé. Cela correspond à la distance (en mètres) qui sépare les centres de deux pixels adjacents. Le *rapport signal sur bruit* définit la qualité de l'image acquise. Il traduit l'importance du bruit affectant le signal acquis.

La *dynamique* d'une image correspond à l'écart entre les niveaux extrêmes du signal utile. Une forte dynamique signifie que l'appareil possède une palette importante pour traduire la scène imagée. Inversement, une faible dynamique ne peut pas retranscrire tous les détails de la scène.

En imagerie sous-marine, il existe trois types d'images numériques, selon le vocabulaire couramment employé :

- les images d'intensité lumineuse obtenues par des capteurs optiques comme les caméras vidéo et photo,
- les images de profondeur obtenues par des capteurs acoustiques, de type sondeur, ou par association de capteurs optiques appariés (stéréoscopie) [Gal02],
- les images acoustiques qui traduisent la réflectivité du sol, obtenues par des capteurs tels que le sonar actif.

5.1.1 Capteurs d'images

Les **images d'intensité lumineuse** sont acquises par des appareils photo ou des caméras vidéo, qui fournissent une représentation visuelle d'une scène 3D. Ils mesurent la quantité de lumière réfléchie et réfractée par les objets de la scène. Deux types d'images d'intensité lumineuse peuvent être acquises. L'une est appelée "image de niveau de gris" ou image noir et blanc, l'autre "image couleur". Une image noir et blanc est représentée par une matrice de niveaux de gris, typiquement d'intensité variant de 0 à 255 pour des images codées sur 8 bits. Une image couleur est composée de 3 couches, chacune représentant une composante primaire (rouge, vert, bleu) codée typiquement sur 256 niveaux. Dans le cadre de ce mémoire, nous nous intéresserons plus particulièrement aux images noir et blanc.

Les principaux capteurs embarqués sur les engins sous-marins sont les suivants :

- la caméra CCD (Charge Coupled Device) : c'est la caméra la plus employée sur les engins sous-marins. En milieu sous-marin, ces caméras ne sont utilisées qu'en eaux peu profondes ou bien, en grande profondeur, elles sont couplées à des projecteurs fixés sur l'engin sous-marin, remplaçant artificiellement la lumière naturelle.
- la caméra SIT (Silicon Intensifier Target) : c'est un capteur à tube qui utilise la technologie de la première génération des intensifiés. Sa sensibilité est plus importante que celle des CCD.
- la caméra ISIT (Intensified SIT) : l'intégration d'une intensification supplémentaire par rapport aux

caméras SIT améliore les performances de ce type de caméra en eaux troubles. Cependant, ces caméras peuvent rendre des images très bruitées en cas de zone très éclairée et sont très sensibles aux bruits électromagnétiques.

- la caméra ICCD (Intensified CCD) : basée sur la technologie CCD, cette caméra a des performances comparables aux ISIT.
- la caméra CCD à effet Peltier : le dispositif à effet Peltier permet de réduire la température des capteurs CCD. Ce type de caméra présente une dynamique plus importante.

Actuellement, les caméras les plus utilisées sont les caméras CCD, pour lesquelles des projecteurs sont ajoutés sur l'engin sous-marin.

Les **images de profondeur**, appelées aussi cartes bathymétriques, sont des images bidimensionnelles représentant les hauteurs du fond marin. Elles sont acquises par des sondeurs acoustiques "actifs", le plus répandu étant le sondeur multifaisceaux. Les sondeurs multifaisceaux utilisent un réseau de transducteurs placé sous la coque du bateau de surface, qui envoie des signaux acoustiques très brefs ("ping"). Connaissant le temps de trajet de l'onde et le profil de célérité du son dans l'eau, on déduit les profondeurs du point d'impact. D'autres capteurs de type sonar sont actuellement du domaine de la recherche appliquée : il s'agit du sonar interférométrique, qui exploite le déphasage des signaux à travers deux antennes superposées pour déduire la direction d'arrivée et du sonar à antenne synthétique (SAS), dont le principe réside dans l'intégration cohérente des signaux de voies d'une antenne réelle sur N récurrences successives constituant l'antenne synthétique [LLFZ03]. Généralement, le sondeur est couplé à un système de navigation afin de déterminer les positions des points d'impact des sondes et compenser les effets de tangage et de roulis subis par l'engin.

Les capteurs d'**images acoustiques**, tels que le sonar latéral, mesurent la réflectivité du sol par rapport à l'angle d'incidence du signal acoustique. L'onde émise par le capteur n'est rétrodiffusée que lorsqu'elle rencontre un obstacle. La fréquence de l'onde émise influe directement sur la détection des éléments constituant le fond (algues, roche, sable...). Ces images sont essentiellement utilisées pour la recherche d'objets (épaves, pipelines pour le domaine civil, mines pour le domaine militaire). Comme pour les sondeurs acoustiques, un système de navigation vient en général en complément du capteur.

Ces trois types d'images sont complémentaires puisque l'image d'intensité lumineuse peut représenter la scène telle qu'elle est vue par un œil humain, tandis que l'image de profondeur donne des informations sur le relief du fond marin et que l'image acoustique fournit une information sur la nature du sol.

5.1.2 Les problèmes dus aux grands fonds

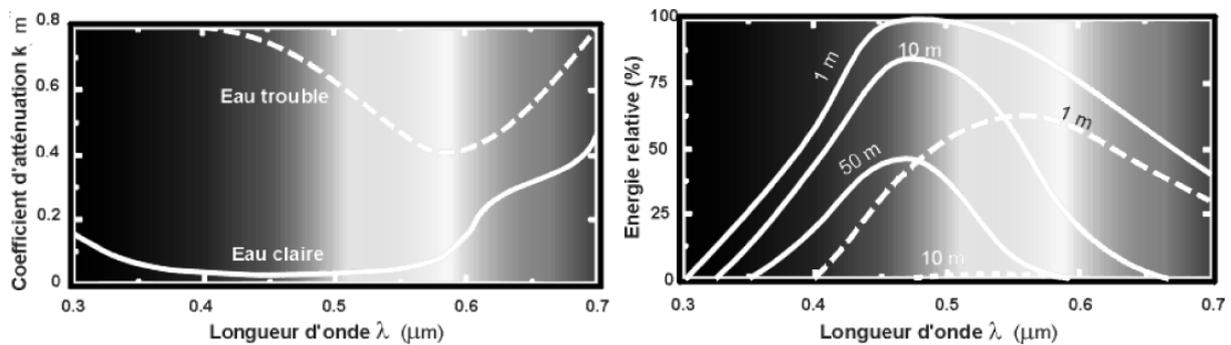
Le principal problème lié à l'exploration d'une zone par grands fonds est l'atténuation de la lumière. Ce phénomène est dû principalement à l'absorption de l'eau et peut être modélisé. Ainsi, pour un éclairage naturel de la mer par le soleil, l'intensité lumineuse à une longueur d'onde donnée λ décroît en fonction de la profondeur selon [Cal04] :

$$\mathbf{I}_\lambda(z) = \mathbf{I}_\lambda(0)e^{-k_\lambda z} \quad (5.1)$$

où

$$\begin{cases} I_\lambda(0) \text{ est l'intensité de la lumière à la surface} \\ k_\lambda \text{ est le coefficient d'atténuation verticale de l'eau (fonction de la longueur d'onde) exprimé en } m^{-1} \\ z \text{ est la profondeur de l'eau (en mètres)} \end{cases}$$

Le coefficient d'atténuation verticale de l'eau k_λ dépend de la longueur d'onde. Ainsi, la loi d'atténuation sera différente suivant que l'on considère le canal rouge (600nm), vert (550nm) ou bleu (450nm) (figure 5.1(a)). D'autre part, ce coefficient dépend également de la turbidité de l'eau (eau claire ou eau trouble). La lumière sera atténuée plus rapidement dans le cas d'une eau trouble.



(a) Evolution du coefficient d'atténuation verticale de l'eau k_λ en fonction de la longueur d'onde

(b) Absorption de l'eau en fonction de la longueur d'onde

FIG. 5.1 – Caractéristiques de l'absorption de la lumière dans l'eau.

Les courbes illustrées figure 5.1(b) montrent que l'absorption de l'eau dépend à la fois de la longueur d'onde considérée et de la turbidité de l'eau. Les images à traiter étant acquises à grande profondeur (jusqu'à -6000m), le milieu peut être considéré comme de l'eau pure du point de vue optique et en première approximation.

A partir de ce diagramme, deux remarques sont à faire :

- Certaines longueurs d'onde seront davantage affectées par la profondeur que d'autres en raison du spectre d'atténuation de l'eau (par exemple, le bleu sera très atténué par rapport aux autres couleurs, même à une distance fond-caméra de 1m). Ainsi, dans le cas d'images couleur, le traitement pour corriger la mauvaise répartition de l'intensité lumineuse ne pourra pas être le même pour tous les canaux.
- A cause de l'atténuation de la lumière dans l'eau (les courbes d'énergie se "tassent" avec la profondeur), le champ de visibilité est très réduit voire nul en grande profondeur. Au moins deux spots lumineux fixés à l'engin sont nécessaires mais ils sont la source d'un halo et d'une non-uniformité spatiale de l'intensité. L'image fig. 5.2 page ci-contre, représentant une épave d'avion, met ce phénomène en évidence.

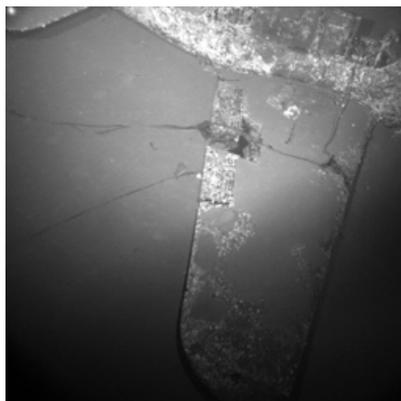


FIG. 5.2 – Image d'une épave d'avion, avec halo lumineux.

5.1.3 Un pré-traitement : la correction d'intensité lumineuse

Afin d'atténuer les effets lumineux qui peuvent perturber la construction de la mosaïque, deux méthodes ont été proposées dans [BRL03] : la correction radiométrique [Tar95] et le filtrage homomorphique [Ade98]. Des tests ont été réalisés sur des images réelles et seront présentés dans la partie résultats 5.8.1.

5.1.3.1 Correction radiométrique

Lors de la prise d'images par une caméra CCD, on fait souvent l'hypothèse que les cellules CCD ont toutes la même sensibilité et la même réponse radiométrique. Ce sera le cas après correction radiométrique du système d'acquisition. Le système aura à subir deux corrections, l'une correspondant à la variation de réponse radiométrique des sites CCD, l'autre correspondant aux phénomènes géométriques et lumineux lors de la prise de vue. Étant donné qu'il est très difficile de modéliser ces effets, on sera amené à construire des images de référence qui vont permettre de calibrer indépendamment chacun des pixels.

Défauts lors de la phase d'acquisition La réponse d'une cellule CCD est proportionnelle à l'éclairement reçu par sa surface (sur la plage linéaire de la caractéristique). Le gain correspondant dépend de l'épaisseur de la couche de silicium. Les écarts de sensibilité seront modélisés par un gain g_{ens} , propre à chaque pixel (x, y) .

Le courant d'obscurité est émis par la matrice CCD en l'absence d'excitation lumineuse. Il peut varier jusqu'à 10% entre pixel. Soit I_0 l'intensité de l'image ainsi créée.

De plus, il existe un bruit dû à la transmission entre la caméra et la carte d'acquisition, centré et indépendant d'un pixel à un autre. On le notera I_{transm} .

D'autre part, il existe deux phénomènes liés l'objectif et à la source lumineuse. Le premier est le phénomène dit de vignettage : l'éclairement du plan image obtenu à partir d'une surface de luminance uniforme varie en $\cos^4\Phi$, Φ étant l'angle entre l'axe optique et le rayon principal passant par le point considéré (voir figure 5.3(a)).

Ainsi, les bords de l'image seront toujours plus sombres que le centre (puisque lorsque l'angle augmente, le cosinus diminue). De plus, nous perdrons d'autant plus d'information que l'objectif de la caméra sera grand

angle.

Le deuxième phénomène est dû à l'éclairage de la zone par une source lumineuse. En effet, dans le cas d'une source ponctuelle (voir sur la figure 5.3(b)), la relation entre l'éclairement au point de visée sur la surface éclairée et en un point éloigné de la normale est proportionnelle à $\cos^3 \theta$ (θ étant l'angle de visée). Ainsi, un objet étendu de luminance uniforme se trouvant décentré par rapport à la source sera vu par la caméra comme de luminance non uniforme. Dans les expériences que nous avons conduites, le projecteur, placé à côté de la caméra, induit ce phénomène.

La non-uniformité spatiale peut aller jusqu'à 10% entre les bords et le centre de l'image. On modélisera cet effet par g_{obj} .

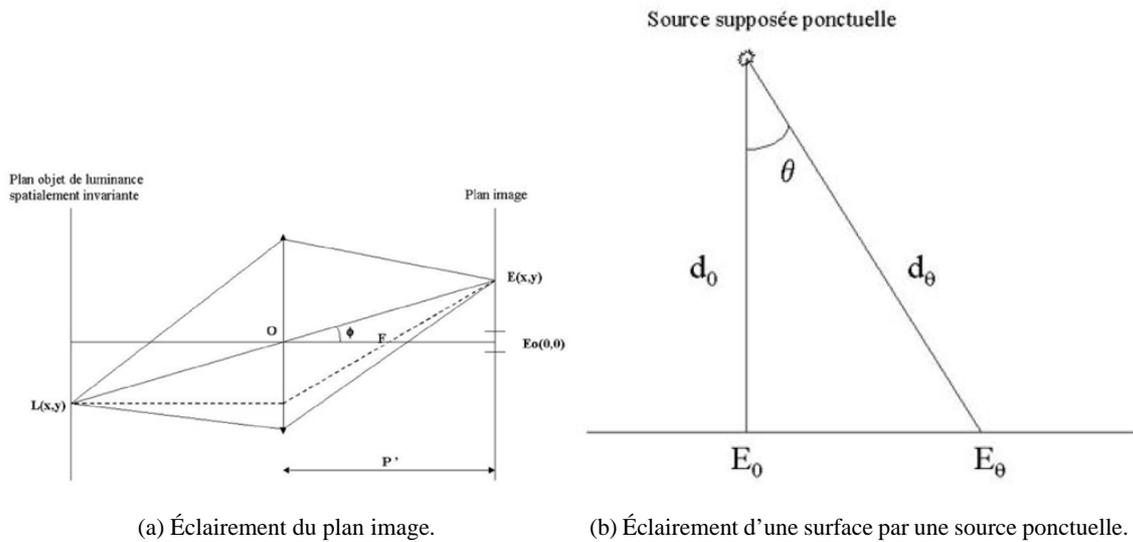


FIG. 5.3 – Phénomènes liés à l'objectif et à la source lumineuse.

Correction radiométrique du système d'acquisition

Modèle théorique Soit \mathbf{I}_a l'image acquise, \mathbf{I}_{exacte} l'image telle qu'elle aurait dû être en l'absence de perturbations. Un modèle théorique a été proposé par Tarel [Tar95], fondé sur la formule suivante :

$$\mathbf{I}_a(x, y) = g_{sens}(x, y)g_{obj}(x, y)\mathbf{I}_{exacte}(x, y) + \mathbf{I}_o(x, y) + \mathbf{I}_{transm}(x, y) \quad (5.2)$$

où \mathbf{I}_o est l'image de référence prise à objectif fermé. On note $g = g_{sens}g_{obj}$ l'image de référence correspondant à une image uniforme éclairée par le même éclairage que l'image acquise.

Modèle expérimental Le modèle théorique précédent est utilisé en imagerie satellitaire essentiellement. Dans notre article [BRL03], il a été adapté à la problématique sous-marine.

L'image due au bruit de transmission \mathbf{I}_{transm} peut être éliminée dans le temps par un moyennage, si l'on

considère que ce bruit est assimilable à un bruit gaussien centré. L'équation 5.2 devient alors :

$$\mathbf{I}_{exacte}(x, y) = \frac{\mathbf{I}_a(x, y) - \mathbf{I}_o(x, y)}{g(x, y)} \quad (5.3)$$

Ce qui revient à une soustraction suivie d'une division, à condition de disposer des deux images de référence \mathbf{I}_o et g .

L'image de référence \mathbf{I}_o est acquise en fermant l'objectif de la caméra. Celle-ci étant en milieu profond, sa température est de l'ordre de 5 à 10°C. Nous avons donc supposé que les valeurs des pixels étaient faibles, donc que \mathbf{I}_o était négligeable. Cette méthode revient donc à une simple division de l'image acquise par une image de référence (en l'occurrence g).

L'image de référence g correspondrait à un fond uniforme sans objet (par exemple un fond sablonneux) éclairé par deux (ou plus) spots lumineux. Malheureusement, opérationnellement, il n'est quasiment jamais possible d'obtenir une telle image. Nous avons donc proposé plusieurs méthodes pour obtenir cette image de référence.

La première est de convoluer un filtre gaussien de taille assez grande avec chacune des images. La seconde est de faire l'hypothèse que le halo s'apparente à une gaussienne basée sur une ellipse. Néanmoins, cette méthode requiert l'intervention d'un utilisateur pour chaque séquence : il faut modéliser l'ellipse (à la main ou automatiquement), et adapter les paramètres de la gaussienne à la séquence. Enfin, la méthode la plus satisfaisante et prometteuse consiste à rechercher une image de référence naturelle, soit en l'acquérant lors de la descente de l'engin, en pleine eau, soit en mettant la focale de la caméra à l'infini, une fois l'engin stabilisé près du fond marin.

Actuellement, nous n'avons pas à notre disposition de séquence d'images comportant une image prise en pleine eau. Nous n'avons donc pu tester la méthode que sur des photos, et n'avons pas pu reconstituer de mosaïque pour valider cette méthode. Les résultats sont présentés dans la section 5.8.1.

5.1.3.2 Filtrage homomorphique

Si l'on interprète une image grâce à un modèle illumination-réflectance, alors l'image représentée par la fonction $\mathbf{I}(x, y)$ peut être écrite comme le produit d'un terme d'illumination $i(x, y)$ et d'un terme de réflectance $r(x, y)$ [Ade98] :

$$\mathbf{I}(x, y) = i(x, y)r(x, y) \quad (5.4)$$

L'idée est de séparer ces deux composantes pour donner plus de poids à la réflectance et atténuer l'illumination -donc le halo-. En effet, dans une image, l'illumination varie lentement en général, alors que la réflectance peut varier de façon très abrupte, particulièrement à la limite entre deux objets. Effectuer un filtrage homomorphique consiste à séparer le terme d'illumination du terme de réflectance. Ainsi, les basses fréquences de l'image (illumination) pourront être modifiées grâce au comportement aux basses fréquences du filtre, tandis que les hautes fréquences du filtre agiront sur la réflectance, qui s'étendent plutôt vers les hautes fréquences. Le principe de ce filtrage est décrit ci-après, et résumé sur la figure 5.4.

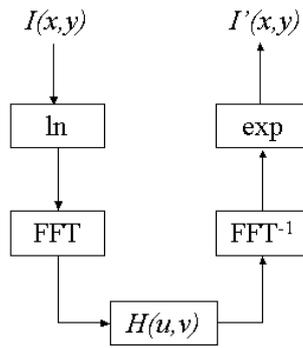


FIG. 5.4 – Principe du filtrage homomorphique.

La transformée de Fourier discrète \mathcal{F} n'ayant pas la propriété de séparabilité, c'est-à-dire que $\mathcal{F}\{\mathbf{I}\} \neq \mathcal{F}\{i\}\mathcal{F}\{r\}$, il a fallu contourner le problème de la façon suivante :

- Dans un premier temps, on applique le logarithme népérien à l'image :

$$\ln \mathbf{I}(x, y) = \ln i(x, y) + \ln r(x, y) \quad (5.5)$$

- Dans un deuxième temps, on applique la transformée de Fourier discrète à cette expression :

$$F_{\mathbf{I}}(u, v) = F_i(u, v) + F_r(u, v) \quad (5.6)$$

où

$$\begin{cases} F_{\mathbf{I}}(u, v) \text{ est la transformée de Fourier de } \ln \mathbf{I}(x, y) : F_{\mathbf{I}}(u, v) = \mathcal{F}\{\ln \mathbf{I}\}(u, v) \\ F_i(u, v) = \mathcal{F}\{\ln i\}(u, v) \\ F_r(u, v) = \mathcal{F}\{\ln r\}(u, v) \end{cases}$$

- On construit ensuite un filtre de fonction de transfert H passe-haut agissant sur F_i et F_r :

$$F_{\mathbf{I}}(u, v)H(u, v) = F_i(u, v)H(u, v) + F_r(u, v)H(u, v) \quad (5.7)$$

Le terme d'illumination se trouve approximativement au centre de l'image F_i et le terme de réflectance se trouve sur les bords (hautes fréquences). Bien que ces termes ne soient pas strictement séparés, l'approche homomorphique fonctionne tout de même. Ainsi, le filtre peut agir sur les basses fréquences associées à l'illumination, indépendamment de son action sur les hautes fréquences associées à la réflectance.

- Il suffit ensuite de revenir dans le domaine spatial grâce à la transformée inverse de Fourier, puis de prendre l'exponentielle du résultat obtenu. Le résultat final est l'image corrigée \mathbf{I} .

5.2 Techniques nécessaires à l'estimation de la transformation entre deux images

Dans cette section, nous allons présenter les étapes nécessaires à l'estimation de la transformation liant deux images, sans connaissance a priori sur cette transformation. Elles consistent en l'extraction puis la sélection de primitives (ou caractéristiques) pour chaque image, puis à leur mise en correspondance.

Dans ce qui suit, on note \mathbf{I}_1 la première image et \mathbf{I}_2 la seconde image. P_1 (respectivement P_2) est un point considéré dans l'image \mathbf{I}_1 (resp. \mathbf{I}_2).

5.2.1 Détection de caractéristiques

Un très grand nombre de travaux ont été réalisés à ce sujet [GW92, MH80, Can86, DG93]... La présentation qui suit n'est pas exhaustive mais resitue les principales méthodes évoquées dans le contexte de notre travail de recherche.

Les caractéristiques détectées dans une image (ou encore *primitives*) peuvent être de plusieurs types, notamment des frontières ou contours, des points d'intérêt ou des régions.

5.2.1.1 Détection de contours

Un contour correspond à la limite entre deux régions distinctes. C'est un ensemble de points marquant une forte variation d'intensité des pixels. Le contour se caractérise par son amplitude et son orientation. Il peut être détecté grâce à un fort pic de gradient ou un passage par zéro du laplacien.

On trouve dans la littérature un grand nombre de publications sur les détecteurs de contours. Parmi eux, les détecteurs les plus simples sont les gradients directionnels. Ils ont été proposés par Sobel et Prewitt, évoqués par exemple dans [GW92], qui ont défini les masques directionnels en x , M_x et en y , M_y , comme :

$$M_x = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix} \quad \text{et} \quad M_y = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix}$$

où $c = 1$ pour les masques de Prewitt, et $c = 2$ pour les masques de Sobel. Les gradients selon x et y de l'image considérée, notés G_x et G_y , sont obtenus par convolution de l'image \mathbf{I} avec respectivement M_x et M_y (le produit de convolution est noté $*$) :

$$G_x = M_x * \mathbf{I} \quad \text{et} \quad G_y = M_y * \mathbf{I}$$

L'image de contours, notée NG (pour Norme du Gradient), est obtenue en repérant les maxima dans l'image représentant la norme du gradient :

$$NG = \sqrt{\|G_x\|^2 + \|G_y\|^2} \quad (5.8)$$

Les contours peuvent être également représentés par les passages par zéro du laplacien, ce qui équivaut aux maxima des gradients. Les masques L_1 et L_2 ci-dessous permettent par exemple de calculer par convolution

le laplacien d'une image (il existe d'autres masques permettant de calculer le laplacien) :

$$L_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad L_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Le laplacien d'une image \mathbf{I} est donné par :

$$LAPL = L * \mathbf{I} \quad (5.9)$$

Il suffit ensuite de détecter les passages par zéros du résultat pour déduire les contours.

Les deux méthodes précédentes sont très sensibles au bruit dans l'image. Aussi, en 1980, Marr et Hildreth [MH80] ont-ils proposé un détecteur de contours qui consiste à convoluer avec un filtre Laplacien l'image lissée par un filtre gaussien. Cela revient en pratique à convoluer l'image par un filtre Laplacien de Gaussien (LoG).

Il existe d'autres filtres de détection de contours par optimisation. Les plus classiques sont les filtres de Canny [Can86] et Deriche [DG93]. L'algorithme de Canny [Can86] consiste à rechercher la fonction qui maximise un critère (non exposé ici), sous trois contraintes : fiabilité de la détection, précision de la localisation et réponse unique pour un contour. La solution recherchée par Canny est caractérisée par un filtrage à réponse impulsionnelle finie. La solution générale est une somme de produits d'exponentielles et de sinus et cosinus. Canny estime que le choix d'un filtre dont la réponse impulsionnelle serait la dérivée d'une gaussienne ne détériore les performances du filtre optimal trouvé que de 20 %. Ce filtre étant facilement implantable en 2 dimensions (c'est d'ailleurs le seul que Canny a implémenté) est celui qui est utilisé en pratique.

Deriche [DG93], partant de la maximisation du même critère et des mêmes contraintes d'optimalité que Canny, a proposé une solution à réponse impulsionnelle infinie, qui met en jeu une exponentielle négative et un sinus.

Les extracteurs de contours sont utilisés lorsque des structures sont significatives. Comme exemple de scènes "structurées", on peut citer dans l'industrie, les pièces usinées (extraction de contours pour vérifier les dimensions d'une pièce), ou des bâtiments, pour l'architecture. En revanche, ces types de détecteurs ne donnent pas de très bons résultats avec les images sous-marines qui sont souvent très bruitées et peu structurées.

5.2.1.2 Extraction de points dans une image

Après avoir présenté un bref état de l'art sur les détecteurs de contours, quelques extracteurs de points classiques et les mieux adaptés à notre problématique vont être exposés.

Dans l'article [Sch98], C. Schmid compare plusieurs détecteurs de points. Deux critères sont donnés pour évaluer les détecteurs : la répétabilité (même nombre de points détectés lors d'un changement d'illumination

de la scène) et le contenu d'information (les points détectés doivent se distinguer les uns des autres). Les détecteurs de points sont classés en :

- méthodes basées sur une extraction de contours suivie d'un choix de points d'intérêt issus des contours ([AB86] par exemple),
- méthodes reposant sur l'estimation des paramètres d'un modèle de coins ([DG93] par exemple),
- méthodes orientées signal : calcul direct des points d'intérêt.

Nous allons ici exposer brièvement quelques-unes des méthodes orientées signal, car elles nous seront utiles pour la suite (partie 5.3.1).

Beaudet [Bea78] propose un opérateur de détection *DET*, invariant à la rotation et utilisant les dérivées secondes de l'image, notées \mathbf{I}_{xx} , \mathbf{I}_{xy} et \mathbf{I}_{yy} . Soit H le hessien de l'image \mathbf{I} au pixel de coordonnées (x, y) :

$$H = \begin{bmatrix} \mathbf{I}_{xx} & \mathbf{I}_{xy} \\ \mathbf{I}_{xy} & \mathbf{I}_{yy} \end{bmatrix}$$

alors le détecteur *DET* est le déterminant de H .

Kitchen et Rosenfeld [KR82] commencent par approximer localement la surface d'intensité de l'image par une surface polynômiale afin de réduire le bruit. Puis ils proposent un opérateur basé sur la variation de l'orientation du gradient, noté K .

$$K = \frac{u_{xx}u_y^2 + u_{yy}u_x^2 - 2u_{xy}u_xu_y}{u_x^2 + u_y^2}$$

Harris et Stephen [HS88] s'appuient sur les dérivées premières de l'image lissée par une gaussienne \mathbf{I}_x et $\mathbf{I}_{\sigma y}$. Soit la matrice 2×2

$$M(x, y) = \begin{bmatrix} \mathbf{I}_{\sigma x}(x, y) & \mathbf{I}_{\sigma x}(x, y)\mathbf{I}_{\sigma y}(x, y) \\ \mathbf{I}_{\sigma x}(x, y)\mathbf{I}_{\sigma y}(x, y) & \mathbf{I}_{\sigma y}(x, y) \end{bmatrix}$$

Les valeurs propres de la matrice $M(x, y)$ sont importantes :

- si les deux valeurs propres sont faibles par rapport à un seuil fixé selon l'image, la région considérée a une intensité constante,
- si les deux valeurs propres sont relativement différentes, la région considérée présente une arête,
- si les deux valeurs propres sont élevées par rapport au seuil, la région présente un point d'intérêt.

Pour éviter d'avoir à évaluer les valeurs propres de M , les auteurs ont introduit le détecteur $HS(x, y)$ suivant :

$$HS(x, y) = \det(M(x, y)) - kTrace^2(M)$$

La valeur de k est fixée à 0.04 par les auteurs. Si $HS(x, y)$ est supérieur à 0, le pixel (x, y) est détecté comme un point d'intérêt.

Le détecteur CSS proposé par Mokhtarian [MS98] repose sur une détection de contours de type Canny à partir desquels des points sont extraits.

Smith et Brady [SB97] ont proposé une approche très simple (détecteur SUSAN), à partir de masques de forme circulaire dont les centres sont appelés "nucleus". Ils proposent de balayer chaque point de l'image avec le masque circulaire et comptent le nombre N de pixels inclus dans le masque ayant une valeur proche de celle du pixel central. Après un seuillage sur N , on détermine si oui ou non, un point d'intérêt est détecté.

Les résultats de la comparaison des détecteurs de points effectuée par Schmid concluent sur les meilleures performances, du point de vue des critères sélectionnés, du détecteur de Harris.

5.2.2 Caractérisation des points extraits

Le problème de recherche des invariants dans une image consiste à rechercher des quantités caractéristiques de l'image, indépendantes du point de vue, des conditions d'illumination, etc... Les invariants peuvent être utilisés dans la mise en correspondance d'images ou bien dans l'indexation d'images, c'est-à-dire la description de l'image par un "résumé" (pouvant être un certain nombre d'invariants) pour faciliter la recherche d'images dans une base de données à partir d'une description synthétique. Dans [GRC01], Gros fait un état de l'art sur l'utilisation des invariants en vision et en robotique.

Dans ce cadre, l'ensemble des points extraits dans l'image peut permettre de représenter cette image par une liste de primitives. Pour mettre en correspondance deux images, il suffit de mettre en correspondance deux ensembles de caractéristiques décrivant les points extraits et leur environnement local. La caractérisation choisie doit être indépendante des paramètres de la transformation calculée entre les images.

Plusieurs types d'invariants sont possibles. Parmi eux, les invariants *géométriques* utilisent la géométrie d'un groupe de points. Un exemple simple est la distance euclidienne entre deux points, qui est invariante à la translation et à la rotation. L'angle entre 3 points et le birapport entre 4 points font également partie des invariants géométriques. Cependant, les méthodes employant les invariants géométriques sont sensibles aux mauvais points extraits et le voisinage des points détectés n'est pas pris en compte.

Pour compléter ces méthodes, une alternative est l'utilisation d'invariants *différentiels*. Ceux-ci prennent en compte la valeur de l'intensité du point et de ses dérivées. Par exemple, Sistiaga [Sis00] définit le vecteur V_{AD} (AD pour Attributs Différentiels) suivant :

$$V_{AD} = [\mathbf{I} \quad NG \quad LAPL \quad DET \quad K]^T \quad (5.10)$$

où \mathbf{I} est le niveau de gris du pixel extrait, NG la norme du gradient en ce point (voir équation 5.8), $LAPL$ le laplacien (voir équation 5.9), DET l'opérateur de Beaudet et K l'opérateur de Kitchen-Rosenfeld. L'avan-

tage de ces invariants est la tolérance aux changements de résolution et la moindre sensibilité au bruit.

5.2.3 Mise en correspondance de points

La mise en correspondance d'images consiste à appairer les primitives (dans ce qui est exposé ici, les primitives sont les points extraits) issues d'une image avec celles de l'image suivante, puis à les utiliser pour estimer la transformation entre les deux images.

5.2.3.1 Quelques critères mesurant la ressemblance entre deux primitives

Plusieurs critères utiles à la mesure de ressemblance entre deux primitives sont à notre disposition. Le critère le plus simple est un critère de corrélation sur un fenêtrage¹ autour d'un point. La SSD (Sum of Squared Differences) représente la somme des différences quadratiques entre termes correspondant aux deux fenêtres [Bro92].

D'autres détecteurs en ont été inspirés : la SAD (Sum of Absolute Differences) est la somme des valeurs absolues des différences entre termes correspondants, la ZSSD (Zero mean SSD) est identique à la SSD, mis à part que la moyenne des termes de la fenêtre est soustraite à chaque terme, la NCC (Normalized Cross Correlation), le critère le plus utilisé, est la somme des produits entre termes correspondants, normalisée par le produit des moyennes quadratiques calculées pour chaque fenêtre, la ZNCC (Zero Mean NCC) est une corrélation croisée entre écarts à la moyenne sur un fenêtrage $[-n, +n]$ autour du pixel central. C'est donc un critère invariant aux variations uniformes d'intensité lumineuse dans une des deux images. Voilà son expression pour deux points $P_1 = (x_1, y_1)^T$ et $P_2 = (x_2, y_2)^T$, appartenant respectivement aux images \mathbf{I}_1 et \mathbf{I}_2 :

$$ZNCC(P_1, P_2) = \frac{\sum_{i=-n}^n \sum_{j=-n}^n [\mathbf{I}_1(x_1 + i, y_1 + j) - \bar{\mathbf{I}}_1(x_1, y_1)] [\mathbf{I}_2(x_2 + i, y_2 + j) - \bar{\mathbf{I}}_2(x_2, y_2)]}{\sqrt{\sum_{i=-n}^n \sum_{j=-n}^n [\mathbf{I}_1(x_1 + i, y_1 + j) - \bar{\mathbf{I}}_1(x_1, y_1)]^2 \sum_{i=-n}^n \sum_{j=-n}^n [\mathbf{I}_2(x_2 + i, y_2 + j) - \bar{\mathbf{I}}_2(x_2, y_2)]^2}} \quad (5.11)$$

Plus le critère ZNCC est proche de 1, plus les points P_1 et P_2 seront semblables par rapport à la distribution des intensités de niveaux de gris sur le fenêtrage considéré.

5.2.3.2 Méthodes de mise en correspondance d'images

Pour la mise en correspondance d'images successives proprement dite, sans connaissance a priori sur les paramètres de la caméra, Brown [Bro92] et Zhang [Zha93] ont répertorié les méthodes principalement utilisées.

La méthode la plus simple utilise les critères de corrélation exposés ci-dessus. Elle repose sur l'hypothèse que le déplacement inter-image est relativement réduit (hypothèse faite lors de l'acquisition d'une séquence d'images). Nous pouvons alors considérer que si un point d'intérêt était dans l'image \mathbf{I} à la position (x, y) ,

¹Un fenêtrage est une zone d'intérêt autour d'un point de l'image.

alors ce point dans l'image I_{i+1} se trouvera dans un voisinage de (x, y) . Cette méthode consiste donc à faire la corrélation (ou ZNCC ou autre...) d'un voisinage du point d'intérêt avec un voisinage sur l'image suivante. Le pic de la corrélation donnera la localisation du point correspondant dans la deuxième image.

Une autre méthode possible est la relaxation : cette méthode consiste à relier initialement chacun des points de la première image avec chacun des points de la deuxième image. Chaque lien entre deux points d'intérêt a un coût, un poids (ou une probabilité) qui tient compte de certaines contraintes, par exemple les contraintes de voisinage, la valeur du gradient, l'angle du coin. Cette méthode est itérative. Les points à appairer sont alors les points correspondant aux liens de poids le plus fort.

On peut également trouver des méthodes qui mettent en jeu la théorie des graphes. Étant données N primitives dans l'image $I_1 : P_{11} \dots P_{1N}$ et n primitives dans l'image $I_2 : P_{21} \dots P_{2n}$. On introduit une primitive spéciale, la primitive nulle, dans chaque image : une liaison entre la primitive d'une image et la primitive nulle signifie que la primitive de l'image ne peut être appariée avec aucune autre primitive dans l'image à mettre en correspondance.

Un arbre d'interprétation est alors construit (voir figure 5.5). Au 1^{er} niveau de l'arbre, chaque noeud représente une association possible entre une primitive plausible de l'image 2 avec la première primitive P_{11} de l'image I_1 (ordonnée par une méthode quelconque). Toutes les primitives de l'image, y compris la primitive nulle, sont des assignations plausibles à P_{11} . Donc au premier niveau de l'arbre, nous avons $n + 1$ associations possibles. Au deuxième niveau, nous considérons toutes les associations possibles, connaissant les associations du premier niveau. Nous avons donc $n + 1$ noeuds sous chaque noeud du premier niveau. L'arbre se construit ainsi, jusqu'à la primitive N de l'image I_1 , P_{1N} . On peut remarquer qu'il y a $(n + 1)^N$ noeuds au niveau N .

A partir de l'arborescence ainsi construite, on réduit les interprétations possibles en ne conservant que les associations de points dites "consistantes" [Zha93], c'est-à-dire respectant l'unicité de l'appariement et l'ordre de gauche-à-droite par exemple.

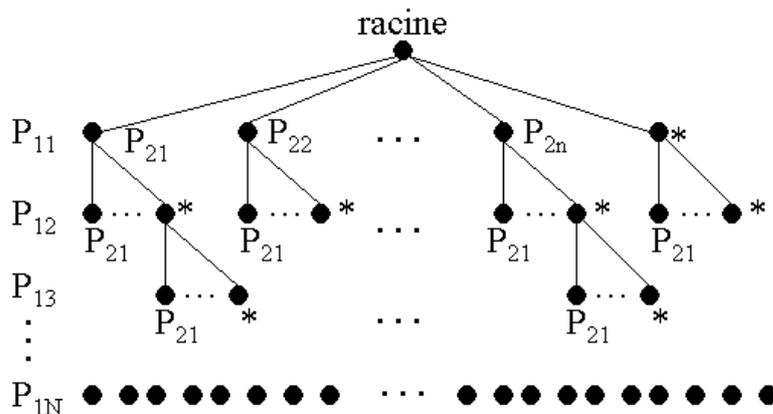


FIG. 5.5 – Appariement graphes.

5.3 Recherche des "instants de rendez-vous" : utilisation de l'algorithme PISA [Sis00]

Une fois obtenue une liste de points appariés par une des méthodes exposées ou autre, les paramètres de la transformation entre les deux images peuvent être estimés. Dans toute la suite, on suppose qu'elle est composée d'une rotation suivie d'un changement de résolution, suivi d'une translation. Pour cela, nous allons tout d'abord poser quelques notations. Il s'agit d'estimer le vecteur de translation $T = (t_x, t_y)^T$, l'angle de rotation α et le rapport de résolution ρ entre les deux images, à partir d'un ensemble de points appariés. Soit P_1 (resp. P_2) la position d'un point extrait de l'image \mathbf{I}_1 (resp. \mathbf{I}_2). Alors :

$$P_2 = \rho R_\alpha P_1 + T + \epsilon \quad (5.12)$$

où R_α est la matrice de rotation d'angle α et ϵ un bruit sur la position. Dans la suite, on désigne P'_1 comme le point correspondant à P_1 (de l'image 1) reprojecté dans l'image 2 grâce la transformation estimée $(\tilde{\rho}, \tilde{\alpha}, \tilde{T})$:

$$P'_1 = \tilde{\rho} R_{\tilde{\alpha}} P_1 + \tilde{T} \quad (5.13)$$

Afin d'estimer la transformation, plusieurs méthodes peuvent être mises en oeuvre à partir de la liste d'appariements, comme par exemple la minimisation des moindres carrés pondérés ou le filtrage de Kalman.

Nous allons maintenant décrire un algorithme complet de mise en correspondance d'images, PISA [Sis00], développé par M. Sistiaga à l'Ifremer dans le cadre d'une thèse. Cette méthode, baptisée PISA pour Procédure Itérative et Statistique d'Appariement, permet de mettre en correspondance deux images de résolution pas forcément identiques, à partir d'une liste de points d'intérêt détectés dans chaque image lors d'une phase préliminaire. Nous proposerons ensuite des extensions à cette méthode puis montrerons son utilité pour la recherche des instants de rendez-vous.

5.3.1 L'algorithme PISA [Sis00]

La PISA consiste à répéter jusqu'à convergence les phases d'appariement de points d'intérêt, d'estimation de la transformation et de réactualisation de paramètres. Le principe en est illustré figure 5.6 page suivante.

Dans ce qui suit, nous décrivons le fonctionnement de chaque élément du diagramme, en utilisant les briques exposées dans l'état de l'art des sections précédentes.

Détection de points d'intérêt Pour réaliser cette étape préalable, Sistiaga propose trois types de détecteurs de points d'intérêt : les passages par zéro du Laplacien, les maxima du gradient et le détecteur de Harris [HS88]. Sistiaga ajoute une étape de seuillage par hystérésis du gradient des points détectés. On pose S et S_h les seuils respectivement bas et haut. Trois types de seuillage sont intégrés dans son logiciel et paramétrés par l'opérateur :

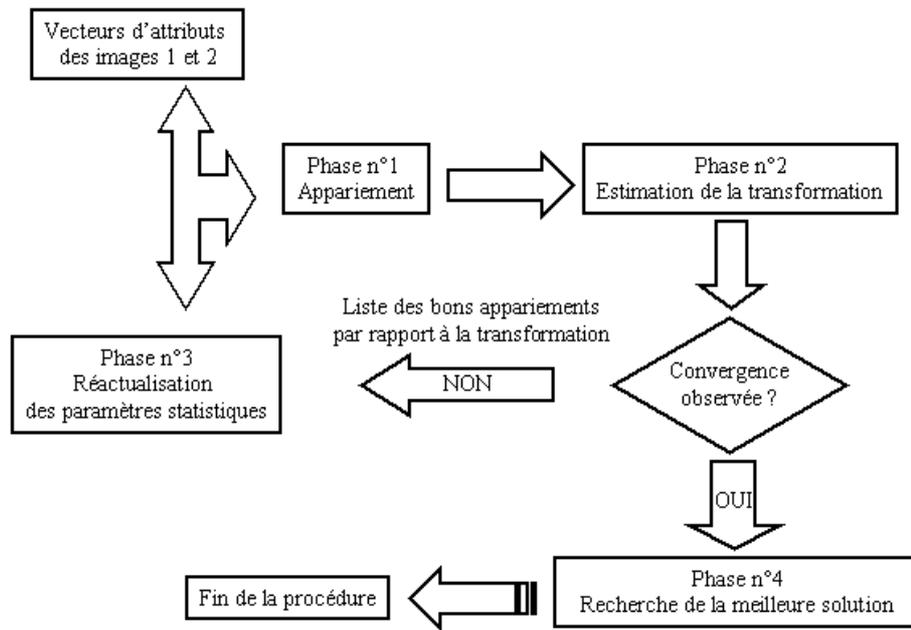


FIG. 5.6 – Principe de la PISA.

- Le pourcentage du maximum des gradients : S_b et S_h correspondent à des pourcentages de la valeur maximale des normes du gradient des points détectés.
- Le pourcentage de points éliminés : la liste des points caractéristiques est triée par ordre croissant des normes du gradient. S_b et S_h sont les pourcentages de ce tri qui permettent d'éliminer le pourcentage de points désiré.
- Nombre de points conservés : il s'agit de déterminer les seuils S_b et S_h qui permettent de conserver un certain nombre de points caractéristiques.

Pour chacun des points détectés, cinq caractéristiques sont calculées et regroupées dans un vecteur d'attribut V_{AD} . Il s'agit de l'intensité de niveau de gris du pixel considéré I , de la norme du gradient NG , du laplacien $LAPL$, de la valeur de l'opérateur de Beudet DET , de la valeur de l'opérateur K de Kitchen-Rosenfeld.

$$V_{AD} = [I \quad NG \quad LAPL \quad DET \quad K]^T \quad (5.14)$$

Une fois les points détectés et leurs attributs calculés pour chaque image, tous les appariements possibles entre les deux images considérées sont évalués dans la phase suivante.

Phase d'appariement Un appariement est détecté si la distance de Mahalanobis entre les deux vecteurs d'attributs des points considérés est inférieure à un seuil. Ce seuil est fixé par l'utilisateur. La distance de Mahalanobis entre les vecteurs V_1 (vecteur d'attribut du point P_1 l'image I_1) et V_2 (vecteur d'attribut du point P_2 de l'image I_2) est donnée par :

$$\delta_a^2 = (V_1 - V_2) \cdot \Lambda_a^{-1} \cdot (V_1 - V_2)^T \quad (5.15)$$

où Λ_a est la matrice de covariance des attributs. On considère que les attributs sont indépendants les uns des autres, ainsi :

$$\Lambda_a = \begin{bmatrix} \sigma_{\mathbf{I}}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{NG}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{LAPL}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{DET}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_K^2 \end{bmatrix}$$

L'initialisation de la matrice Λ est réalisée de manière empirique, sur une méthode reposant sur l'utilisation de l'histogramme de $\mathbf{I}_1 - \mathbf{I}_2$. Pour plus de détails, le lecteur peut se reporter à [Sis00].

Si l'on considère que V_1 et V_2 sont des vecteurs aléatoires suivant une loi gaussienne, alors \mathcal{Q}_a^2 suit une loi du χ^2 . Pour sélectionner les appariements, Sistiaga a donc choisi une méthode de seuillage du χ^2 , qui revient, à partir des degrés de liberté du vecteur V_1 , à rechercher la probabilité d'appartenance à la distribution. En pratique, le seuil est fixé à 6.1, qui correspond à une probabilité d'appartenance à la distribution de 30% pour un vecteur d'attributs de 5 éléments. Les appariements ayant une distance inférieure au seuil sont conservés pour la suite. Remarquons que les appariements multiples sont possibles.

Phase d'estimation de la transformation A partir des appariements retenus pendant la phase de mise en correspondance, l'estimation de la transformation peut être effectuée. Avant d'être intégrés dans l'estimateur, les appariements obtenus lors de la phase précédente subissent un filtrage : la distance de Mahalanobis sur les vecteurs de position λ_p^2 est déterminée, puis, de la même façon que pour la distance de Mahalanobis sur les attributs, un seuillage du χ^2 est réalisé (le seuil est fixé à 2.41, ce qui correspond à une probabilité d'appartenance à la distribution de 30% pour un vecteur de position de taille 2).

$$\lambda_p^2 = (P_1 - P'_1)\Lambda_p^{-1}(P_1 - P'_1)^T \quad (5.16)$$

avec

$$\Lambda_p = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

Une fois les appariements aberrants éliminés, on obtient une liste de n couples validés, sur laquelle l'estimation va être réalisée.

Il s'agit alors d'estimer les paramètres ρ , α et T définis pour l'équation 5.12. Pour son logiciel, Sistiaga propose trois méthodes : la méthode de minimisation des moindres carrés, la méthode des moindres carrés pondérés et le filtrage de Kalman étendu. Notons que le modèle choisi étant non-linéaire, la minimisation des moindres carrés est réalisée par l'algorithme de Gauss-Newton.

Tout d'abord, pour tous les estimateurs proposés, une initialisation des paramètres ρ , α et T est réalisée. Pour plus de détails, le lecteur peut se reporter à la thèse de Sistiaga [Sis00].

Soit $\{P_{1k}\}_{k=1..n} = \{P_{11}...P_{1n}\}$ la liste des points de l'image \mathbf{I}_1 et $\{P_{2k}\}_{k=1..n} = \{P_{21}...P_{2n}\}$ la liste des points correspondants de l'image \mathbf{I}_2 et $\{P'_{1k}\}_{k=1..n} = \{P'_{11}...P'_{1n}\}$ la liste des points $\{P_{1k}\}$ reprojétés

par la transformation estimée.

Le critère des moindres carrés à minimiser est le suivant :

$$Q(T, \rho, \alpha) = \sum_k [P_{2k} - (\rho R_\alpha P_{1k} + T)]^2 = \sum_k [P_{2k} - P'_{1k}]^2 \quad (5.17)$$

En posant $\theta = (t_x, t_y, \rho, \alpha)^T$, $P_{mes} = [P_{21} \dots P_{2n}]^T$ et $P'(\theta) = [P'_{11} \dots P'_{1n}]^T$, la minimisation du critère est réalisée en itérant l'estimée $\hat{\theta}_k$ suivante :

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \left(J_{\hat{\theta}_{k-1}}^T J_{\hat{\theta}_{k-1}} \right)^{-1} J_{\hat{\theta}_{k-1}}^T [P_{mes} - P'(\hat{\theta}_{k-1})] \quad (5.18)$$

où $J_{\hat{\theta}_{k-1}}$ est la matrice jacobienne de P' calculée en $\hat{\theta}_{k-1}$. $\hat{\theta}_0$ est l'initialisation de la transformation et l'algorithme s'arrête lorsque $\|\hat{\theta}_k - \hat{\theta}_{k-1}\|$ est inférieur à un seuil choisi ou lorsque le nombre d'itérations maximal fixé par l'opérateur est atteint.

L'estimateur des moindres carrés pondérés est le suivant :

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \left(J_{\hat{\theta}_{k-1}}^T J_{\hat{\theta}_{k-1}} \right)^{-1} J_{\hat{\theta}_{k-1}}^T W [P_{mes} - P'(\hat{\theta}_{k-1})] \quad (5.19)$$

avec la matrice de confiance $W = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \lambda_n & 0 \\ 0 & 0 & 0 & 0 & \lambda_n \end{bmatrix}$ déduite des termes $\{\lambda_k\}_{k=1..n}$ qui repré-

sentent la confiance accordée au $k^{\text{ème}}$ appariement, calculés à partir des distances de Mahalanobis et des seuils appliqués (voir annexe B partie B.1).

Le principe du filtrage de Kalman n'est pas réintroduit ici (voir chapitre 3). Le filtrage est basé ici sur l'équation 5.12 à laquelle est ajouté un bruit de mesure, pour les n couples de points. Remarquons que l'estimation à l'itération k est optimisée pour les observations $\{1..k\}$ et non pour l'ensemble des données, contrairement à l'estimateur des moindres carrés. Cet estimateur peut être réitéré plusieurs fois pour la même liste d'appariements afin de réactualiser la matrice jacobienne J_{θ_0} calculée pour une nouvelle valeur initiale. Les termes λ_k peuvent être également utilisés pour pondérer les observations.

Dans le programme de Sistiaga, l'utilisateur choisit le type d'estimateur et fixe le nombre d'itérations de cet estimateur.

Remarque : le nombre d'itérations de l'estimateur employé et le nombre d'itérations de la procédure PISA sont deux données bien différentes. Le nombre d'itérations de l'estimateur est fixé par l'utilisateur, tandis que le nombre d'itérations de la PISA est évalué une fois la procédure terminée (c'est-à-dire une fois que les résultats ont convergé).

Convergence de l'estimation Après avoir estimé les paramètres de transformation, la convergence de cette transformation est recherchée sur les trois dernières itérations de la PISA. Si le nombre de couples de points après l'estimateur est faible ou si les variances convergent (écart quadratique inférieur à 10^{-6}) ou si les trois dernières transformations sont proches au sens géométrique, alors la convergence est observée. Dans ce cas, la meilleure solution parmi les trois dernières est déterminée à partir des critères de confiance sur la qualité de l'estimation (voir annexe B partie B.2). Dans le cas contraire, une nouvelle itération de la PISA est effectuée, en commençant par l'actualisation des paramètres statistiques (matrices de variance-covariance Λ_a et Λ_p) à partir de la liste de n couples validés.

5.3.2 Extension proposée à la PISA : les moindres carrés linéaires (MCL)

Sistiaga [Sis00] a développé la méthode des moindres carrés pour prendre en compte les cas d'estimation de 2, 3 ou 4 paramètres (translation seule, translation + rotation, translation + rotation + zoom). Cependant, nous avons développé un algorithme plus simple dans le cas où les 4 paramètres sont inconnus, c'est-à-dire que l'on souhaite estimer le rapport de résolution ρ . Dans ce cas, le modèle peut se linéariser et une forme analytique de l'estimateur est obtenue.

Nous avons fait commencé nos travaux en faisant l'hypothèse que la distance de la caméra au fond marin était constante. Néanmoins, ce n'est pas le cas en pratique, c'est pourquoi l'estimation du paramètre ρ n'est pas incohérente.

Soit P_{1k} le point de l'image 1, de coordonnées $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}_k$ et P_{2k} le point mis en correspondance dans l'image 2, de coordonnées $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}_k$. On a alors d'après l'équation 5.12 :

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}_k = \rho \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}_k + T + \epsilon_k \quad (5.20)$$

Soit

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}_k = \begin{pmatrix} \rho \cos \alpha x_1 - \rho \sin \alpha y_1 + t_x \\ \rho \sin \alpha x_1 + \rho \cos \alpha y_1 + t_y \end{pmatrix}_k + \epsilon_k \quad (5.21)$$

On pose :

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{pmatrix} \triangleq \begin{pmatrix} \rho \cos \alpha \\ \rho \sin \alpha \\ t_x \\ t_y \end{pmatrix} \quad (5.22)$$

De l'estimation de θ , on déduira donc les paramètres ρ, α, t_x et t_y de la façon suivante :

$$\begin{pmatrix} \rho \\ \alpha \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \sqrt{\theta_1^2 + \theta_2^2} \\ \arctan \frac{\theta_2}{\theta_1} \\ \theta_3 \\ \theta_4 \end{pmatrix} \quad (5.23)$$

On obtient alors :

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}_k = \begin{pmatrix} \theta_1 x_1 - \theta_2 y_1 + \theta_3 \\ \theta_2 x_1 + \theta_1 y_1 + \theta_4 \end{pmatrix}_k + \epsilon_k \quad (5.24)$$

En posant $A_k = \begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \end{pmatrix}_k$, on a :

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}_k = A_k \theta + \epsilon_k \quad (5.25)$$

Le nouveau modèle 5.25 étant linéaire, l'estimation des moindres carrés linéaires (**MCL**) s'obtient classiquement par la formule suivante :

$$\hat{\theta} = (A^T A)^{-1} A^T X \quad (5.26)$$

avec $A = \begin{bmatrix} A_1 \\ \dots \\ A_p \end{bmatrix}$ et $X = \begin{bmatrix} X_1 \\ \dots \\ X_p \end{bmatrix}$ avec $X_k = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}_k$. L'estimateur proposé est sans biais et sa variance est calculable, sous réserve que la variance des ϵ_k soit connue. Contrairement à la méthode proposée par Sistiaga, le résultat est obtenu directement et l'algorithme est plus simple.

De la même façon, on peut déterminer l'estimé des moindres carrés pondérés linéaires (**MCPL**) selon :

$$\hat{\theta} = (A^T W A)^{-1} A^T W X \quad (5.27)$$

où W est la matrice de confiance introduite par Sistiaga.

Ces deux méthodes **directes** reposant sur un **modèle linéaire** permettent d'éviter les initialisations des paramètres de la transformation, les itérations des estimateurs proposés par Sistiaga et évitent les écueils numériques puisque le problème des minima locaux n'existe pas.

5.3.3 Utilisation de la PISA à la recherche des instants de rendez-vous

A partir de l'algorithme PISA, nous avons mis au point une méthode permettant de rechercher un couple d'images les plus proches dans un fenêtrage.

Une première estimation "grossière" des instants de rendez-vous est donnée par la localisation $A/(C,V)$ (intégration simple des caps et vitesses bruités). Par exemple, si on considère une trajectoire à une boucle, on note les deux instants correspondants \hat{I} et \hat{J} .

Pour réaliser une estimation plus fine, on recherche alors, à partir de cette première estimation grossière, dans un intervalle $[n_I, n_J]$ autour de \hat{I} et \hat{J} les couples d'images $(\mathbf{I}_k, \mathbf{I}_l)_{k \in [\hat{I}-n_I, \hat{I}+n_I], l \in [\hat{J}-n_J, \hat{J}+n_J]}$ qui présentent le plus de ressemblance. Le critère choisi est une translation la plus faible possible entre 2 images dans le fenêtrage : $\|T\| = \sqrt{t_x^2 + t_y^2}$.

D'autres critères seraient possibles mais nous avons retenu celui-ci car il paraît le plus adapté à notre problème. En effet, le zoom importe peu. D'autre part, il n'est pas important que l'image qui correspond au retour de l'engin au-dessus du point de rendez-vous ait subi une rotation (ou pas) par rapport à l'image de la première passe au-dessus du point de rendez-vous. L'important est que les centres des images acquises à des instants différents se retrouvent le plus près possible, au-dessus du même point géométrique. C'est pour ces raisons que nous avons choisi comme critère de proximité la plus faible translation entre deux images.

Remarque : la taille de la fenêtre ΔI (comme celle de la fenêtre ΔJ) sera déterminée grâce à la connaissance des variances des positions intégrées à partir des mesures de cap et vitesse (localisation A/(C,V)).

Les résultats obtenus sont présentés dans la partie 5.7.

5.4 Estimation de mouvement inter-image et construction de mosaïques d'images à partir d'un flux vidéo

Dans cette partie, après une brève introduction sur la construction de mosaïques et plus particulièrement sur la construction de mosaïques sous-marines, nous décrivons quelques algorithmes d'estimation de mouvement pour une séquence vidéo, qui permettent de construire des mosaïques d'images.

5.4.1 Construction de mosaïques d'images sous-marines

Les mosaïques d'images sont très utilisées en imagerie lorsqu'il s'agit d'établir une carte globale ou une vue panoramique d'une zone d'intérêt. Notons qu'elles peuvent être également utilisées dans la localisation des engins sous-marins. Nous pouvons citer les travaux de Negahdaripour [NX02] et [XN01], qui a adapté la construction d'une carte pour la localisation instantanée (SLAM) en milieu sous-marin (voir définition du SLAM en section 2.3.3.2). Ce qui nous intéresse plus particulièrement dans le cadre de nos recherches, reste la construction d'une carte visuelle pour l'exploration d'un site et non pour la localisation instantanée d'un véhicule sous-marin.

Une mosaïque est une juxtaposition d'images de même taille ou de tailles différentes, de résolution identique ou différente. Il est important qu'elles présentent un recouvrement suffisant pour repérer des parties communes. Pour pouvoir juxtaposer deux images, il est nécessaire de déterminer la transformation géométrique reliant ces deux images.

Dans cette partie, un état de l'art sur la construction de mosaïques dans le domaine spécifique de l'exploration sous-marine est proposé. La plupart des algorithmes cités utilisent les techniques de base évoquées dans les sections précédentes, comme l'extraction de caractéristiques et leur mise en correspondance. Il existe essentiellement deux types de méthodes : les unes reposent sur une mise en correspondance de caractéristiques d'intérêt dans les images, les autres reposent sur une analyse du mouvement inter-image, ce que nous verrons plus en détail dans les parties suivantes de ce chapitre (parties 5.4.2 et 5.4.3).

Rappelons que le contexte est celui de l'exploration sous-marine d'une zone d'intérêt par une unique ca-

méra² verticale fixée sur un véhicule sous-marin se déplaçant horizontalement au-dessus du fond marin à une distance réduite du fond (de 3 à 10 mètres) et de façon relativement stable.

Marks a publié dans [MRL94] une méthode qui construit des mosaïques temps réel, qui repose sur les quatre étapes suivantes :

- Filtrage de l'image courante
- Mise en correspondance des primitives de l'image courante avec celles de l'image précédente
- Estimation du mouvement entre les deux images
- Rendu de la mosaïque : "collage" des images.

Le filtrage de l'image est réalisé grâce à un LoG (Laplacien de Gaussien), qui rend une image de contours binaire. L'étape de mise en correspondance repose sur une corrélation. Soient \mathbf{I}_i et \mathbf{I}_{i+1} les images entre lesquelles le déplacement est à estimer. La corrélation est réalisée avec une fenêtre de 20×20 pixels de l'image \mathbf{I}_{i+1} , autour de la position du contour considéré dans l'image \mathbf{I}_i . Pour estimer le déplacement entre les images \mathbf{I}_i et \mathbf{I}_{i+1} , l'auteur utilise un modèle de mouvement à 4 paramètres (translation, zoom et rotation). Il utilise l'estimation la plus simple et directe : les moindres carrés directs. A chaque étape, Marks a choisi une technique simple et rapide pour atteindre le temps réel. En revanche, ces techniques ne sont pas les plus précises ni les plus robustes.

L'étape du rendu de la mosaïque peut être réalisée de plusieurs façons, plus ou moins simples. Il s'agit de coller l'image traitée à la mosaïque déjà construite. La façon la plus simple est de faire une rotation et une translation de l'image suivant les paramètres du mouvement calculés précédemment.

Santos-Victor et Gracias ont mis au point une méthode de construction de mosaïques qu'ils ont publiée et améliorée dans leurs articles [GSV98, GSV99, GSV00, JSV01]. Le principe de cette méthode repose sur une estimation du mouvement d'image, qui revient à estimer une homographie inter-image (transformation plane). Leur méthode est composée des mêmes quatre étapes que celle de Marks. Ils proposent une méthode plus robuste mais incidemment leur construction de mosaïque est plus lourde en temps de calcul et le temps réel n'est pas atteint. Pour la première étape, ils réalisent une extraction de points grâce au détecteur de Harris. La mise en correspondance s'effectue de la même façon que Marks, par corrélation (critère SSD). Ils ajoutent une étape d'estimation robuste pour éliminer les mauvaises mises en correspondance (LMedS). A partir des mises en correspondance valides, ils estiment le déplacement inter-image grâce à un modèle à 8 paramètres.

D'autres approches autres que la mise en correspondance de primitives d'intérêt dans l'image ont été mises au point. Garcia, dans [GCP00] et [GBCA01], utilise des "patches" texturés pour effectuer la mise en correspondance d'images couleur. En effet, en imagerie sous-marine, il arrive que peu de primitives soient détectables. Aussi, il introduit la notion de texture en plus de la notion de primitive, pour améliorer l'étape de mise en correspondance avant l'estimation de la transformation. Cette alternative peut être très utile lors du survol de champs d'algues, mais elle ne fonctionnerait pas en présence de structures comme des pipes ou des épaves.

²Notons toutefois qu'il existe des méthodes utilisant un montage stéréoscopique pour construire des mosaïques d'images. On peut se référer par exemple aux travaux de [ZRH01].

Parmi les méthodes d'estimation du mouvement inter-image, Rolfes [RR04] a proposé une méthode reposant sur un suivi de contours (snakes ou contours déformables statistiques, dont on prédit la position d'une image à l'autre, comme pour le suivi de points). Le lecteur peut se reporter à une abondante littérature sur le sujet des contours actifs et à l'article fondateur de Kass [KWT88]. Pour que cette méthode soit applicable, les scènes observées doivent avoir deux champs délimités (par exemple, une partie de l'image représente un champ d'algues, tandis que l'autre partie représente un sol sableux). La méthode n'est donc pas adaptée aux cas du survol d'une épave ou d'une zone d'intérêt comportant plus de deux champs délimités (cas de la séquence des amphores où plusieurs objets sont présents, comme les algues ou les amphores).

Afin d'améliorer la précision de la mosaïque, Marks propose dans [MRL95] une étape supplémentaire, bien que très lourde en temps de calcul et nécessitant une mémoire adéquate. Elle consiste à mettre également en correspondance l'image courante et son image adjacente, appartenant à la colonne précédente³ dans la mosaïque, par des techniques de corrélation. Cette problématique s'apparente à la nôtre puisqu'elle tient compte d'information sur la géométrie de l'exploration de la zone d'intérêt : on suppose que l'engin repasse au-dessus d'une même zone. Cependant, cette étape additionnelle ne tient compte que des informations image. Ainsi, on a aussi affaire à une dérive de la mosaïque et peut-être également à une déformation non-contrôlée de celle-ci.

Dans la suite de ce chapitre, nous allons présenter quelques méthodes d'estimation de mouvement inter-image qui supposent un fort recouvrement inter-image (environ 90%), ce qui est le cas des séquences vidéo. Il est donc possible d'avoir un a priori sur les transformations estimées. Les méthodes KLT [ST94] et RMR [OB95], plus particulièrement, seront exposées pour être utilisées par la suite. Puis nous étudierons la possibilité d'intégrer les instants de rendez-vous afin de réaliser une localisation AR/I, en suivant le même modèle que la localisation AR/(C,V).

5.4.2 Suivi de points par l'algorithme KLT [ST94]

Le suivi de points ou "tracking" est une méthode d'estimation du déplacement entre deux images successives à partir d'un modèle de mouvement et en présupposant qu'un point doit se retrouver dans une fenêtre de taille donnée, dans l'image suivante. Le principe est illustré figure 5.7 page suivante. On rappelle que P_1 est le point de l'image \mathbf{I}_1 , P_2 le point correspondant dans l'image \mathbf{I}_2 et d la translation liant \mathbf{I}_1 à \mathbf{I}_2 . La méthode KLT proposée par Kanade, Lucas et Tomasi [LK81][ST94] repose sur une extraction de points dans la première image, et le suivi de ces points dans les images suivantes. Seuls les voisinages des points caractéristiques sont traités d'une image à la suivante. On suppose dans le cas de déplacements inter-image relativement faibles (caméra bougeant lentement), que le voisinage du point P_1 suivi dans l'image \mathbf{I}_1 se retrouve dans l'image \mathbf{I}_2 par une translation :

$$\mathbf{I}_2(x, y) = \mathbf{I}_1(x - \alpha, y - \beta) + n(x, y) \quad (5.28)$$

³On rappelle que la zone d'intérêt est quadrillée d'où la présence de trajectoires de l'engin en ligne droite, ce qui au niveau mosaïque, est représenté par une colonne d'images.

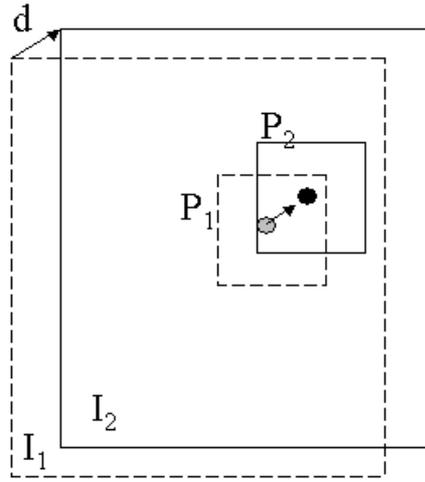


FIG. 5.7 – Principe du suivi de points.

où $\mathbf{d} = (\alpha, \beta)^T$ est le vecteur translation entre l'image \mathbf{I}_1 et l'image \mathbf{I}_2 , et $n(x, y)$ est le bruit à la position (x, y) . Le vecteur \mathbf{d} est alors choisi de manière à minimiser l'erreur ϵ calculée sur une fenêtre de voisinage W de la manière suivante :

$$\epsilon = \sum_{\mathbf{x} \in W} [\mathbf{I}_1(\mathbf{x} - \mathbf{d}) - \mathbf{I}_2(\mathbf{x})]^2 \omega(\mathbf{x}) \quad (5.29)$$

où $\omega(\mathbf{x})$ est une fonction de pondération et $\mathbf{x} = (x, y)^T$. En général, $\omega(\mathbf{x}) = 1$, mais elle peut aussi prendre une forme gaussienne si on veut donner plus d'importance au centre de la fenêtre.

Pour minimiser ϵ , nous différencions l'équation (5.29). En écrivant le développement de Taylor au premier ordre, on obtient :

$$\mathbf{I}_1(\mathbf{x} - \mathbf{d}) = \mathbf{I}_1(\mathbf{x}) - g g^T \mathbf{d} \quad (5.30)$$

où $g = \left[\frac{\partial \mathbf{I}_1(\mathbf{x})}{\partial x}, \frac{\partial \mathbf{I}_1(\mathbf{x})}{\partial y} \right]^T$ est le gradient de \mathbf{I}_1 .

En développant l'équation (5.29), nous obtenons tous calculs faits, une équation du type :

$$\boxed{Z \mathbf{d} = \mathbf{e}} \quad (5.31)$$

avec

$$\begin{cases} Z = \sum_W g g^T \omega(\mathbf{x}) \\ \mathbf{e} = \sum_W [\mathbf{I}_1(\mathbf{x}) - \mathbf{I}_2(\mathbf{x})] g(\mathbf{x}) \omega(\mathbf{x}) \end{cases} \quad (5.32)$$

Le but est d'estimer \mathbf{d} dans l'équation (5.31), connaissant Z et \mathbf{e} . Pour déterminer s'il est possible de "suivre" une primitive, il faut que le système (5.31) soit bien conditionné et robuste au bruit. Pour ce faire, il est nécessaire que le rapport entre les deux valeurs propres de Z ne soit pas trop grand (système bien conditionné) et que ces valeurs soient supérieures à un seuil (robustesse au bruit).

5.4.3 Flot optique et pyramide multi-résolution : cas particulier du RMR [OB95]

La méthode d'analyse multi-résolution (RMR), proposée par Odobez et Bouthémy [OB95] et intégrée par l'Ifremer dans le logiciel de mosaïque optique géo-référencée MATISSER® [ROSP01, ABO⁺04] permet d'estimer le mouvement global en deux dimensions d'une image ou d'un bloc contenu dans l'image sans tenir compte des mouvements aléatoires. L'estimation robuste multi-échelle consiste à utiliser pour chaque image une pyramide gaussienne de L niveaux afin d'estimer au mieux le mouvement 2D entre deux images successives à partir d'un modèle de mouvement à plusieurs paramètres. Soulignons que seule l'estimation du mouvement global de l'image sera abordée ici, tandis que le rendu de la mosaïque ne sera pas traité. Après avoir explicité deux principes généraux, le flot optique et la pyramide multi-résolution, nous détaillerons la méthode du RMR.

5.4.3.1 Flot optique

Le flot optique est défini comme le mouvement apparent du modèle de luminance, c'est-à-dire qu'il correspond au mouvement inter-image, donné par les variations spatio-temporelles de l'intensité lumineuse. Le mouvement d'une image à la suivante n'étant qu'une projection 2D dans le plan image d'un mouvement 3D relatif entre la scène et la caméra, le flot optique est également appelé champ des vitesses apparentes. Pour déterminer ce champ de vecteur, plusieurs méthodes sont disponibles : les méthodes *par filtrage fréquentiel*, les méthodes *discrètes* et les méthodes *différentielles* [Smi97].

La plupart des approches *fréquentielles* reposent sur une hypothèse de mouvement translationnel pur et sont assez coûteuses en temps de calcul car elles requièrent l'analyse de longues séquences d'images. Leurs principes ne seront pas abordés ici.

Les approches *discrètes* sont celles exposées dans la partie 5.2.3. Elles sont composées d'une phase de détection de primitives (coins, points d'intérêt...) suivie d'une phase de mise en correspondance.

Nous allons plus particulièrement détailler dans cette partie les méthodes *différentielles*, qui sont utilisées dans la méthode RMR déjà utilisée à l'Ifremer pour la construction de mosaïques d'images.

Les méthodes différentielles sont basées sur l'analyse du gradient spatio-temporel de la fonction de l'intensité lumineuse. L'hypothèse fondamentale est que la luminance d'un point est constante d'une image à la suivante. Ainsi, la variation de luminance est uniquement due au mouvement. Cette hypothèse se traduit de la façon suivante :

$$\mathbf{I}(x, y, t) = \mathbf{I}(x + dx, y + dy, t + dt) \quad (5.33)$$

où $\mathbf{I}(x, y, t)$ est l'intensité lumineuse d'un point correspondant au pixel (x, y) de l'image l'instant t . A l'instant $t + dt$, le point s'est déplacé de la quantité (dx, dy) . Si l'on suppose que l'intensité lumineuse est une fonction différentiable, cette hypothèse peut également se réécrire selon :

$$d\mathbf{I} = \frac{\partial \mathbf{I}}{\partial x} dx + \frac{\partial \mathbf{I}}{\partial y} dy + \frac{\partial \mathbf{I}}{\partial t} dt = 0 \quad (5.34)$$

où $\frac{\partial \mathbf{I}}{\partial x}$ et $\frac{\partial \mathbf{I}}{\partial y}$ sont les gradients spatiaux et $\frac{\partial \mathbf{I}}{\partial t}$ le gradient temporel. On notera dans la suite $\mathbf{I}_x = \frac{\partial \mathbf{I}}{\partial x}$, $\mathbf{I}_y = \frac{\partial \mathbf{I}}{\partial y}$, $\mathbf{I}_t = \frac{\partial \mathbf{I}}{\partial t}$ et le vecteur vitesse $(u, v)^T = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)^T$. On obtient alors l'équation de contrainte du flot optique :

$$\boxed{\mathbf{I}_x u + \mathbf{I}_y v + \mathbf{I}_t = 0} \quad (5.35)$$

Cette équation ne permet pas de déterminer de manière unique le flot optique, car en chaque point de l'image il n'existe qu'une seule contrainte scalaire pour déterminer un vecteur de deux coefficients (u, v) . Pour trouver LA solution, il est nécessaire d'avoir une équation de contrainte supplémentaire. On peut fournir un modèle de vitesse [OB95] ou bien introduire des contraintes de lissage global. Dans ce cas, le champ des vitesses est défini par le minimum d'une fonctionnelle sur toute l'image.

5.4.3.2 Intérêt d'une approche multi-résolution pour la robustesse

L'extraction "brute" de points d'intérêt présente un défaut, c'est-à-dire que des points inintéressants ou peu robustes peuvent être détectés. Pour remédier à ce problème, une approche multi-résolution utilisant une pyramide d'images est une méthode complémentaire intéressante. Une pyramide a pour but d'éliminer le bruit en diminuant l'échelle et la résolution d'une image.

Une pyramide d'images gaussienne est formée à la base de l'image pleine résolution. Lorsque l'on passe à un niveau plus grossier, la taille de la pyramide est divisée par quatre. Afin que la structure de l'image soit conservée lors du passage d'un niveau à un niveau plus grossier (une droite reste une droite), et ce malgré la perte de résolution, un filtre gaussien est appliqué. Le principe de construction d'une pyramide à L niveaux est illustré sur la figure 5.8. Un exemple est proposé sur les figures 5.9(a) et 5.9(b) page suivante qui représentent les deux premiers niveaux de la pyramide construite à partir de l'image 3 de la séquence "Lucky Luke".

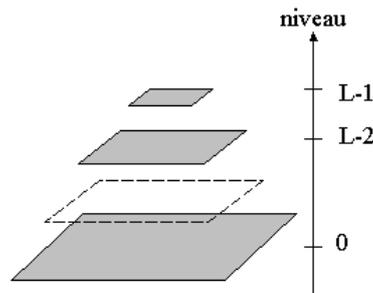


FIG. 5.8 – Principe de la construction d'une pyramide d'images.

5.4.3.3 Modèle introduit

Le mouvement 2D dans l'image peut être modélisé de la façon suivante :

$$V_A(X_i) = \begin{bmatrix} u_A(X_i) \\ v_A(X_i) \end{bmatrix} \quad (5.36)$$

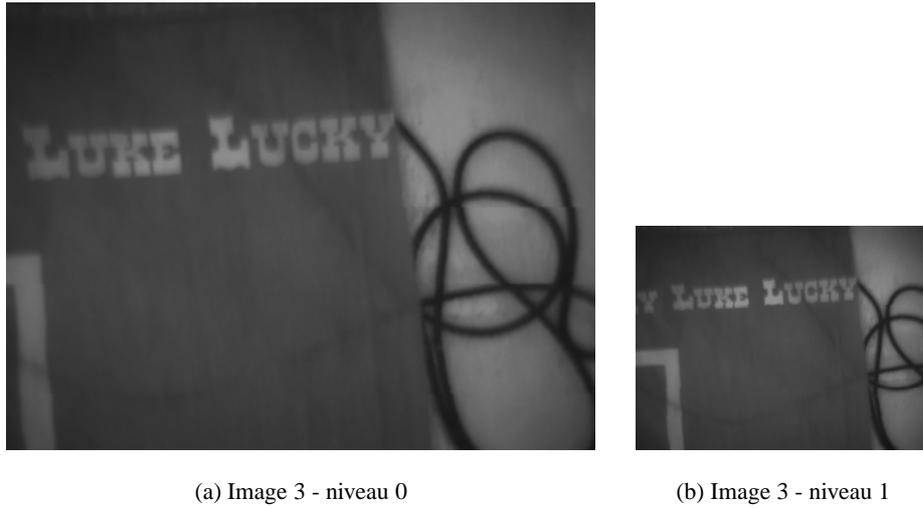


FIG. 5.9 – Deux niveaux de la pyramide de l'image 3 de la séquence "Lucky Luke".

où $X_i = (x_i, y_i)$ est le point courant. u_A et v_A sont des fonctions des coordonnées du point X_i , qui modélisent la vitesse selon l'axe des abscisses et des ordonnées dans l'image. Plusieurs modélisations du mouvement sont possibles : constant, affine, quadratique, etc...

Dans ce qui suit, le modèle retenu est le modèle affine. Il permet de prendre en compte un grand nombre de types de déplacements : translation, rotation, déformation et changement d'échelle, même si, en général, un mouvement 3D projeté dans une image donne un champ des vitesses représenté par un modèle quadratique.

$$\begin{cases} u_A(X_i) = a_1 + a_2x_i + a_3y_i \\ v_A(X_i) = a_4 + a_5x_i + a_6y_i \end{cases} \quad (5.37)$$

Dans ce cas précis, l'équation 5.36 peut se réécrire de la façon suivante :

$$V_A(X_i) = B(X_i)A \quad (5.38)$$

où $A^t = (a_j)_{j=1..6}$ est un vecteur et $B(X_i)$ est une matrice. Dans ce cas précis, A est de dimension 1×6 et $B(X_i)$ 2×6 et s'expriment de la façon suivante :

$$A = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6]^T \quad B(X_i) = \begin{bmatrix} 1 & x_i & y_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_i & y_i \end{bmatrix} \quad (5.39)$$

5.4.3.4 Critère de minimisation

Pour rechercher le mouvement, nous allons tenter de résoudre l'équation du flot optique 5.35. Pour chaque point X_i , elle peut être réécrite sous forme vectorielle :

$$\frac{d\mathbf{I}}{dt}(X_i, t) = \vec{V}(X_i, t) \cdot \vec{\nabla} \mathbf{I}(X_i, t) + \mathbf{I}_t(X_i, t) \quad (5.40)$$

où $\left\{ \begin{array}{l} \frac{d\mathbf{I}}{dt}(X_i, t) \text{ est la dérivée totale de la fonction intensité par rapport au temps} \\ \vec{V}(X_i, t) = \left(\frac{dx}{dt}, \frac{dy}{dt} \right) \\ \vec{\nabla}\mathbf{I} = (\mathbf{I}_x, \mathbf{I}_y) \text{ est le gradient spatial de la fonction d'intensité} \end{array} \right.$

Rappelons que le terme $\frac{d\mathbf{I}}{dt}(X_i, t)$ correspond à la variation d'intensité du point X_i sur sa trajectoire 2D. Ainsi, l'hypothèse d'invariance de l'intensité d'un point sur sa trajectoire conduit à l'équation de contrainte du mouvement apparent :

$$\frac{d\mathbf{I}}{dt}(X_i, t) = 0 \quad (5.41)$$

Soit :

$$\vec{V}(X_i, t) \cdot \vec{\nabla}\mathbf{I}(X_i, t) + \mathbf{I}_t(X_i, t) = 0 \quad (5.42)$$

Dans le cas où des changements globaux d'illumination surviendraient, on ne peut plus émettre l'hypothèse d'invariance de l'intensité d'un point X_i sur sa trajectoire. On peut donc étendre l'équation de contrainte du mouvement apparent 5.41 en introduisant un terme d'illumination ξ , constant :

$$\frac{d\mathbf{I}}{dt}(X_i, t) = -\xi \quad (5.43)$$

On définit l'expression suivante :

$$r_i = \frac{d\mathbf{I}}{dt}(X_i, t) + \xi = \vec{V}(X_i, t) \cdot \vec{\nabla}\mathbf{I}(X_i, t) + \mathbf{I}_t(X_i, t) + \xi \quad (5.44)$$

Si l'on introduit le modèle de mouvement 2D V_A , r_i peut se réécrire de la façon suivante :

$$r_i = \chi_i \theta - Y_i \quad (5.45)$$

où $\theta^T = (A^T, \xi)$, $Y_i = -\mathbf{I}_t(X_i)$ et $\chi_i = (\nabla\mathbf{I}(X_i)^T B(X_i), 1)$.

Minimisation d'une erreur Pour pouvoir estimer le mouvement global dans l'image ou un bloc inclus dans l'image, c-à-d. estimer le paramètre θ il convient de trouver l'argument qui minimise une fonction d'erreur. On peut par exemple minimiser le critère quadratique suivant par rapport à θ :

$$Q(\theta) = \sum_{X_i \in W} r_i^2 = \sum_{X_i \in W} (\chi_i \theta - Y_i)^2 \quad (5.46)$$

où W est la région où la minimisation est réalisée. Elle peut être l'image entière ou bien un bloc. On obtient la solution classique obtenue par minimisation des moindres carrés :

$$\hat{\theta} = \left[\sum_{X_i \in W} \chi_i^T \chi_i \right]^{-1} \sum_{X_i \in W} \chi_i^T Y_i \quad (5.47)$$

Dans ce qui suit, on approxime la dérivée totale $\frac{d\mathbf{I}}{dt} = -\xi$ par une différence finie. On pose $\delta X_i = V(X_i) \delta t$, avec δt l'intervalle de temps entre l'acquisition de deux images successives. Pour simplifier les notations,

on posera $\delta t = 1$. On peut alors réécrire l'expression r_i sous la forme :

$$DFD_\theta(X_i) = \mathbf{I}(X_i + B(X_i)A, t + 1) - \mathbf{I}(X_i, t) + \xi \quad (5.48)$$

Le nom de l'expression DFD est choisi par les auteurs pour "model-based Displaced Frame Difference".

Le critère Q se réécrit donc :

$$Q(\theta) = \sum_{X_i \in W} (DFD_\theta(X_i))^2 \quad (5.49)$$

Robustesse aux mesures aberrantes : choix d'un M-estimateur Pour des raisons de robustesse aux mesures aberrantes, Odobez et Bouthémy ont choisi un estimateur de la classe des M-estimateurs (le 'M' correspond à une estimation selon le maximum de vraisemblance). Cet estimateur consiste en la minimisation de la somme suivante :

$$\hat{\theta} = \arg \min \sum_i \rho(y_i - M(\theta, X_i), \sigma) \quad (5.50)$$

où ρ est la fonction bipoids de Tuckey⁴. Le critère Q (équation (5.49)) est donc reformulé selon :

$$Q_r(\theta) = \sum_{X_i \in W} \rho(DFD_\theta(X_i), C) \quad (5.52)$$

Transformation du problème de M-estimation Le M-estimateur est réputé robuste aux points aberrants mais il est très lourd à mettre en oeuvre. La méthode des Moindres Carrés Pondérés Itérés (MCPI ou IRLS en anglais) est classiquement employée pour résoudre le problème de M-estimation. Le problème de M-estimation se traduit en problème des moindres carrés pondérés selon :

$$\sum_i \rho(r_i) = \sum_i \frac{1}{2} w_i r_i^2 \quad (5.53)$$

avec $r_i = y_i - M(\theta, X_i)$. Les poids w_i sont donnés par la dérivée ψ de fonction de Tuckey $\rho : w_i = \frac{\psi(r_i)}{r_i}$.

Estimation incrémentale Pour la méthode du RMR, les auteurs ont choisi de réaliser la minimisation de l'erreur E_r définie par la formule (5.52) de façon incrémentale. Si $\hat{\theta}_k$ est l'estimation courante de θ , on peut alors écrire $\hat{\theta}_{k+1} = \hat{\theta}_k + \Delta\hat{\theta}_k$. Après un développement de l'expression DFD au premier ordre autour du point $X_i + B_i\hat{A}_k$ (détaillé dans l'article), le critère Q_r (eq. 5.52) devient :

$$Q_r(\Delta\theta) = \sum_{X_i \in W} \rho(r'_i, C) \quad (5.54)$$

où $r'_i = \mathbf{I}(X_i + B_i\hat{A}_k, t + 1) - \mathbf{I}(X_i, t) + \hat{\xi}_k + \nabla\mathbf{I}^t(X_i + B_i\hat{A}_k, y + 1)B_i\Delta A_k + \Delta\xi_k$. Pour éviter un trop grand nombre de minima locaux de la fonction, on ajuste le paramètre C à chaque incrément.

⁴La fonction bipoids de Tuckey est la primitive de la fonction ψ définie de la façon suivante :

$$\psi(x, C) = \begin{cases} x(C^2 - x^2)^2 & \text{si } |x| < C \\ 0 & \text{sinon} \end{cases} \quad (5.51)$$

où C est une constante de valeur comprise entre 5 et 20.

Enfin, on transforme le problème de M-estimation en un problème des MCPI comme précédemment décrit.

5.4.3.5 Estimation multi-résolution

Odohez et Bouthémy proposent une estimation robuste multi-résolution. Une pyramide d'images gaussienne est donc formée à la base de l'image pleine résolution.

Le principe de l'estimation robuste consiste à considérer les paramètres du mouvement estimés au niveau l de la pyramide comme initialisation des paramètres à estimer au niveau $l - 1$ plus fin. Remarquons que pour le passage d'un niveau de la pyramide au niveau inférieur, les termes constants de A sont multipliés par 2, les termes linéaires sont identiques et les termes quadratiques sont divisés par 2.

Le principe du RMR est résumé dans le tableau 5.1 (tableau repris de l'article [OB95]).

| |
|--|
| $C \leftarrow \text{Max}(\mathbf{I}_t^{L-1})$ $\hat{\theta}^{L-1} \leftarrow \text{MCPI}^a(\text{residual}(5), C)$ <p>Pour $l = \text{niveau } L - 1$ jusqu'au niveau 0, faire</p> $\text{iter} \leftarrow 0$ <p>faire</p> $C \leftarrow f(C)$ $\hat{\Delta}\theta^l \leftarrow \text{MCPI}(\text{residual}(10), C)$ $\hat{\theta}^l \leftarrow \hat{\theta}^l + \hat{\Delta}\theta^l \text{ et } \text{iter} \leftarrow \text{iter} + 1$ <p>tant que ($\text{iter} < \lambda$ et $\ \hat{\Delta}\theta^l\ > \frac{d}{2^l}$)</p> <p>si $l \neq 0$ $\hat{A}^{L-1} \leftarrow P(\hat{A}^L)$, $\hat{\epsilon}^{L-1} \leftarrow P(\hat{\epsilon}^L)$</p> $\hat{\theta}_{est} \leftarrow \hat{\theta}^0$ |
| <p>^aMCPI signifie Moindres Carrés Pondérés Itérés</p> |

TAB. 5.1 – Méthode RMR.

Remarque : Les biais et variances des paramètres estimés du modèle peuvent être déterminés. En effet, l'estimation robuste revient à utiliser une technique des moindres carrés pondérés itérés, pour laquelle les biais et variances des paramètres sont connus. Cependant, on ne calcule ainsi que le biais et la variance de l'estimateur à la dernière étape du schéma incrémental d'estimation. Le lien entre ces biais et variances et le contenu d'information de l'image (son entropie⁵) n'étant pas connue, ces biais et variances ne sont pas exploitables.

5.4.4 Comparaison des méthodes d'estimation KLT et RMR

La méthode RMR est une méthode d'estimation robuste, qui repose sur l'utilisation d'une pyramide pour améliorer la rapidité de l'algorithme et sa fiabilité. C'est une méthode d'estimation globale, qui peut estimer plusieurs paramètres (translation, rotation, zoom, etc...) selon le modèle de mouvement choisi.

La méthode de KLT, quant à elle, est une méthode rapide, qui repose sur une sélection de points et leur suivi.

⁵En physique statistique, l'entropie mesure l'information du système étudié. Dans le cas d'une image \mathbf{I} de N pixels, si on pose n_i le nombre de pixels par niveau de gris, la probabilité qu'un pixel X soit dans le niveau i est donnée par $p(X \in i) = \frac{n_i}{N}$. On définit alors l'entropie de l'image comme $H(\mathbf{I}) = -\sum_i p(X \in i) \ln p(X \in i)$.

Ce n'est donc pas une méthode globale. D'autre part, elle ne permet pas d'estimer de multiples paramètres de transformation, comme le peut la méthode RMR.

Une étude comparative a été réalisée sur des séquences d'images acquises en bassin. Elle est présentée dans la partie 5.8.3.

5.5 Localisation AR/I

Après avoir exposé deux algorithmes d'estimation du mouvement pour la construction de mosaïques, nous avons étudié la possibilité de réaliser la localisation AR/I à partir de la localisation A/I estimée par le RMR ou le KLT.

Étant donné que le but de ces recherches est de fusionner une estimation de la localisation à partir de mesures de cap et vitesse et une localisation à partir d'une séquence d'images, il est important de garder une cohérence entre les deux localisations pour réaliser la fusion. Ainsi, les deux localisations sont réalisées en utilisant les mêmes hypothèses. Celles-ci sont donc la réalisation d'une estimation aller-retour connaissant les points extrémaux, ainsi que la gestion des instants de rendez-vous estimés.

De façon analogue à réalisation de la localisation AR/(C,V) du chapitre 3, il s'agit donc de réaliser une estimation des positions du véhicule sous-marin à partir des estimations de mouvements donnés par le RMR, des positions extrémales connues, et des instants de rendez-vous estimés. La trajectoire est donc considérée comme un graphe et un parcours de graphe par le plus court chemin (détaillé dans la partie 3.1.2) est réalisé.

La localisation A/I est réalisée par une intégration des mesures de déplacement estimés par le RMR à partir d'une position initiale connue ou estimée (dans le cas d'une trajectoire complexe). La localisation R/I est réalisée à partir d'une intégration rétrograde des déplacements estimés par le RMR, à partir de la position finale, connue ou estimée. L'idéal serait, comme dans le cas de la localisation A/(C,V) ou R/(C,V), de réduire le biais. Malheureusement, dans notre cas, nous n'avons pas accès aux biais (voir la remarque dans la partie 5.4.3). Ainsi, nous le négligerons pour réaliser la localisation AR/I. La fusion des localisation A/I et R/I est réalisée ici par un estimateur des moindres carrés pondérés.

En revanche, pour mettre en oeuvre l'estimateur des moindres carrés pondérés, il est nécessaire de connaître les variances des paramètres d'estimation du RMR. Ces variances ne pouvant pas être déterminées (remarque dans partie 5.4.3), l'idée consiste à considérer que la variance est invariante dans le temps. Cette hypothèse n'est pas dénuée de sens puisque lors de l'exploration d'une zone, quasiment la même quantité d'information est visualisée sur chaque image. Dans ce cas, les paramètres de l'estimation par la pyramide d'image RMR ne varient pas. Si σ est constant, alors la variance de l'intégration des déplacements augmente linéairement avec le temps : $\mathbf{Var} \left(\sum_{i=1}^k \Delta x_i \right) = k\sigma^2$ (voir annexe A.4.3).

Les résultats d'estimation d'une localisation AR/I sont exposés dans la section 5.8.4.

5.6 Cadre expérimental

Les expérimentations ont été effectuées dans le bassin intérieur de l'Ifremer (centre de Méditerranée), à l'aide d'un portique instrumenté monté sur le bassin et d'une caméra. Ce bassin est rectangulaire, de dimensions $2\text{m} \times 3\text{m}$ et de profondeur 2.5m . Le matériel ainsi que les conditions d'expérimentation sont présentés dans le paragraphe suivant.

Nous présentons ensuite différents résultats. La partie 5.7 propose des résultats relatifs à l'étude de la PISA et à son utilisation dans la recherche d'une estimation des instants de rendez-vous. Dans la partie 5.8 sont tout d'abord présentés quelques résultats relatifs à la correction des problèmes d'intensité lumineuse des images sous-marines. Puis une comparaison des algorithmes RMR et KLT est effectuée sur des séquences acquises en bassin. Enfin, dans la partie 5.8.4 sont présentés des résultats relatifs à la localisation AR/I.

5.6.1 PISCATOR et matériel

PISCATOR

Le système PISCATOR (Portique Instrumenté Sous-marin de Calibration et de qualification pour l'Acoustique, la Trajectographie, l'Optique et la Robotique), présenté sur la figure 5.10 page suivante, permet le positionnement sous-marin de charges embarquées dans le bassin d'essai du centre de l'IFREMER se prête tout à fait à la réalisation des données réelles. En effet PISCATOR offre une répétitivité de mouvement et une précision sub-millimétrique en position qui tout en permettant une validation concrète, associent la maîtrise de plusieurs paramètres expérimentaux.

PISCATOR est un portique robotique à deux axes linéaires horizontaux (X, Y) et un troisième axe vertical Z immergeable, piloté en translation et rotation. La partie terminale de l'axe vertical est équipée d'un poignet à deux rotations instrumentées par des codeurs optiques pour une mesure des angles à grande précision.

PISCATOR est essentiellement dédié à l'essai, la calibration et la qualification de capteurs de différentes natures utilisés comme équipement des engins sous-marins : équipements optiques (caméras, projecteurs), acoustiques (sondeurs, sonars) et systèmes de navigation (centrales inertielle, loch, gyromètres, centrales d'attitude).

Les atouts principaux du portique résident dans ses caractéristiques techniques. A sa grande précision de positionnement et de contrôle de trajectoires (commande numérique NUM) s'ajoute une capacité d'emport importante (jusqu'à 35 kg dans l'air et 20 kg dans l'eau) et des vitesses de déplacement importantes.

La précision absolue obtenue sur les trois axes linéaires est sub-millimétrique avec une répétitivité de mouvement inférieure au micron mètres. Quant à la précision absolue des rotations, elle est inférieure au centième de degrés.

Matériel

Étant donné que le bassin fait environ 6 m^2 et que nous avons besoin d'avoir des objets sur toute cette surface, nous avons choisi de déposer au fond du bassin un drap plan, qui représente une scène structurée (motifs de dessin animé).



FIG. 5.10 – Le portique PISCATOR sur le bassin d'essai.

Ainsi, comme exposé dans le chapitre 1, nous avons pris pour nos expériences des hypothèses simplifiant notre problématique, qui, rappelons-le, s'inscrivait dans une nouvelle thématique de recherche pour l'Ifremer.

5.6.2 Le système optique

La caméra utilisée, la DX 20 du fabricant KAPPA, est une caméra numérique CCD. Afin d'assurer l'étanchéité du système optique, la caméra est insérée dans une enceinte étanche. Le logiciel d'acquisition fourni avec la KAPPA (KAPPA Image Base ou KIB) permet le contrôle de la caméra, la prise de vues d'images, la gestion et l'archivage ainsi que la mesure et le travail sur image. Le module Time propose des fonctions de contrôle de temps pour la gestion d'acquisition de séquences d'images et également le contrôle d'appareils externes à travers le port parallèle (la synchronisation nous intéresse particulièrement).

Par le biais de ce logiciel, nous avons configuré la fréquence d'acquisition des images à une image par seconde. D'autre part, la synchronisation a été réalisée avec le système PISCATOR. Le robot ne se déplace qu'à partir du moment où une impulsion est donnée par le logiciel. L'acquisition des images démarre au même moment.

Au niveau de l'éclairage, 8 spots lumineux équipent le bassin intérieur. L'éclairage adopté est donc relativement diffus. Un spot lumineux a été fixé à proximité de la caméra pour reproduire les conditions d'acquisitions en milieu sous-marin. Mais à cause de la faible profondeur du bassin et de phénomènes de

réflexion sur le drap déposé au fond de l'eau, la pose de ce spot lumineux. Seuls les 8 projecteurs de la piscine ont été utilisés tout au long des acquisitions.

5.6.3 Les trajectoires programmées

Les dimensions du bassin intérieur sont de $2\text{m} \times 3\text{m}$, ce qui a été un facteur limitant à la réalisation de trajectoires. Nous avons donc choisi d'effectuer deux types de trajectoires : l'une représente le "alpha" de la partie 3 (voir figure 5.11(a)), qui comporte 196 points (ou images), l'autre représente une trajectoire à deux boucles (voir figure 5.11(b)) qui comporte 223 points. Le robot ne pouvant être programmé qu'en translation, les transformations inter-images sont composées d'une translation seule. D'autre part, étant donné que la trajectoire est plane, il n'existe pas d'effet de zoom ou de changement d'échelle pendant l'acquisition d'une séquence d'images.

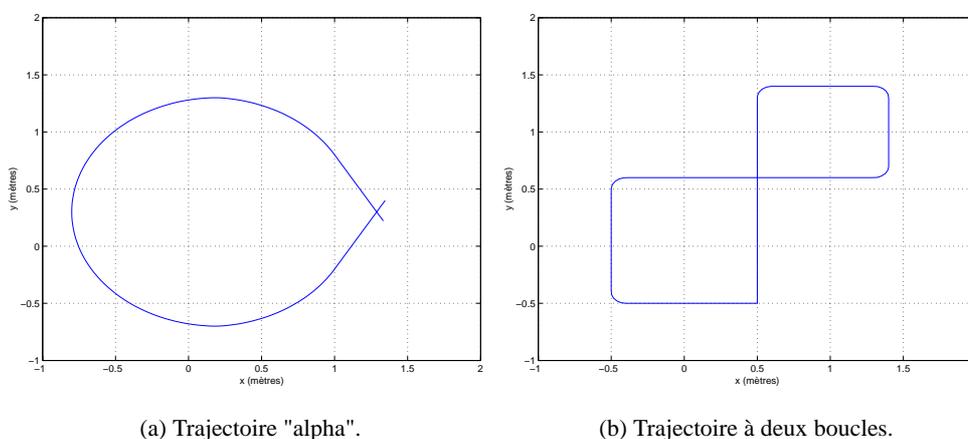


FIG. 5.11 – Trajectoires réalisées dans le bassin d'essais intérieur.

5.6.4 Instants de rendez-vous

Sur les figures 5.12(a) et 5.12(b) page suivante est représentée la trajectoire "alpha" programmée, avec un zoom sur la partie de la trajectoire comportant un instant de rendez-vous. D'après la trajectoire programmée, le couple d'instant de rendez-vous est $(I = 4, J = 192)$.

5.7 Résultats relatifs à la recherche des instants de rendez-vous

5.7.1 Etude des paramètres de l'algorithme PISA

Afin de mettre en oeuvre la méthode exposée précédemment, nous avons déterminé quels paramètres étaient les mieux adaptés à nos images.

Rappelons que les paramètres réglables par l'opérateur sont le type de détecteur de points, la méthode de seuillage par hystérésis ainsi que les seuils associés, l'estimateur et le nombre d'itérations de l'estimateur.

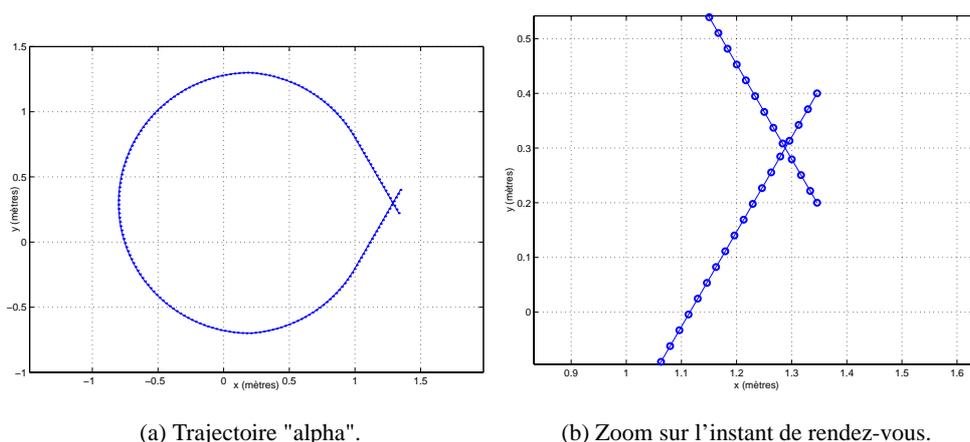


FIG. 5.12 – Zoom sur l'instant de rendez-vous de la trajectoire "alpha".

Les tests ont été effectués sur deux images d'une séquence acquise en piscine, "Lucky Luke". Elles sont illustrées sur les figures 5.13(a) et 5.13(b).



(a) Image 3

(b) Image 191

FIG. 5.13 – Images 3 et 191 de la séquence "Lucky Luke".

5.7.1.1 Paramètres de détection des points d'intérêt et mise en correspondance avant estimation

Dans cette partie, le détecteur de Harris est retenu pour l'extraction de points. Nous étudions ici les méthodes de seuillage : le pourcentage du maximum de la norme du gradient et le pourcentage de points éliminés. La méthode consistant à fixer le nombre de points détectés est abandonnée car nous n'avons pas de contrainte particulière sur le nombre de points détectés.

Dans le tableau qui regroupe les résultats, nous avons ajouté le nombre d'appariements possibles avant l'étape d'estimation, c'est-à-dire après seuillage de Mahalanobis sur les vecteurs d'attributs. Le seuil de Ma-

halanobis sur la distance entre vecteurs d'attributs est fixé à 6.1. Les résultats sont regroupés dans le tableau 5.2. Nous pouvons remarquer que les points sélectionnés par le détecteur de Harris dépendent fortement de la méthode de seuillage employée, ainsi que de ses paramètres. La méthode de seuillage par pourcentage du maximum apparaît comme étant la plus adaptée aux images choisies : le nombre de points détectés n'est pas trop élevé. En effet, quels que soient les seuils fixés, le pourcentage de la norme du gradient rendait trop de points. Pour la méthode de seuillage par pourcentage du maximum, les seuils bas et haut ont été fixés à 5% et 5%. En effet, cela semble être le meilleur choix de paramètres car d'autres tests ont été réalisés avec des seuils différents et souvent trop peu de points étaient détectés.

NB : le nombre important de couples de points (plus de 17 000) est dû au fait que les appariements multiples sont possibles.

| Méthode de seuillage | Nb points détectés 1ère image | Nb points détectés 2ème image | Nb appariements possibles avant estimateur |
|---|-------------------------------|-------------------------------|--|
| Pourcentage max norme du gradient 5%-5% | 393 | 357 | 817 |
| Pourcentage points éliminés 5%-5% | 17692 | 17791 | 2457 |

TAB. 5.2 – Etude des paramètres des méthodes de seuillage par hystérésis.

5.7.1.2 Etude du nombre d'itérations des estimateurs proposées par Sistiaga

Dans cette partie, nous étudions l'influence du nombre d'itérations des estimateurs proposés par Sistiaga sur le nombre de couples de points retenus pour réaliser l'estimation. En effet, un grand nombre de points retenus pour l'estimation implique un nombre non négligeable de couples de points aberrants à gérer dans l'estimation. Un petit nombre de couples implique en revanche une estimation qui peut s'avérer peu précise si parmi les couples, il en existe un ou plusieurs aberrants. Il faut donc trouver un compromis sur le nombre de couples de points, ni trop peu, ni pas assez.

Les résultats des comparaisons sont présentés sur le tableau 5.3. Le tableau 5.3 page ci-contre présente pour le filtre de Kalman pondéré (EKFP) et l'estimateur des moindres carrés pondérés (MCP) proposé par Sistiaga, le nombre de couples de points appariés en fonction du nombre d'itérations de l'estimateur pour qu'à la convergence de la PISA, il reste assez de couples de points pour réaliser l'estimation. Le détecteur de points d'intérêt choisi est celui de Harris, la méthode de seuillage est le pourcentage du maximum de la norme du gradient, avec des seuils de 5%. Dans le tableau, * signifie que l'algorithme a échoué.

Le nombre d'itérations de la procédure le plus optimal expérimentalement est 3. En effet, dans le cas de 10 ou 30 itérations, le nombre de paires est insuffisant pour l'estimation et dans le cas du MCP, l'algorithme échoue. Pour toutes les expérimentations mettant en jeu la PISA, les paramètres choisis sont donc le détecteur de Harris pour sélectionner des points, le pourcentage du maximum à 5% – 5% comme méthode de seuillage, le nombre d'itérations pour le filtrage de Kalman pondéré et l'estimateur des moindres carrés pondérés étant fixé à 3.

| Méthode d'estimation ↓ Nombre d'itérations → | 3 | 10 | 30 |
|--|-----|----|----|
| EKFP | 250 | 2 | 2 |
| MCP | 273 | * | * |

TAB. 5.3 – Etude des paramètres de la PISA pour une estimation par EKFP et MCP. Nombre de couples de points utilisés pour l'estimation finale de la transformation, pour les images 4 et 192 de la séquence prise en piscine ("Lucky Luke").

5.7.1.3 Comparaison des différents estimateurs

Dans cette partie, nous comparons les estimateurs proposés par Sistiaga : l'estimateur des moindres carrés (MC), celui des moindres carrés pondérés (MCP), le filtrage de Kalman (EKF), le filtrage de Kalman pondéré (EKFP), et les estimateurs sur modèle linéaire que nous avons proposés dans la partie 5.3.2 : les moindres carrés linéaires (MCL) et les moindres carrés pondérés linéaires (MCPL).

Nous avons retenu le détecteur de Harris pour extraire les points et les mêmes paramètres de seuillage par hystérésis que précédemment. Les seuils de Mahalanobis sont toujours 6.1 pour la distance δ_k^2 entre vecteurs d'attributs et 2.41 pour la distance δ_p^2 entre les vecteurs de position. Pour les estimateurs MC, MCP, EKF et EKFP, le nombre d'itérations a été fixé à 3 pour avoir suffisamment de points pour l'estimation.

Les expériences n'ont pas été menées sur les images de la séquence "Lucky Luke", car la transformation exacte entre les images n'est pas connue. En effet, nous avons accès à la transformation en mètres, programmée sur le robot PISCATOR. Cependant, nous recherchons la transformation en pixels, donc nous avons besoin d'une connaissance précise des paramètres de la caméra et de la translation fond du bassin-caméra. Notre connaissance de ces données n'étant pas assez précise, nous avons fait une autre comparaison des différents estimateurs sur des images d'une scène d'intérieur contenant des objets structurés et dont la transformation d'une image à l'autre est connue. L'image proposée est une image de produits d'entretien (voir figure 5.14). La transformation que nous avons appliquée à l'image originale pour obtenir la deuxième image est une rotation d'angle 1° suivie d'une translation ($t_x = -5$ pixels, $t_y = -12$ pixels).



FIG. 5.14 – Image de produits d'entretien.

Les résultats sont présentés 5.4. Nous pouvons remarquer tout d'abord que la PISA converge moins vite avec les estimateurs que nous avons proposés (5 itérations de la PISA sont nécessaires pour arriver à convergence alors que 2 ou 3 seulement pour MC, MCP, EKF et EKFP). Néanmoins, le nombre d'itérations fixé pour chaque estimateur MC, MCP, EKF, EKFP est 3. Ainsi, les MCL et MCPL nécessitent finalement moins d'itérations (PISA + estimateur) que les estimateurs proposés par Sistiaga, puisque ce sont des méthodes directes (une itération de l'estimateur suffit).

Nous pouvons remarquer que les résultats sur les paramètres de la transformation sont quasiment tous équivalents. Globalement, l'indice de confiance (voir sa définition annexe B.2) pour tous les estimateurs est élevé (entre 63 et 85%). Le zoom est toujours très bien estimé pour tous les estimateurs. La rotation est également très bien estimée par les estimateurs pondérés (moins de 4% d'erreur). Ces résultats peuvent être considérés comme relativement "normaux" car la scène utilisée est une scène structurée et l'image qui a subi la transformation n'est pas bruitée.

En revanche, les translations en x et y ne sont pas très bien estimées : les t_x estimés tournent autour de -2 pixels (pour les estimateurs pondérés) au lieu de -5 pixels (valeur vraie) et t_y autour de -14 pixels au lieu de -12 pixels (valeur vraie). Dans tous les cas, les résultats rendus par les estimateurs pondérés sont plus proches des valeurs vraies, pour un indice de confiance plus élevé. Ainsi, pour la suite, on préférera un estimateur pondéré (EKFP, MCP, MCPL).

| | EKF | MC | MCL | EKFP | MCP | MCPL |
|-------------------------|-------|-------|-------|--------|--------|-------|
| Nb itérations PISA | 2 | 2 | 5 | 5 | 3 | 5 |
| Nb couples de points | 24 | 21 | 6 | 50 | 50 | 8 |
| Indice confiance (%) | 82.7 | 73.7 | 63.6 | 84.9 | 74.2 | 64.9 |
| t_x (pixels) | -0.35 | 2.3 | -1.1 | -1.9 | -1.8 | -2.1 |
| t_y (pixels) | -15.2 | -14.6 | -14.2 | -14.2 | -14.2 | -13.9 |
| angle α (degrés) | 1.4 | 1.5 | 1.2 | 1.03 | 1.02 | 0.96 |
| zoom ρ | 1 | 0.994 | 0.998 | 0.9998 | 0.9997 | 0.998 |

TAB. 5.4 – Etude des différents estimateurs sur les images de produits d'entretien.

Pour terminer, soulignons que les estimateurs que nous avons proposés donnent des résultats comparables aux estimateurs proposés par Sistiaga. L'apport de ces méthodes réside essentiellement en leur simplicité, leur rapidité de convergence et le fait que ce soient des méthodes directes dont les biais et les variances sont calculables.

5.7.2 Recherche des instants de rendez-vous à partir de la séquence d'images

Dans cette partie, on se propose de rechercher le couple d'images les plus proches dans un fenêtrage pour déterminer les instants de rendez-vous. On utilise la séquence d'images "Lucky Luke", de forme "alpha".

On suppose que les instants de rendez-vous initiaux sont donnés par la localisation $A/(C,V)$ (intégration

des caps et vitesses bruités) et une première estimation d'un couple d'instants de rendez-vous. Ces instant grossièrement estimés sont, pour l'expérience considérée, 3 et 191.

On a donc initié les instants à 3 et 191. Les variances de la localisation A/(C,V) peuvent nous permettre de déterminer la largeur du fenêtrage d'images (voir 5.3.3), néanmoins, de par le coût algorithmique assez élevé de la PISA, on a choisi ici de rechercher la paire d'images correspondantes dans un fenêtrage d'une seule image autour des images 3 et 191 (images 2, 3, 4 et 190, 191, 192). Le nombre de points détectés pour respectivement les images 2, 3 et 4 est resp. 448, 393 et 416. Le nombre de points détectés pour les images 190, 191 et 192 est resp. 407, 357 et 384.

La recherche du couple d'images les plus proches a été réalisée par le biais de deux estimateurs différents : les moindres carrés pondérés MCP et les moindres carrés directs pondérés MCPL que nous avons proposés. Les résultats sont présentés dans le tableau 5.5 pour l'estimateur MCP et dans le tableau 5.6 page suivante pour l'estimateur MCPL.

D'après le critère retenu ($\sqrt{t_x^2 + t_y^2}$ minimal), les images correspondantes sont donc les images 4 et 191 pour l'estimateur MCP et les images 3 et 192 pour l'estimateur MCPL. Dans les deux cas, on peut remarquer que les indices de confiance ne sont pas très élevés et ne dépassent pas 20%. On remarque également que les images candidates présentent le zoom le plus proche de la réalité (c'est-à-dire 1).

En réalité, d'après la programmation de la trajectoire "alpha" (voir partie 5.6.4), le couple d'instants de rendez-vous est (4, 192). Il s'avère qu'aucun des deux estimateurs évalués ne renvoie ce résultat. Ils sont de qualité comparable, néanmoins, l'estimateur des moindres carrés pondérés basé sur un modèle linéaire (MCPL) propose une méthode d'estimation directe, ce qui évite les écueils numériques (qui sont engendrés par la descente du gradient du MCP sur modèle non-linéaire) et est également plus rapide. Il sera donc retenu pour réaliser la recherche des instants de rendez-vous grâce à la PISA.

| Images mises en correspondance : | 2/190 | 2/191 | 2/192 | 3/190 | 3/191 | 3/192 | 4/190 | 4/191 | 4/192 |
|----------------------------------|-------|-------|-------|-------|-------|-------|-------|--------------|-------|
| Nombre d'itérations PISA | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Indice de confiance | 18.1 | 13.7 | 14.7 | 15.3 | 14.6 | 14.3 | 13.3 | 15.7 | 15.6 |
| Couples de points apparés | 9 | 179 | 16 | 6 | 273 | 294 | 102 | 11 | 89 |
| t_x | 18.8 | 108 | 67 | 22.4 | 87.6 | 55.6 | 51.1 | 14.8 | 31.3 |
| t_y | -21.2 | 146 | -27 | -21 | 113.3 | 58.3 | 168.7 | 16 | 31.9 |
| ρ | 0.84 | 0.46 | 0.89 | 0.84 | 0.54 | 0.78 | 0.47 | 0.8 | 0.84 |
| α | -1.2 | 5.2 | 2.5 | 2.9 | 4.1 | -0.7 | 0.7 | 0.5 | 0.7 |
| $\sqrt{t_x^2 + t_y^2}$ | 28 | 182 | 72 | 30 | 143 | 81 | 177 | 22 | 45 |

TAB. 5.5 – Recherche du couple d'images correspondantes dans la séquence "Lucky Luke" par MCP.

| Images mises en correspondance : | 2/190 | 2/191 | 2/192 | 3/190 | 3/191 | 3/192 | 4/190 | 4/191 | 4/192 |
|----------------------------------|-------|-------|-------|-------|-------|--------------|-------|-------|-------|
| Nombre d'itérations PISA | 4 | 3 | 3 | 4 | 4 | 3 | 3 | 4 | 3 |
| Indice de confiance | 14 | 13.7 | 14.3 | 13.2 | 14 | 14.6 | 14.3 | 16 | 14.1 |
| Couples de points appariés | 144 | 19 | 73 | 305 | 817 | 24 | 10 | 29 | 335 |
| t_x | 110 | 138 | 104 | 110 | 139 | 50 | 55 | 43 | 103 |
| t_y | 202 | 247 | 105 | 183 | 214 | 89 | 195 | 142 | 179 |
| ρ | 0.32 | 0.2 | 0.59 | 0.36 | 0.31 | 0.75 | 0.44 | 0.55 | 0.44 |
| α | -0.8 | 0.1 | 1.9 | -0.2 | 1.4 | -1.7 | 0.2 | -0.3 | 1.2 |
| $\sqrt{t_x^2 + t_y^2}$ | 230 | 283 | 148 | 214 | 255 | 102 | 203 | 148 | 207 |

TAB. 5.6 – Recherche du couple d'images correspondantes dans la séquence "Lucky Luke" par MCPL.

5.7.3 Conclusion

Les paramètres de la PISA ont été déterminés au mieux. Pour la phase d'extraction des points d'intérêt, le détecteur de Harris a été retenu. Pour sélectionner ces points dont certains peuvent être aberrants, la méthode de seuillage par hystérésis par pourcentage du maximum des normes du gradient a été retenue. En ce qui concerne les estimateurs intégrés à la PISA, les expériences ont montré que les estimateurs pondérés ont de meilleurs résultats que les estimateurs non-pondérés. Les estimateurs que nous avons proposés (MCL, MCPL) améliorent la rapidité de la PISA, pour des estimations de qualité équivalente aux autres estimateurs (MCP, EKFP). La recherche d'instant de rendez-vous est très sensible à la qualité de l'estimation.

5.8 Résultats relatifs à la construction de mosaïques 2D

Dans cette partie, nous allons exposer les résultats relatifs à l'atténuation d'effets qui peuvent perturber la construction de mosaïques (influence du halo lumineux et influence des bords de l'image qui est apparue lors des mises en oeuvre expérimentales). Puis une comparaison des deux algorithmes présentés en détail, le KLT et le RMR, sera effectuée. Enfin, quelques résultats sur l'étude d'une localisation AR/I seront exposés. Remarque : pour toutes les comparaisons qui suivent, les rapports entre les tailles des mosaïques sont respectés.

5.8.1 Influence de la correction d'intensité lumineuse sur la génération et le rendu de mosaïques

Deux méthodes ont été présentées pour corriger la présence d'un halo dans les images : la correction radiométrique et le filtrage homomorphique. Afin de valider ces méthodes, nous proposons deux critères. Le premier critère est subjectif : il consiste à vérifier que l'image corrigée est de meilleure qualité et que le halo est visiblement réduit.

Le deuxième critère est un critère plus objectif : il consiste à créer une séquence "augmentée". Cette nouvelle séquence est construite à partir d'un collage de la séquence d'images initiale (images de \mathbf{I}_1 à \mathbf{I}_N) et de cette même séquence symétrique (images de \mathbf{I}_N à \mathbf{I}_1). C'est donc la séquence d'images $[\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_{N-1}, \mathbf{I}_N, \mathbf{I}_{N-1}, \dots, \mathbf{I}_2, \mathbf{I}_1]$, comme l'illustre la figure 5.15 page suivante. L'image finale de la séquence "augmentée" est donc identique à son image initiale. Ainsi donc, Si l'estimation des déplacements inter-image et les conditions d'éclairage étaient parfaits, la mosaïque d'images créée à partir de la séquence augmentée reviendrait sur elle-même. Le critère est donc le suivant : si la mosaïque construite à partir de la séquence "augmentée" avec les images corrigées revient sur elle-même de façon plus précise que la mosaïque construite à partir de la séquence "augmentée" avec les images non-corrigées, alors la mosaïque construite à partir des images corrigées est de meilleure qualité et la méthode est retenue. Quelques résultats ainsi que leurs performances au niveau de ces critères, seront présentés dans la partie 5.8.1.

Dans cette partie, nous présentons les résultats associés à ces méthodes et leurs performances sur des images seules et pour certaines sur des mosaïques.

Les images utilisées sont des images acquises lors de missions en mer. En effet, les séquences en bassin ont été acquises avec un éclairage relativement diffus (8 spots lumineux répartis 2 par 2 sur les 4 murs du bassin) et non par un spot lumineux éclairant la zone imagée, comme en conditions réelles. D'autre part, le fond du bassin étant granuleux, les images prises avec le drap ou de référence (sur fond blanc) n'auraient pas eu la même texture (lisse et réfléchissant la lumière pour le drap, granuleux et peu réfléchissant pour l'image de référence prise sans rien au fond). C'est pour ces raisons que les expériences sur la correction d'intensité lumineuse sur les images sous-marines n'ont pas pu être effectuées sur les séquences acquises en bassin mais avec des photos.

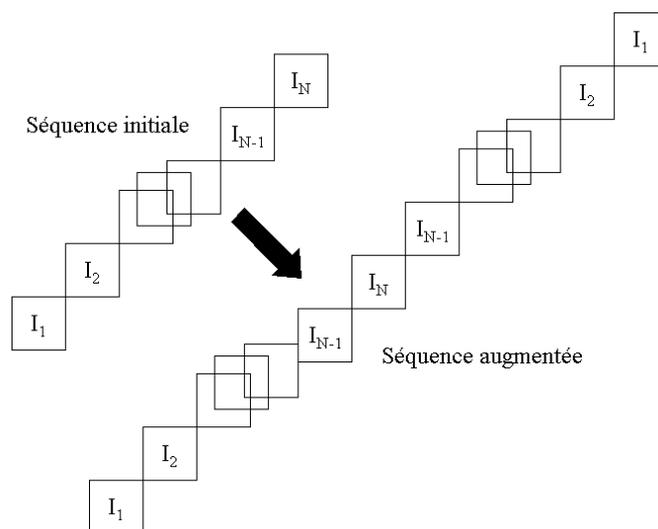


FIG. 5.15 – Principe de la séquence augmentée.

5.8.1.1 Correction radiométrique automatique

Nous présentons sur les figures 5.16(a), 5.16(b) et 5.16(c) page ci-contre le résultat de la méthode employant une image acquise lors de la descente de l'engin. Ces images représentent une épave d'avion photographiée à 1200 mètres de profondeur, en Mer Méditerranée. Nous pouvons remarquer que certains détails non-visibles dans l'image originale le sont dans l'image résultat. L'effet de halo est visiblement atténué pour le traitement proposé.

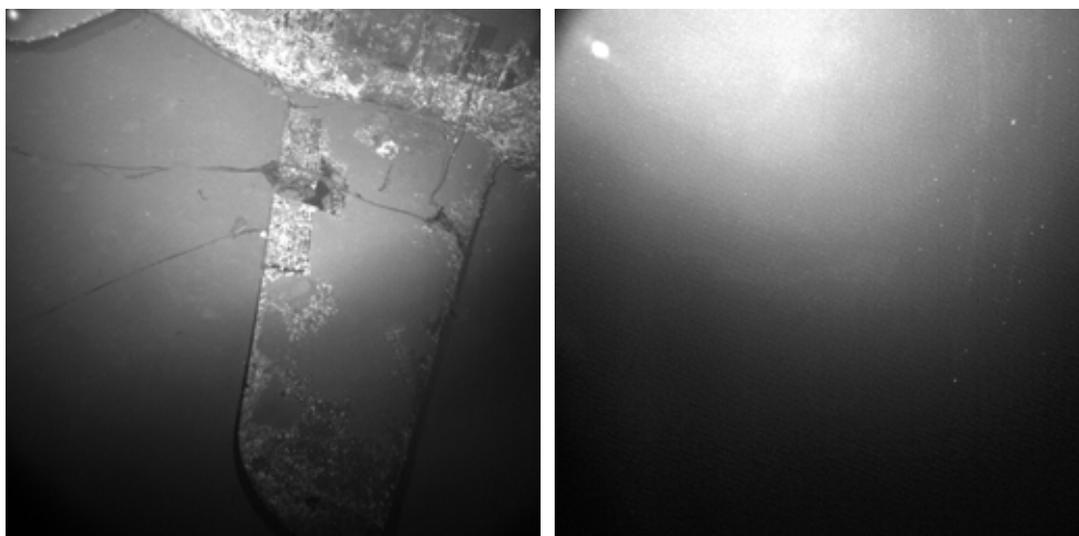
Actuellement, nous n'avons pas à notre disposition de séquence d'images en conditions réelles comportant une image prise en pleine eau. Nous n'avons donc pu tester la méthode que sur des photos, et n'avons pas pu reconstituer de mosaïque pour valider notre méthode.

5.8.1.2 Filtrage homomorphique

Les figures 5.17(a) à 5.17(d) page 112) présentent une séquence d'images représentant des amphores, acquises en Mer Méditerranée par 300 mètres de fond. Cette séquence est utilisée pour valider l'utilisation et tester l'efficacité de la méthode de filtrage homomorphique.

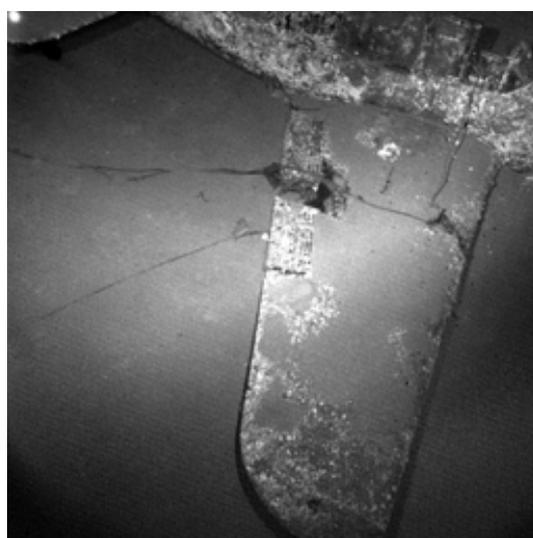
Nous pouvons constater que l'effet de halo est moins présent sur l'image traitée. D'autre part, la construction de la mosaïque n'est pas altérée par le traitement et le deuxième critère défini dans la partie 5.1.3 montre que la mosaïque "augmentée" avec les images corrigées revient au même endroit que la mosaïque "augmentée" avec les images non-corrigées. Cette méthode est prometteuse, cependant, elle requiert l'intervention d'un opérateur pour régler les paramètres du filtre fréquentiel.

Les deux méthodes présentées ici sont en phase d'intégration dans le logiciel MATISSE® pour être utilisées opérationnellement.



(a) image initiale

(b) image de référence (acquise en pleine eau)

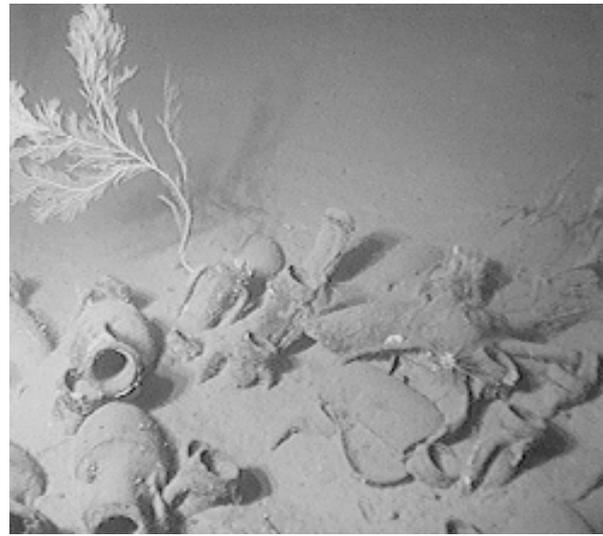


(c) image corrigée

FIG. 5.16 – Correction radiométrique automatique avec image de référence acquise en pleine eau.



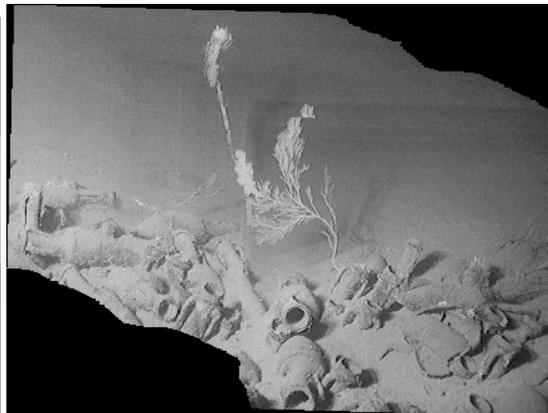
(a) image initiale



(b) image traitée



(c) mosaïque initiale



(d) mosaïque construite à partir des images traitées

FIG. 5.17 – Filtrage homomorphique.

5.8.2 Influence du fenêtrage de l'image lors de l'estimation des déplacements de la mosaïque

Le logiciel MATISSE® (voir annexe C) donne la possibilité à l'utilisateur de ne considérer qu'une partie de l'image pour l'estimation du mouvement. Les problèmes liés à l'éclairage (notamment un halo lumineux) apparaissent à partir d'une profondeur supérieure à 5m (voir partie 5.1.2). Le halo lumineux est généralement centré donc si les bords sont supprimés, on peut supposer que l'intensité des niveaux de gris sera plus homogène dans l'image. D'autre part, lors de l'acquisition d'images, des phénomènes de distorsion existent. Les bords de l'image ne subissent pas la même transformation lors du passage 3D→2D que le centre de l'image.

Nous avons remarqué tous ces phénomènes dans la phase d'expérimentation et avons donc étudié l'influence des bords sur la construction de la mosaïque par les deux méthodes RMR et KLT.

La séquence d'images a été acquise grâce au portique PISCATOR, dans le bassin d'essai intérieur d'Ifrermer Toulon, en allumant les 8 spots lumineux du bassin. La trajectoire programmée pour le robot est une trajectoire à une boucle, appelée "alpha" dans le chapitre 3.

5.8.2.1 Résultats obtenus pour la méthode RMR

La figure 5.18(a) page 115 représente la mosaïque construite par la méthode du RMR, avec l'estimation de la translation seule (sans estimation de la rotation). La figure 5.18(b) page 115 représente la mosaïque construite avec le RMR, après avoir retiré des bords de 100 pixels de chaque côté de l'image. Nous pouvons remarquer d'un point de vue strictement visuel que la mosaïque construite en supprimant les bords est plus continue, que la trajectoire semble mieux estimée puisque la mosaïque revient mieux au-dessus du point de rendez-vous.

Sur la figure 5.20 page 117, sont montrées la trajectoire programmée pour le robot⁶ ainsi que les trajectoires reconstruites à partir des estimations du RMR, selon que l'on a supprimé les bords ou non lors de l'estimation.

Nous pouvons remarquer que la mosaïque créée en supprimant les bords se rapproche plus de la trajectoire réellement suivie par le bras du PISCATOR. Ceci est visible notamment dans la dernière partie de la trajectoire, où l'on pouvait remarquer que la trajectoire estimée sans suppression des bords était rectiligne alors que la trajectoire estimée avec suppression des bords suit relativement bien la trajectoire du robot.

Ainsi, les bords jouent un rôle très important dans la construction de la mosaïque. Cela est dû à l'éclairage non-uniforme dans le bassin, mais aussi à la distorsion des ondes lumineuses au niveau de la lentille d'entrée de la caméra. Il serait intéressant de prendre en compte ce phénomène de distorsion radiale, notamment dans la modélisation de la caméra, puisque dans le chapitre 6, nous travaillons avec une caméra calibrée.

⁶Cette trajectoire est déterminée à partir d'une trajectoire robot en mètres, exprimée en pixels par une transformation 3D→2D : voir partie 6.2.

5.8.2.2 Résultats obtenus pour la méthode KLT

Une étude a été menée pour choisir les meilleurs paramètres intervenant dans la méthode KLT : la taille de la fenêtre et le nombre de points sélectionnés dans chaque fenêtre. Des tests ont été réalisés en faisant varier le fenêtrage de taille 7, 9 ou 11 pixels et le nombre de points sélectionnés dans chaque fenêtre : 10, 20, 30, 50 ou 60. Les résultats optimaux, en terme de proximité avec la trajectoire programmée et sur des critères visuels, ont été trouvés pour une fenêtre de 9 pixels et 50 points, pour la séquence "alpha" ainsi que pour la trajectoire à deux boucles que nous utiliserons un peu plus loin. Dans l'état actuel des choses, ces paramètres sont déterminés expérimentalement pour chaque séquence.

La figure 5.19(a) page 116 représente la mosaïque construite par la méthode du KLT, avec une sélection de 50 points et un fenêtrage pour l'estimation de 9 pixels, sans estimation de la rotation. La figure 5.19(b) page 116 représente la mosaïque construite avec le KLT, en ayant retiré 100 pixels de chaque côté de l'image. Comme dans le cas du RMR, nous pouvons constater une amélioration visuelle de la mosaïque. Elle est plus continue et moins déformée.

Les figures 5.21 page 117 montrent la trajectoire réellement suivie par le bras du PISCATOR ainsi que les trajectoires reconstruites à partir des estimations du KLT, selon que l'on a supprimé les bords ou non lors de l'estimation. Comme pour le RMR, la mosaïque créée en supprimant les bords se rapproche également plus de la trajectoire réelle.

La mosaïque construite étant d'assez mauvaise qualité, nous pouvons également conclure que les bords (et donc les effets de halo et de distorsion) jouent une grande influence dans l'estimation du mouvement.

5.8.2.3 Conclusion

Les expériences montrent bien l'influence des bords des images sur l'estimation du mouvement. En effet, la distorsion géométrique due à la lentille ainsi que l'éclairage lumineux non-uniforme induisent des effets de bords qui nuisent à l'estimation du mouvement, et donc pour la suite, à une estimation de la position de l'engin à partir de ces estimations de mouvement image (localisation AR/I).



(a) Mosaïque obtenue sans suppression des bords



(b) Mosaïque obtenue en supprimant les bords

FIG. 5.18 – Méthode RMR : influence du fenêtrage des images sur la construction de mosaïque.



(a) Mosaïque obtenue sans suppression des bords



(b) Mosaïque obtenue en supprimant les bords

FIG. 5.19 – Méthode KLT : influence du fenêtrage des images sur la construction de mosaïque.

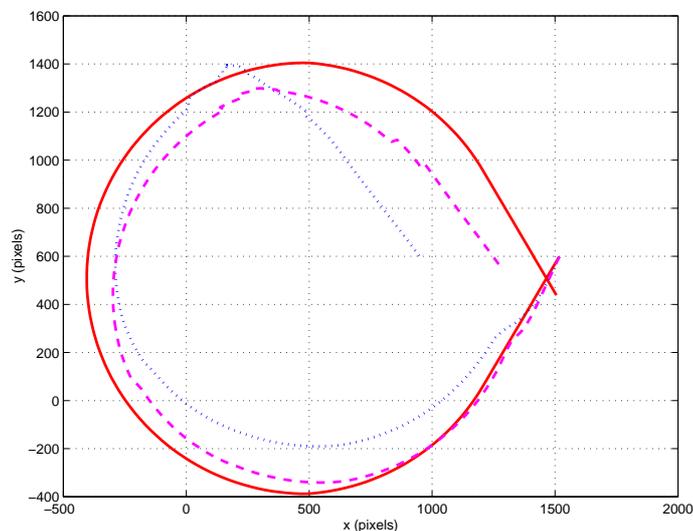


FIG. 5.20 – Influence du fenêtrage des images sur la construction de mosaïque RMR, au niveau de la trajectoire. En trait plein, la trajectoire programmée pour le robot, en pointillés ‘.’ la trajectoire obtenue par la mosaïque avant le retrait des bords des images, en trait pointillés ‘-’ la trajectoire obtenue après le retrait des bords des images.

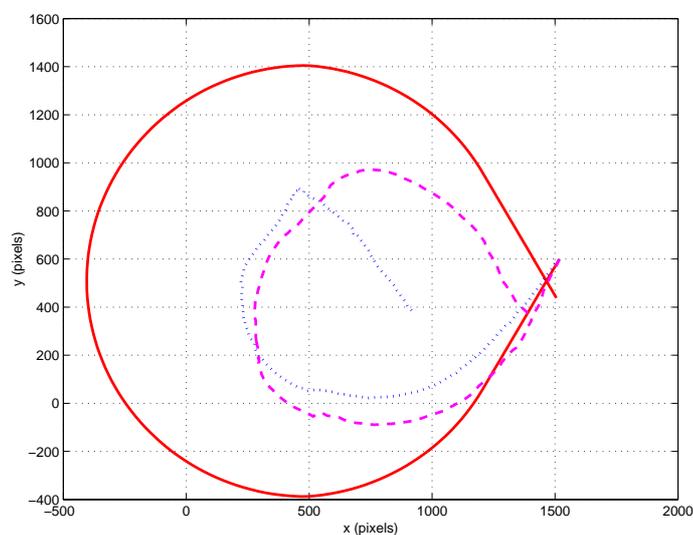


FIG. 5.21 – Influence du fenêtrage des images sur la construction de mosaïque KLT, au niveau de la trajectoire. En trait plein, la trajectoire programmée pour le robot, en pointillés ‘.’ la trajectoire obtenue par la mosaïque avant le retrait des bords des images, en trait pointillés ‘-’ la trajectoire obtenue après le retrait des bords des images.

5.8.3 Comparaison des algorithmes de construction de mosaïque : KLT et RMR

Le RMR est une méthode multi-résolution robuste alors que le KLT est une méthode qui repose sur un suivi de points, donc sur une connaissance a priori de la localisation des points d'une image à l'image suivante. Ainsi, lors d'un mouvement un peu brusque, le KLT risque de "décrocher" car il présuppose un mouvement faible. Les résultats présentés dans ce qui suit ont été réalisés à partir de séquences prises en bassin intérieur, grâce au portique PISCATOR auquel a été fixée une caméra. Deux types de trajectoires ont été programmées. L'une est la trajectoire "alpha", décrite dans le chapitre 3 et comportant un instant de rendez-vous, l'autre est une trajectoire comportant deux instants de rendez-vous et deux boucles. Pour le RMR comme pour le KLT, seules les translations ont été estimées. Pour le KLT, une fenêtre de 9 pixels a été choisie, pour une sélection de 50 points dans cette fenêtre.

5.8.3.1 Comparaison sur la mosaïque construite

Sur les 2 figures 5.22 (trajectoire "alpha") et 5.23 page 120 (trajectoire à deux boucles) sont représentées les mosaïques construites par resp. la méthode RMR et par la méthode KLT, à partir de la séquence "Lucky Luke" acquise en bassin intérieur. Nous pouvons remarquer que la mosaïque construite par le RMR est visuellement plus belle que celle construite par le KLT, qui est plutôt déformée. D'autre part, pour les deux types de trajectoires, on peut remarquer que la mosaïque RMR revient plus près des points de recoupement que la mosaïque KLT. D'un point de vue purement visuel, on peut donc constater que la méthode RMR est plus efficace que la méthode KLT.

5.8.3.2 Comparaison sur les déplacements estimés

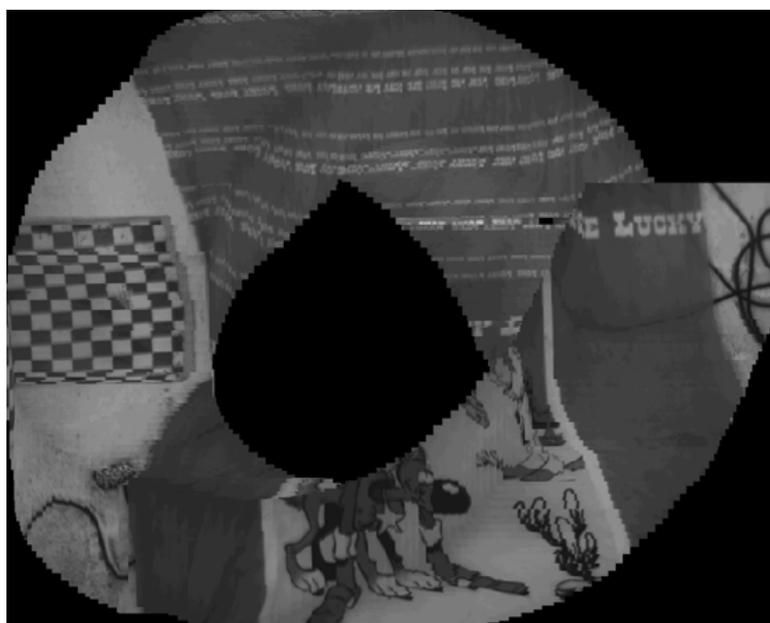
Après avoir comparé les deux méthodes visuellement, nous nous proposons de comparer les déplacements estimés Δ_x et Δ_y . Pour cela, nous présentons sur les figures 5.24 page 121, en trait plein les déplacements déterminés par la trajectoire que nous avons programmée sur le robot (la trajectoire avait été programmée en mètres mais elle a été transformée en pixels, voir partie 6.2), les déplacements estimés par le RMR en "+", les déplacements estimés par le KLT en "×". Sur les figures 5.24(a) et 5.24(b) (trajectoire "alpha") et 5.25(a) et 5.25(b) page 122 (trajectoire à deux boucles), nous pouvons constater que le RMR "suit" bien les déplacements programmés pour le robot, et ses déplacements sont continus en général. En revanche, le KLT ne montre pas de continuité dans l'estimation des déplacements, qui sont de plus relativement loin des déplacements programmés.

D'autre part, le KLT semble "saturer" pour des déplacements supérieurs à +15 pixels ou inférieurs à -25 pixels. Ceci peut s'expliquer car le KLT est une méthode de suivi de points, qui s'effectue à partir d'une fenêtre qui se déplace d'un certain nombre de pixels d'une image à l'autre. En quelque sorte, le mouvement est contraint par le déplacement de cette fenêtre. Les paramètres choisis pourtant de façon optimale pour ces séquences semblent donc mal adaptés aux séquences...

Sur les figure 5.24(c) (trajectoire "alpha") et 5.25(c) page 122 (trajectoire à deux boucles), nous pouvons constater que les déplacements $\sqrt{\Delta_x^2 + \Delta_y^2}$ obtenus par le RMR sont assez proches des déplacements pro-



(a) Mosaïque obtenue par le RMR



(b) Mosaïque obtenue par le KLT

FIG. 5.22 – Comparaison des mosaïques construites à partir du RMR et du KLT pour la trajectoire "alpha".

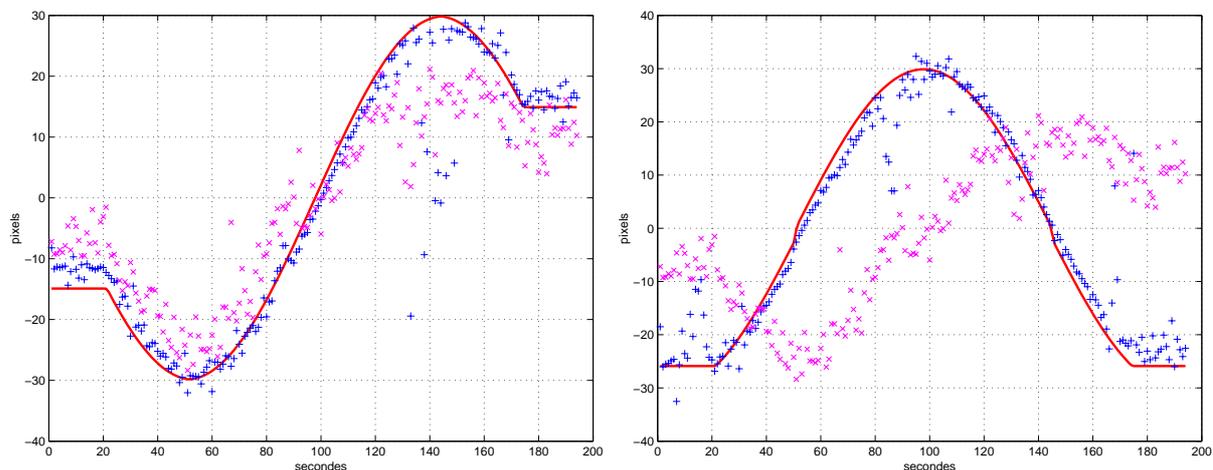


(a) Mosaïque obtenue par le RMR



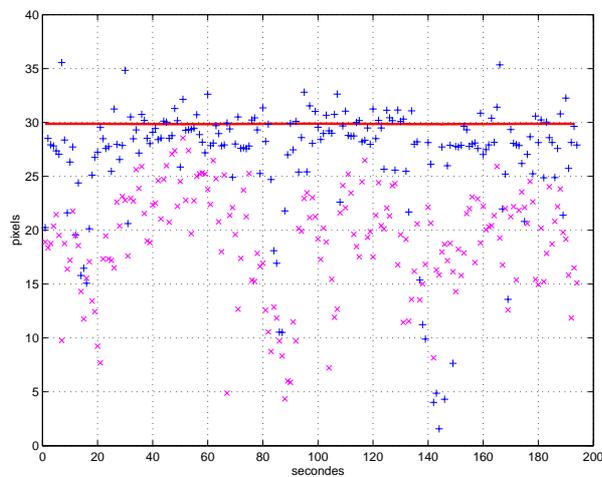
(b) Mosaïque obtenue par le KLT

FIG. 5.23 – Comparaison des mosaïques construites à partir du RMR et du KLT pour une trajectoire à deux boucles.



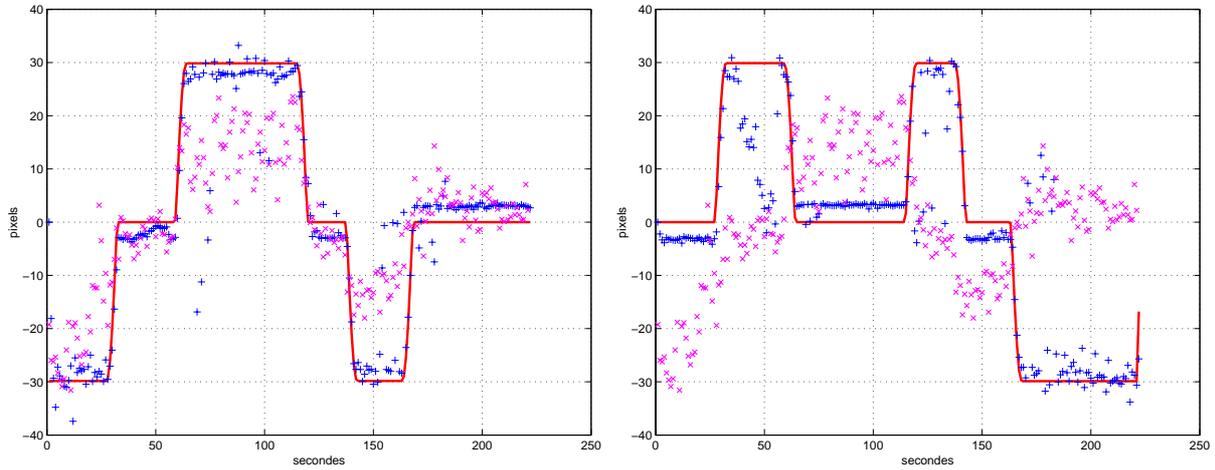
(a) Comparaisons des déplacements en x obtenus par le RMR et le KLT

(b) Comparaisons des déplacements en y obtenus par le RMR et le KLT



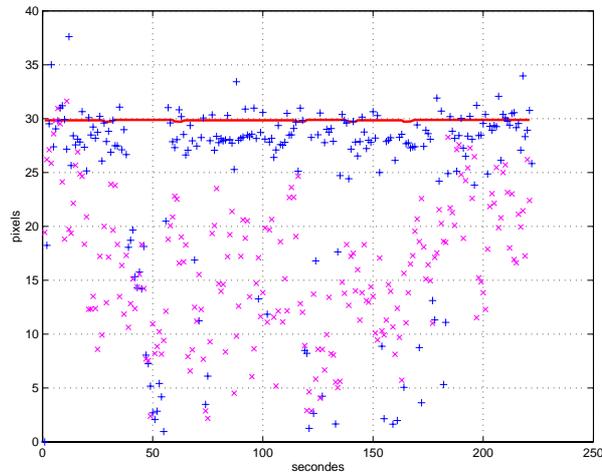
(c) Comparaisons des déplacements $\sqrt{\Delta_x^2 + \Delta_y^2}$ obtenus par le RMR et le KLT

FIG. 5.24 – Comparaison des déplacements obtenus par les méthodes RMR (+) et KLT (×), pour la trajectoire "alpha".



(a) Comparaisons des déplacements en x obtenus par le RMR et le KLT

(b) Comparaisons des déplacements en y obtenus par le RMR et le KLT



(c) Comparaisons des déplacements $\sqrt{\Delta_x^2 + \Delta_y^2}$ obtenus par le RMR et le KLT

FIG. 5.25 – Comparaison des déplacements obtenus par les méthodes RMR (+) et KLT (x), pour la trajectoire à deux boucles.

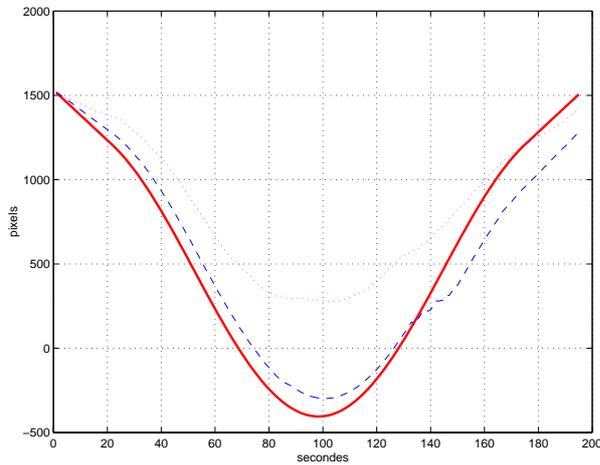
grammés (à ± 5 pixels près), contrairement aux déplacements KLT, qui restent assez bas (cela est dû à la mauvaise estimation des Δ_x et Δ_y). Cependant, le RMR présente quelques points aberrants.

5.8.3.3 Comparaison sur les trajectoires reconstruites

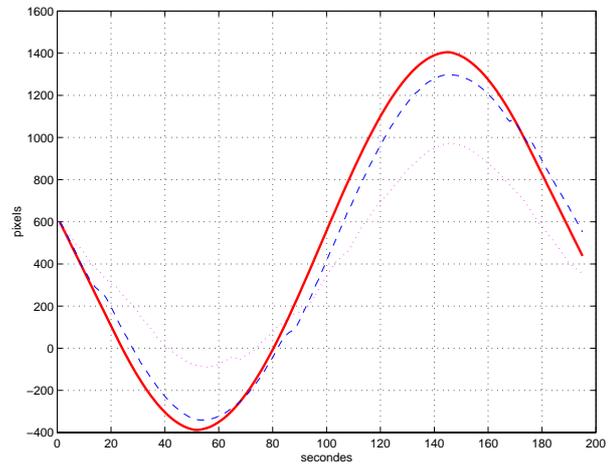
Nous comparons maintenant les trajectoires reconstruites à partir des déplacements estimés (localisations A/I). Nous remarquons sur les figures 5.26(a), 5.27(a), 5.26(b) et 5.27(b) page 125 que les trajectoires en x et y reconstruites à partir des déplacements estimés par le RMR sont relativement proches de la trajectoire programmée. En revanche, ce n'est pas le cas pour le KLT. On remarque d'autre part que malgré une bonne estimation des déplacements du RMR, une dérive apparaît après intégration des déplacements. Ainsi, une correction est bien nécessaire pour améliorer la construction de la mosaïque.

5.8.3.4 Conclusion

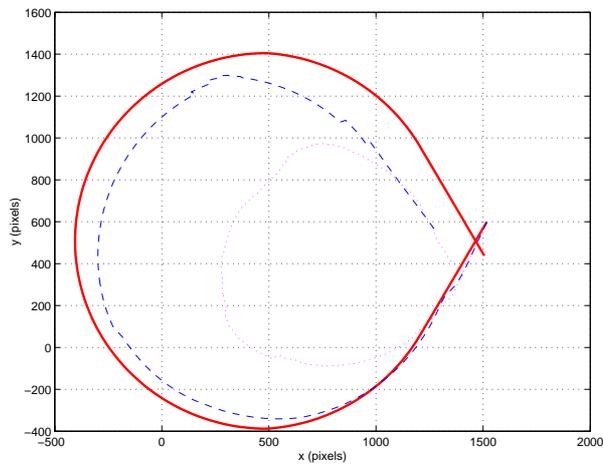
Aussi bien sur la trajectoire "alpha" que sur la trajectoire à deux boucles, la méthode RMR offre une bonne estimation des déplacements, contrairement à la méthode KLT dont les paramètres doivent être réglés selon la séquence. Ceci peut s'expliquer par la robustesse de l'estimation du RMR ainsi que par l'approche multi-résolution qui améliore encore l'estimation. D'un point de vue coût algorithmique, le RMR est plus rapide que le KLT -qui n'a pas encore été optimisé pour l'instant à l'Ifremer- bien qu'il s'agisse d'un algorithme multi-résolution. Ainsi, la méthode RMR est retenue pour les développements du chapitre suivant, visant à réduire la dérive de la mosaïque.



(a) Comparaisons des trajectoires en x obtenues par le RMR et le KLT

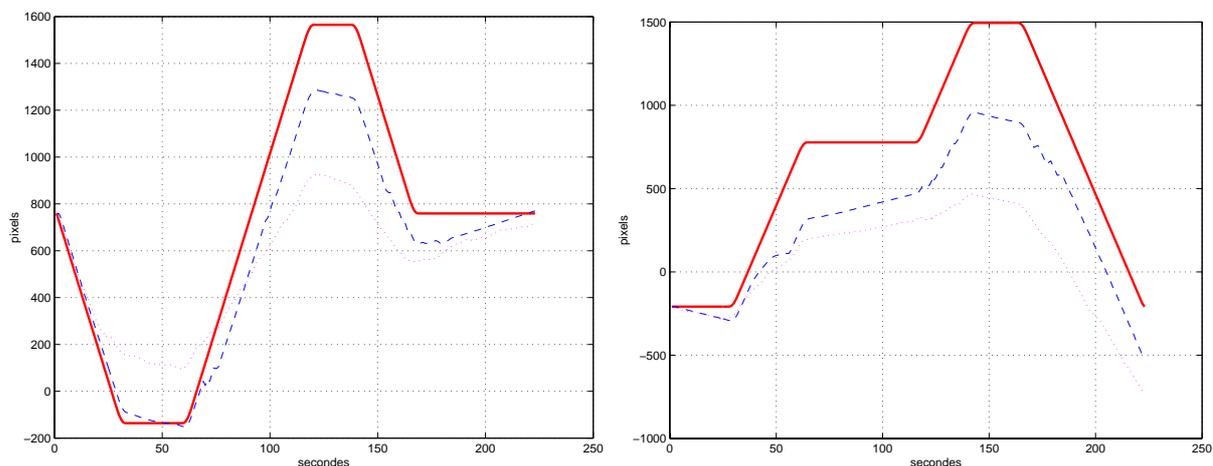


(b) Comparaisons des trajectoires en y obtenues par le RMR et le KLT



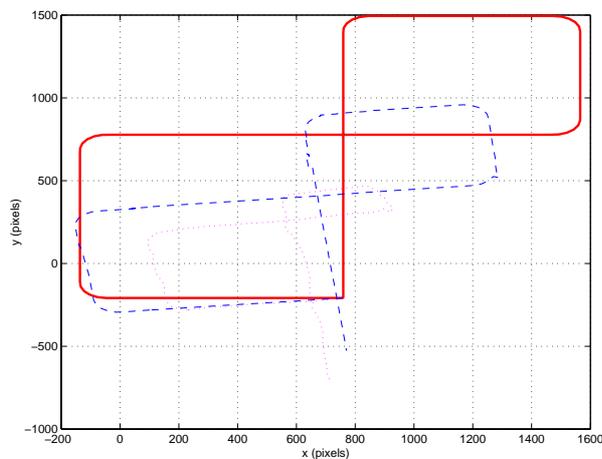
(c) Comparaisons des trajectoires obtenues par le RMR et le KLT

FIG. 5.26 – Comparaison des trajectoires obtenues par les méthodes RMR (en pointillés '-') et KLT (en pointillés '..'), pour la trajectoire "alpha".



(a) Comparaisons des trajectoires en x obtenues par le RMR et le KLT

(b) Comparaisons des trajectoires en y obtenues par le RMR et le KLT



(c) Comparaisons des trajectoires obtenues par le RMR et le KLT

FIG. 5.27 – Comparaison des trajectoires obtenues par les méthodes RMR (en pointillés '-') et KLT (en pointillés '..'), pour la trajectoire à deux boucles.

5.8.4 Réalisation de la localisation AR/I

Sur les figures 5.28(a) et 5.28(b) sont présentés les résultats de la fusion aller-retour du RMR, pour la trajectoire "alpha" comportant un instant de rendez-vous.

L'estimation est réalisée grâce à un estimateur des moindres carrés pondérés, en considérant une localisation A/I (intégration des déplacements estimés par le RMR) et une localisation R/I (intégration rétrograde des déplacements). Pour ce faire, on a supposé que la variance des localisations A/I et R/I augmentait linéairement avec le temps (voir partie 5.5).

En trait plein est représentée la trajectoire programmée pour le bras manipulateur, en pointillés "-" est présentée la trajectoire obtenue par les estimations de mouvement du RMR et en pointillés "..." est présentée la trajectoire correspondant à la localisation AR/I (réalisée à partir des déplacements estimés par le RMR).

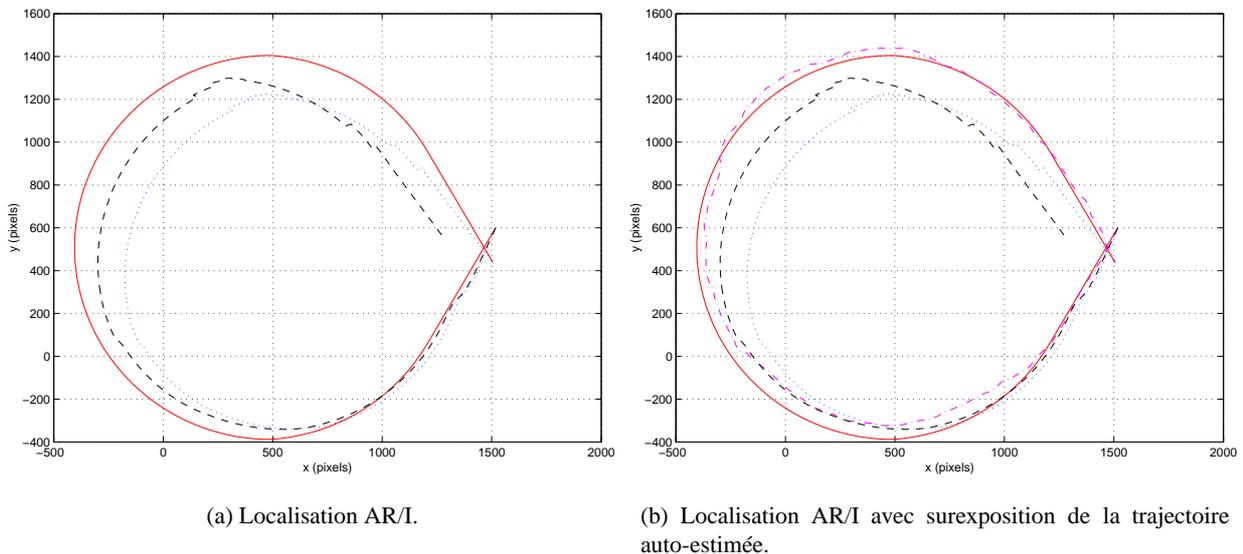


FIG. 5.28 – Localisation A/I (RMR), localisation AR/I, et trajectoire auto-estimée pour la trajectoire alpha.

Les résultats montrent que l'estimation aller-retour à partir du RMR, en tenant compte de l'instant de rendez-vous, ne permet pas de se rapprocher de la trajectoire vraie avec l'hypothèse d'une variance fixe tout au long de la trajectoire. Sous réserve que les biais et variances de l'estimateur RMR soient connus, une fusion "correcte" des localisation A/I et R/I est néanmoins possible.

5.9 Conclusion

Ce chapitre a présenté dans un premier temps les problèmes liés à l'acquisition d'images en milieu sous-marin. Puis une méthode d'estimation des instants de rendez-vous est proposée, s'appuyant sur la PISA développée par Sistiaga. Deux nouvelles méthodes ont été intégrées à la PISA (MCL, MCPL). Des premiers résultats sont proposés à ce sujet.

Puis deux algorithmes d'estimation du mouvement sont étudiés et la méthode RMR est retenue.

Enfin, la réalisation de la localisation AR/I est étudiée et des résultats proposés. Ces résultats montrent que cette fusion est facilement réalisable, sous réserve que les biais et variances des paramètres estimés par le RMR soient disponibles.

Chapitre 6

Faisabilité de la fusion entre localisation AR/I et localisation AR/(C,V)

La problématique de ces recherches consiste à améliorer la construction de mosaïque d'images en introduisant des données extérieures afin d'en améliorer la précision et la localisation.

Le véhicule sous-marin se déplace à 3 mètres de fond marin, sur une trajectoire qu'on suppose horizontale. Les capteurs qui sont embarqués sont une caméra verticale (elle regarde en-dessous de l'engin), un loch-doppler et un gyroscope (voir explications partie 2.2.1). Le but de ces recherches est d'améliorer la construction et la localisation de la mosaïque d'images, qui est jusqu'à présent effectuée à partir des images seules. Elle est sujette à une dérive, quelle que soit la méthode employée à Ifremer (KLT ou RMR, voir parties 5.4.2 et 5.4.3). L'estimation optimale de la trajectoire du porteur de la caméra par des données indépendantes devrait permettre de corriger certains aspects de la mosaïque. Dans ce but, plusieurs estimateurs ont été proposés pour réaliser la localisation AR/(C,V) et évalués dans les chapitres 3 et 4. On souhaiterait également tenir compte de la trajectoire particulière du véhicule lors de l'exploration d'une zone d'intérêt : il s'agit d'un quadrillage qui présente plusieurs recouvrements. Les instants où l'engin repasse par le même lieu géométrique (instants de rendez-vous) peuvent être déterminés. La méthode proposée afin d'estimer ces instants de rendez-vous a été décrite dans le chapitre 5.

Nous avons également recherché une estimation optimale de la trajectoire de l'engin (localisation AR/I) à partir des déplacements estimés par l'image seule (voir partie 5.5).

Nous étudions dans ce chapitre la faisabilité de la fusion entre la localisation AR/(C,V) et la localisation AR/I. Après avoir recherché un référentiel commun à la localisation AR/I et à la localisation AR/(C,V), nous exposerons le système que nous proposons et terminerons par des expérimentations.

6.1 Fusion de données

La fusion de données permet de "combiner des informations provenant de plusieurs sources afin d'améliorer la prise de décision" (I. Bloch). Elle intervient dans nombre de domaines, notamment dans le domaine de l'imagerie médicale pour l'observation du corps et de ses pathologies, ou pour l'aide au geste et au diag-

nostic médical. On peut citer par exemple des applications comme la construction d'atlas anatomiques et fonctionnels 3D à partir d'images médicales 2D, la chirurgie guidée par l'image (intégration des données virtuelles pré-opératoires et des données réelles (vue directe du patient en salle d'opération)), etc... [Aya03] [SAN⁺04].

D'autres applications dans le domaine militaire (détection, identification et suivi de cibles, surveillance de champs de bataille) ou spatial (imagerie satellitaire et commande d'engins spatiaux) utilisent également la fusion de données.

En imagerie sous-marine, comme décrit dans la partie 5.1, il existe plusieurs types d'images. En ce qui concerne les images de profondeur, des cartes bathymétriques sont construites et géo-référencées, c'est-à-dire recalées par rapport à des coordonnées absolues. Dans ce but, Ifremer a créé le logiciel REGINA (REcalage Graphique Interactif de la NAvigation) [Edy02] qui permet de créer des cartes bathymétriques géo-référencées.

Dans le cadre de notre problématique, la fusion concerne la trajectoire issue des données de cap et de vitesse et de la mosaïque (et incidemment de la localisation AR/I).

6.2 Référentiel commun aux localisation AR/(C,V) et AR/I

Afin de pouvoir fusionner des données de trajectoire capteur, exprimée en mètres ou image, exprimée en pixels, il est nécessaire de trouver un référentiel commun. Il est donc nécessaire de connaître la relation qu'il existe entre la dimension d'un objet dans la scène 3D observée (en mètre) et la dimension de ce même objet dans l'image (en pixel).

Dans cette partie, nous allons tout d'abord décrire le modèle de caméra utilisé, puis nous expliciterons les transformations à appliquer pour passer du repère monde au repère caméra. Enfin, nous donnerons quelques méthodes qui permettent de calibrer des caméras.

6.2.1 Modélisation d'une caméra

Dans la littérature, on peut trouver plusieurs modèles de caméra : la caméra affine, orthographique, projective ou perspective. Le modèle de caméra perspective est le plus répandu, c'est le modèle "trou d'épingle" ou sténopé (voir figure 6.1 page ci-contre).

6.2.2 Changement de repère

Plusieurs repères sont nécessaires pour expliquer la transformation géométrique pour passer d'un point 3D de l'espace à un point 2D de l'image (voir figure 6.2 page suivante).

Le repère scène est un repère tridimensionnel où un point P est représenté par ses coordonnées $P = (X, Y, Z)^T$. Le repère caméra est un repère tridimensionnel lié à la caméra, dans lequel un point P est représenté par ses coordonnées $P = (X_C, Y_C, Z_C)^T$, l'axe (Z_C) étant orthogonal au plan image et

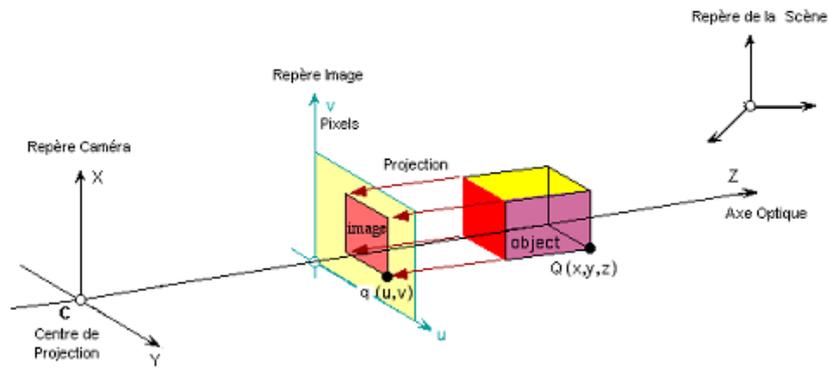


FIG. 6.1 – Modèle de caméra sténopé.

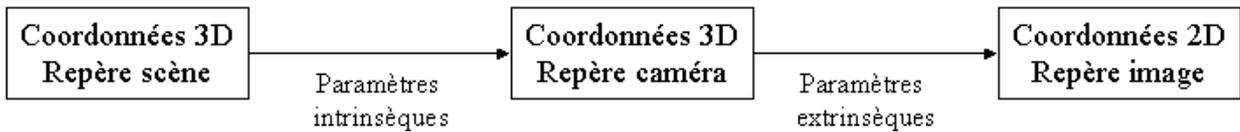


FIG. 6.2 – Changement de repère.

définissant l’axe optique. Le dernier repère est le repère image. C’est un repère bidimensionnel dont l’origine est placée dans le coin supérieur gauche de l’image et dont les axes sont les mêmes que ceux du repère caméra.

La projection perspective peut maintenant être décomposée en une suite de transformations entre les repères définis ci-dessus :

- \mathbb{T}_R est la transformation rigide entre le repère scène et le repère caméra :

$$\mathbb{T}_R = \begin{pmatrix} R^{3 \times 3} & t^{3 \times 1} \\ 0 & 1 \end{pmatrix} \tag{6.1}$$

où R est une matrice de rotation composée des rotations autour des 3 axes et t est le vecteur 3×1 qui regroupe les 3 translations autour des axes. R et t sont les paramètres extrinsèques.

- \mathbb{T}_P est la transformation perspective, un point Q de coordonnées homogènes $(X, Y, Z, 1)^T$ se projette en un point q de coordonnées homogènes $(x, y, 1)^T$ avec :

$$\mathbb{T}_P = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{6.2}$$

où f est la focale de la caméra.

– \mathbb{T}_A est la transformation affine reliant les repères caméra et image :

$$\mathbb{T}_A = \begin{pmatrix} k_u & -k_u \cot \theta & u_0 \\ 0 & \frac{k_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.3)$$

où k_u et k_v sont les facteurs d'échelle suivant les axes u et v (exprimés en pixel/mm), u_0 et v_0 sont les coordonnées du point principal et θ est l'angle entre les deux axes du repère image (cas de pixels non-rectangulaires).

La transformation $\mathbb{P} = \mathbb{T}_A \mathbb{T}_P \mathbb{T}_R$ permet alors de passer du repère scène au repère image. Elle peut éga-

lement s'écrire $\mathbb{K}R(I_3 | -t)$ en posant $\mathbb{T}_A \mathbb{T}_P = \begin{pmatrix} k_u f & -k_u f \cot \theta & u_0 & 0 \\ 0 & \frac{k_v f}{\sin \theta} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = (\mathbb{K}_{3 \times 3} \quad 0_3)$. \mathbb{K} s'appelle la matrice des paramètres intrinsèques. En posant $\alpha_u = k_u f$ et $\alpha_v = k_v f$, on a

$$\mathbb{K} = \begin{pmatrix} \alpha_u & -\alpha_u \cot \theta & u_0 \\ 0 & \frac{\alpha_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.4)$$

Si les pixels sont rectangulaires (ce qui est une approximation très acceptable), $\theta = 90^\circ$ et qu'on pose $\frac{k_u}{k_v} = 1$, on a :

$$\mathbb{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.5)$$

Pour une caméra immobile qui se déplace à altitude constante t_Z , on note \mathbb{P}_{t_Z} la transformation 3D→2D suivante :

$$\mathbb{P}_{t_Z} = \begin{bmatrix} \alpha_u & 0 & u_0 & u_0 t_Z \\ 0 & \alpha_v & v_0 & v_0 t_Z \\ 0 & 0 & 1 & t_Z \end{bmatrix} \quad (6.6)$$

6.2.3 Calibrage d'une caméra

Après avoir défini les principes et notations ci-dessus, nous allons exposer le principe de calibrage d'une caméra, c'est-à-dire la recherche des paramètres intrinsèques et extrinsèques de la caméra, afin de transformer un point 3D de l'espace en point 2D de l'image.

Pour ce faire, il existe plusieurs approches possibles [Dev03]. La régression linéaire, l'optimisation non-linéaire, les points de fuite sont quelques exemples de méthode. La méthode la plus employée actuellement reste le calibrage multi-plans [Zha00], à partir de plusieurs images acquises d'une mire plane et dont les motifs sont parfaitement connus (disposition, dimensions...).

6.2.4 Auto-calibrage en milieu sous-marin

En milieu sous-marin, il est nécessaire d'obtenir les paramètres de la caméra. En effet, ceux-ci dépendent de l'indice optique de l'eau, qui varie en fonction du milieu et de la salinité. Malheureusement, il est quasiment impossible de déployer une mire sous-marine et d'acquérir les images adéquates. En effet, cette opération a un coût opérationnel assez élevé : en temps pour le véhicule sous-marin muni d'un bras et en moyens techniques et humains : le technicien manoeuvrant le bras doit acquérir des images bien précises. Ainsi, un calibrage tel qu'expliqué dans la partie précédente ne peut pas être effectué. Il serait possible de calibrer la caméra hors de l'eau mais malheureusement, les données de profondeur ne sont pas toujours connues. Aussi, cette technique a été abandonnée.

On a donc recours à un auto-calibrage de la caméra dans le milieu. N. Pessel [Pes03] a proposé une méthode complète d'auto-calibrage, qui repose sur l'adaptation de méthodes de calibrage stéréoscopique (deux caméras solidaires) à une caméra en mouvement. Nous allons brièvement présenter cette méthode, dont les principales étapes sont regroupées sur la figure 6.3.

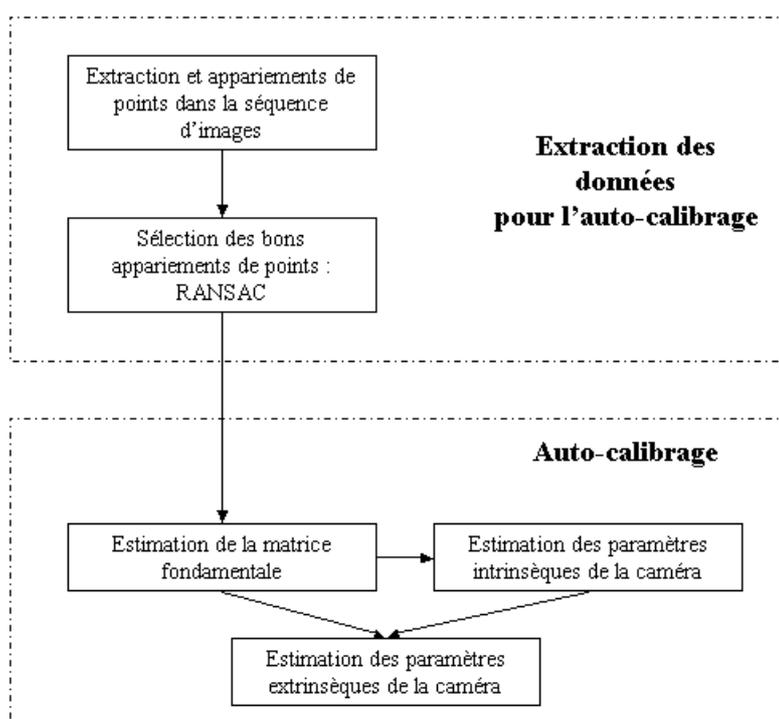


FIG. 6.3 – Méthode d'auto-calibrage développée par N. Pessel [Pes03].

La première étape consiste en une extraction de points, qui est réalisée grâce à l'algorithme de suivi de points KLT (décrit dans la partie 5.4.2). A partir des points caractéristiques suivis dans la séquence d'images, la géométrie de la scène observée est déterminée grâce aux différentes orientations de la caméra lors des prises de vue. Hartley [Har95] propose une méthode pour estimer la matrice fondamentale reliant deux images. Cette méthode nécessite au moins 8 appariements de points. Afin de rendre l'estimation de la matrice fondamentale plus robuste, N. Pessel a intégré la méthode RANSAC (RANdom SAMpling Consensus), développée par Fischler et Bolles [FB81]. Celle-ci permet de sélectionner, parmi les points suivis dans deux images

successives, ceux qui présentent les "meilleurs" appariements. L'estimation de la matrice fondamentale est réalisée à partir de ces "meilleurs" appariements.

L'estimation des paramètres intrinsèques est réalisée par la méthode proposée par Mendonça et Cipolla [MC99], appliquée à une séquence de 5 images. Cet algorithme est basé sur la minimisation d'une fonction de coût. Celle-ci est réalisée à partir des matrices fondamentales estimées (sur une séquence de 5 images, on estime 10 matrices fondamentales), et retourne le minimum correspondant aux paramètres intrinsèques. Lors de nos expérimentations, nous avons utilisé la méthode d'auto-calibrage de N. Pessel pour déterminer les paramètres de la caméra que nous avons utilisée.

6.2.5 Influence de la précision sur la distance (caméra, sol)

Dans cette partie, on suppose que la caméra se déplace toujours de façon horizontale. Jusqu'à présent, nous avons supposé implicitement que le sol était horizontal ou en tout cas, qu'il n'avait pas de relief trop accentué ou que la caméra ne subissait pas de brusque remontée, à cause d'un courant quelconque. Il arrive cependant que ce soit le cas. Ainsi, nous avons étudié l'influence de la distance caméra-sol sur la position du pixel dans l'image. Soit $P = [X, Y, Z, 1]^T$ un point fixe du fond marin, exprimé dans le repère monde 3D en coordonnées homogènes et U le pixel correspondant dans l'image. Soit \mathbb{P} la matrice de passage du repère monde au repère image. Pour mettre en valeur la contribution de la translation verticale entre la scène et la caméra t_Z dans \mathbb{P} , nous noterons la transformation \mathbb{P}_{t_Z} .

$$\mathbb{P}_{t_Z} = \begin{bmatrix} \alpha_u & 0 & u_0 & u_0 t_Z \\ 0 & \alpha_v & v_0 & v_0 t_Z \\ 0 & 0 & 1 & t_Z \end{bmatrix} \quad (6.7)$$

On suppose que le véhicule porteur de la caméra fait une "embardée" et se retrouve dt_Z plus haut ou plus bas. Alors

$$\mathbb{P}_{t_Z+dt_Z} = \begin{bmatrix} \alpha_u & 0 & u_0 & u_0(t_Z + dt_Z) \\ 0 & \alpha_v & v_0 & v_0(t_Z + dt_Z) \\ 0 & 0 & 1 & t_Z + dt_Z \end{bmatrix} \quad (6.8)$$

Alors le point imagé devient $U' = \mathbb{P}_{t_Z+dt_Z} P$ soit

$$U' = \begin{bmatrix} \alpha_u x + u_0(t_Z + dt_Z) \\ \alpha_v y + v_0(t_Z + dt_Z) \\ Z + t_Z + dt_Z \end{bmatrix} \quad (6.9)$$

Supposons que $Z = 0$ (ce qui revient à prendre un axe du repère monde passant par P). On a alors

$$U' = \begin{bmatrix} \alpha_u X + u_0(t_Z + dt_Z) \\ \alpha_v Y + v_0(t_Z + dt_Z) \\ t_Z + dt_Z \end{bmatrix} \quad (6.10)$$

Soit en passant aux coordonnées homogènes

$$U' = \begin{bmatrix} \frac{\alpha_u X}{t_Z + dt_Z} + u_0 \\ \frac{\alpha_v Y}{t_Z + dt_Z} + v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\alpha_u X}{t_Z (1 + \frac{dt_Z}{t_Z})} + u_0 \\ \frac{\alpha_v Y}{t_Z (1 + \frac{dt_Z}{t_Z})} + v_0 \\ 1 \end{bmatrix} \quad (6.11)$$

On suppose $t_Z \gg dt_Z$, on a alors $\frac{1}{1 + \frac{dt_Z}{t_Z}} \sim 1 - \frac{dt_Z}{t_Z}$. On en déduit donc :

$$U' = \begin{bmatrix} \frac{\alpha_u X}{t_Z + dt_Z} + u_0 \\ \frac{\alpha_v Y}{t_Z + dt_Z} + v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\alpha_u X}{t_Z (1 + \frac{dt_Z}{t_Z})} + u_0 \\ \frac{\alpha_v Y}{t_Z (1 + \frac{dt_Z}{t_Z})} + v_0 \\ 1 \end{bmatrix} \quad (6.12)$$

Soit encore :

$$U' = \begin{bmatrix} \frac{\alpha_u X}{t_Z} + u_0 - \frac{dt_Z}{t_Z^2} \alpha_u X \\ \frac{\alpha_v Y}{t_Z} + v_0 - \frac{dt_Z}{t_Z^2} \alpha_v Y \\ 1 \end{bmatrix} \sim U - \begin{bmatrix} \frac{dt_Z}{t_Z^2} \alpha_u X \\ \frac{dt_Z}{t_Z^2} \alpha_v Y \\ 0 \end{bmatrix} \quad (6.13)$$

Pour résumer : un changement de position verticale de la caméra induit un déplacement

$$U' - U = - \begin{bmatrix} \frac{dt_Z}{t_Z^2} \alpha_u & 0 & 0 & 0 \\ 0 & \frac{dt_Z}{t_Z^2} \alpha_v & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} P \quad (6.14)$$

6.3 Fusion des localisations AR/(C,V) et AR/I

Dans le chapitre précédent a été expliqué comment estimer les instants de rendez-vous à partir de la séquence d'images et de la localisation A/(C,V). Ces instants de rendez-vous estimés sont ensuite utilisés dans le chapitre 3 pour réaliser une localisation AR/(C,V) de l'engin.

Dans le chapitre 5, deux méthodes d'estimation de mouvement pour la construction de mosaïque ont été comparées. Le RMR [OB95] a été retenu et une localisation AR/I a été réalisée à partir des déplacements estimés par le RMR, en partant d'une hypothèse de variances augmentant linéairement avec le temps pour les localisation A/I et R/I.

Ces deux localisations AR/(C,V) et AR/I apportent chacune des avantages. Il convient alors de les fusionner par un estimateur des moindres carrés pondérés, par exemple, pour profiter des avantages de chacune. Malheureusement, comme expliqué dans la partie 5.4.3 du chapitre 5, les variances des paramètres de la méthode RMR retenue ne sont pas disponibles. Ainsi, pour réaliser tout de même une fusion des deux localisations AR, un rapport ρ constant a été introduit entre les variances de la localisation AR/(C,V) et celles de la localisation AR/I. Un rapport de variances constant signifie que la confiance que l'on accorde à l'une ou l'autre des localisations ne varie pas dans le temps (on fait toujours plus confiance à l'une qu'à l'autre, de manière constante).

Au niveau des expérimentations, nous proposons tout d'abord une comparaison des localisations AR/(C,V) et AR/I à partir de données acquises grâce au portique PISCATOR (voir partie 6.4.4). Puis un premier résultat pour la localisation AR/(C,V,I) -résultant de la fusion des localisations AR/(C,V) et AR/I et de l'hypothèse sur le rapport de variances constant- est présenté dans la partie 6.4.5.

6.4 Expérimentations/résultats

Dans cette partie sont présentés les résultats illustrant l'auto-calibrage de la caméra (partie 6.4.1), l'influence de la position de la caméra dans la partie 6.4.2. Une comparaison des localisations A/(C,V) et AR/(C,V) nous permet de tirer quelques conclusions dans la partie 6.4.3. Puis une comparaison des localisations AR/(C,V) et AR/I est présentée dans la partie 6.4.4. Enfin, une étude expérimentale est menée pour établir la faisabilité de la fusion des localisations AR/(C,V) et AR/I dans la partie 6.4.5.

6.4.1 Auto-calibrage de la caméra

Comme expliqué dans la section 6.2, pour pouvoir fusionner des données, il faut qu'elles soient basées sur un même référentiel. La matrice \mathbb{P}_{t_Z} de passage est constituée de la matrice des paramètres intrinsèques \mathbb{K} et de la matrice des paramètres extrinsèques \mathbb{T}_R . N. Pessel [Pes03] a auto-calibré la caméra Kappa que nous avons également utilisée. Elle a estimé \mathbb{K} pour une image de 646×509 pixels :

$$\mathbb{K} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 823.77 & 0 & 322.50 \\ 0 & 824.77 & 254 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.15)$$

\mathbb{K} étant déterminée pour une taille d'image donnée, nous devons la transformer pour pouvoir l'utiliser. Il faut simplement modifier u_0 et v_0 qui sont égaux à $u_0 = \frac{NbColonnes+1}{2}$ et $v_0 = \frac{NbLignes+1}{2}$. Les images acquises ont une taille de 622×478 , ce qui donne pour une "bonne" évaluation de u_0 et v_0 les résultats suivants : $u_0 = 311.5$ et $v_0 = 239.5$.

Pour obtenir \mathbb{P}_{t_Z} , il suffit donc de déterminer la distance caméra-fond du bassin t_Z . Dans le cadre de nos expérimentations, nous avons mesuré $t_Z = 0.92\text{m}$. D'où la matrice de passage $3\text{D} \rightarrow 2\text{D}$ \mathbb{P}_{t_Z} :

$$\mathbb{P}_{t_Z} = \begin{bmatrix} 823.77 & 0 & 311.5 & 286.58 \\ 0 & 824.77 & 239.5 & 220.34 \\ 0 & 0 & 1 & 0.92 \end{bmatrix} \quad (6.16)$$

6.4.2 Influence de la position de la caméra

La trajectoire programmée pour être exécutée par le robot est exprimée en mètres. Pour pouvoir comparer celle-ci et la localisation AR/I, il faut la projeter dans le plan de la caméra.

Pour ce faire, nous utilisons la transformation de changement de repère exposée en 6.2.2. Elle est déterminée à partir de la matrice des paramètres intrinsèques de la caméra (voir partie 6.4.1) et de la matrice des paramètres extrinsèques. On suppose que la transformation consiste en une simple translation verticale. Dans la partie 6.2.5, nous avons parlé de l'influence de la précision sur la position verticale de la caméra dans la transformation $3\text{D} \rightarrow 2\text{D}$.

Dans cette expérience, nous allons donc étudier l'influence de la précision sur la translation (en mètres) pour exprimer les positions de la trajectoire du robot en pixels.

Nous avons recalculé la trajectoire 2D à partir de la trajectoire 3D connue en faisant varier t_Z de 5cm autour de la vraie valeur de 0.92m. 10 simulations ont été réalisées. Les trajectoires 2D obtenues (exprimées en

pixels) sont illustrées sur les figures 6.4(a), 6.4(b) et 6.4(c). Remarquons tout d'abord que plus la distance sol-caméra est petite, plus la trajectoire image est grande, ce qui est intuitif. D'autre part, nous pouvons remarquer des écarts assez importants, de l'ordre de 500 pixels pour les x et pour les y pour une différence d'altitude de 5cm.

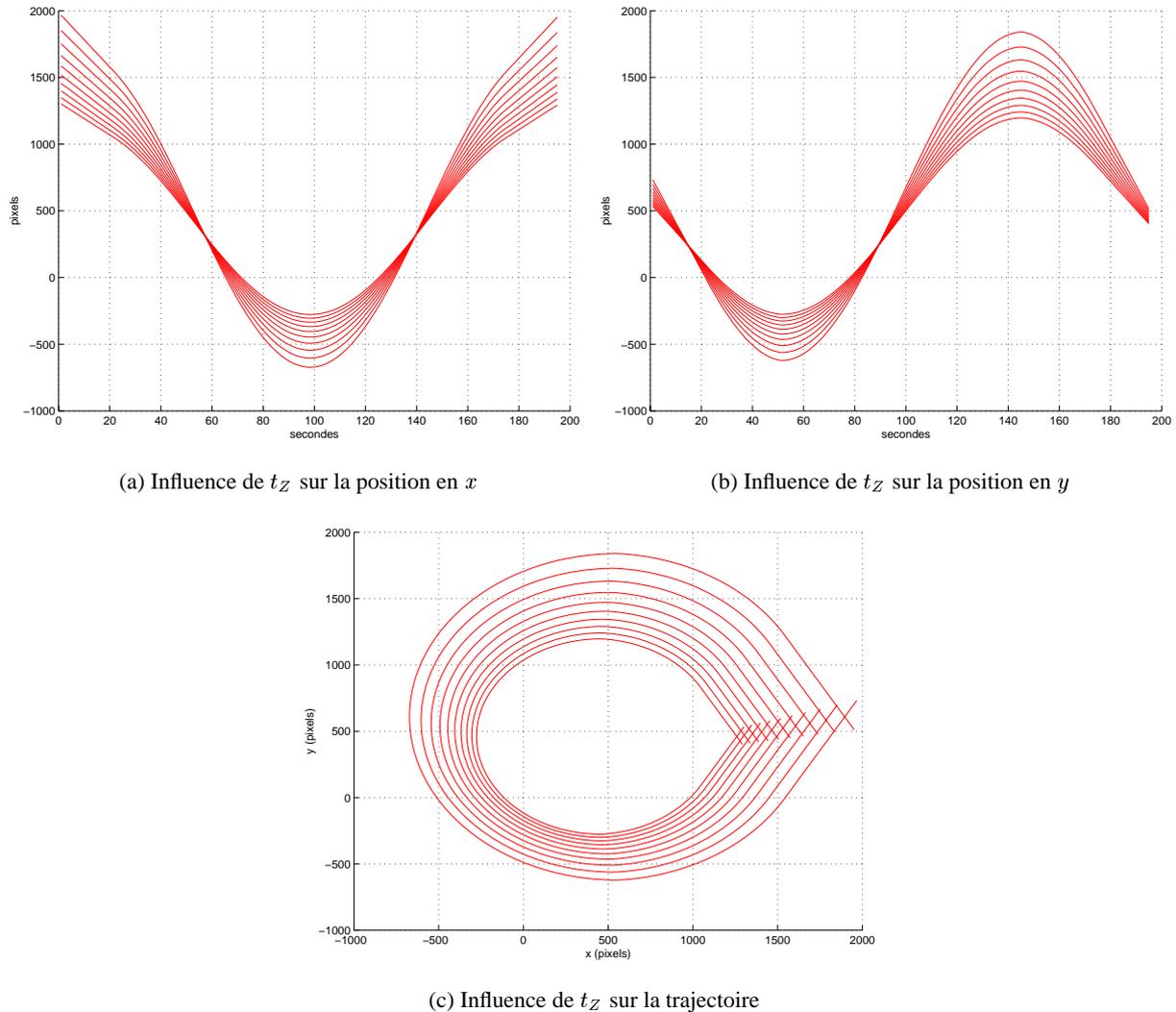
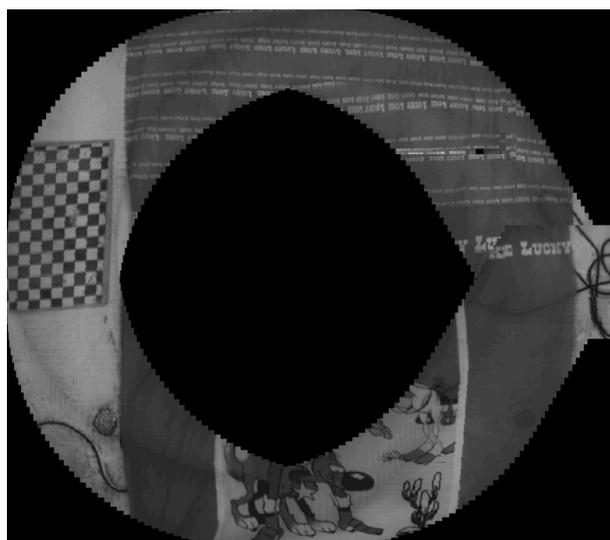
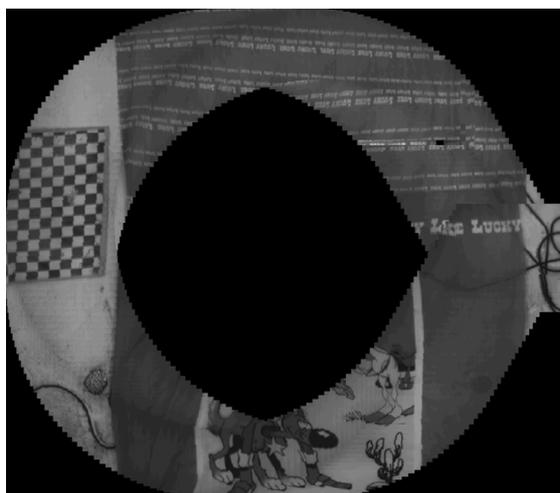


FIG. 6.4 – Influence de t_Z sur la transformation $3D \rightarrow 2D$ pour la localisation $AR/(C,V)$.

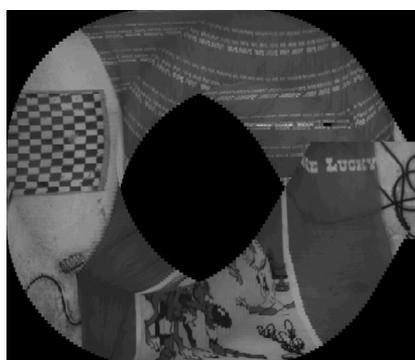
Sur les figures 6.5(a) à 6.5(c) page ci-contre, sont présentées les mosaïques reconstruites à partir de la trajectoire programmée sur le robot, selon l'altitude t_Z . On remarque que la mosaïque construite à partir de notre mesure $t_Z = 0.92m$ est l'image la mieux reconstruite du fond du bassin, et celle pour laquelle on voit le mieux que le robot revient sur sa position initiale.



(a) Mosaïque construite pour $t_z = 0.92\text{m}$.



(b) Mosaïque construite pour $t_z = 1\text{m}$.



(c) Mosaïque construite pour $t_z = 1.5\text{m}$.

FIG. 6.5 – Comparaison des mosaïques construites à partir de la trajectoire vraie 2D calculée pour trois altitudes différentes, pour la trajectoire "alpha".

6.4.3 Comparaison localisation A/(C,V) et localisation AR/(C,V)

Les résultats que nous présentons ici peuvent sembler ne pas être à une place adéquate, ils sembleraient mieux placés dans le chapitre 3. Pourtant, ils ont été obtenus après avoir réalisé la comparaison entre localisation AR/(C,V) et localisation AR/I (section 6.4.4) où nous avons comparé les positions, les déplacements et les mosaïques. Aussi, la présence de ces résultats à cet endroit du manuscrit ne présente pas d'incohérence.

A partir des trajectoires "alpha" et à deux boucles que nous avons modélisées pour le robot, nous avons comparé les localisation A/(C,V) et localisation AR/(C,V) à la trajectoire vraie. Pour toutes les figures 6.6 page suivante et 6.7 page 142, nous avons représenté les résultats correspondant à la trajectoire vraie en trait plein gras, les résultats correspondant à localisation A/(C,V) en pointillés "..." ou "." et les résultats correspondant à la localisation AR/(C,V) en pointillés "- -" ou par des "+". Sont représentés pour la trajectoire "alpha" ainsi que pour la trajectoire à deux boucles :

1. les positions pour l'abscisse x , en pixels (fig. 6.6(a), et 6.7(a)),
2. les positions pour l'ordonnée y , en pixels (fig. 6.6(b), et 6.7(b)),
3. les positions dans le plan 2D de la caméra, en pixels (fig. 6.6(c), et 6.7(c)),
4. les déplacements calculés à partir des trajectoires estimées Δ_x , en pixels (fig. 6.6(d) et 6.7(d)),
5. les déplacements calculés à partir des trajectoires estimées Δ_y , en pixels (fig. 6.6(e) et 6.7(e)),
6. la distance entre deux points consécutifs $\sqrt{\Delta_x^2 + \Delta_y^2}$, en pixels (fig. 6.6(f) et 6.7(f)).

Nous pouvons remarquer que du point de vue des positions, la localisation AR/(C,V) est bien plus proche de la trajectoire vraie que la localisation A/(C,V). C'est un résultat que nous avons déjà obtenu dans la partie 3. Le résultat le plus frappant provient des déplacements : en effet, on remarque que les points "." ou "+" représentant les localisations A/(C,V) et AR/(C,V) sont très proches. On peut donc en conclure que bien que les trajectoires soient améliorées par l'auto-estimation, les déplacements ne sont pas lissés ni corrigés. Cela représente donc une limitation à l'algorithme d'auto-estimation du point de vue des déplacements.

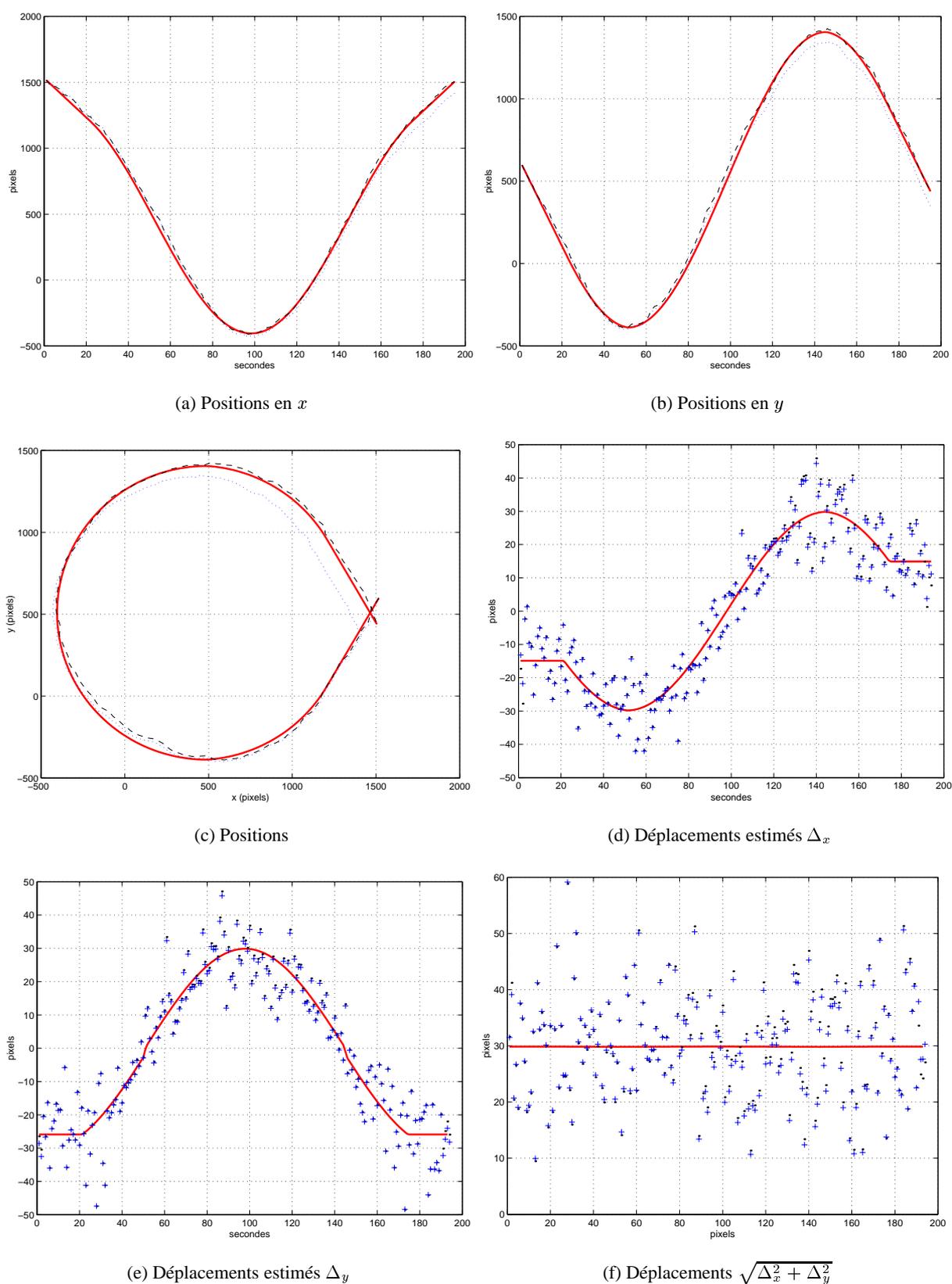
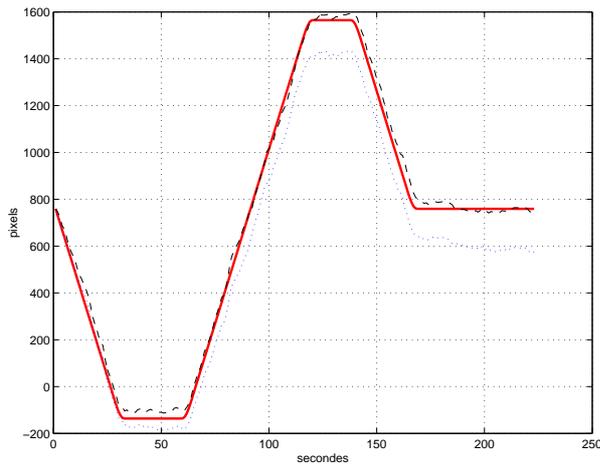
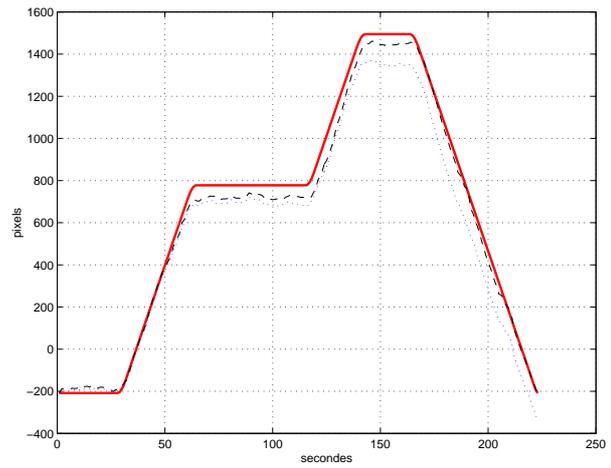
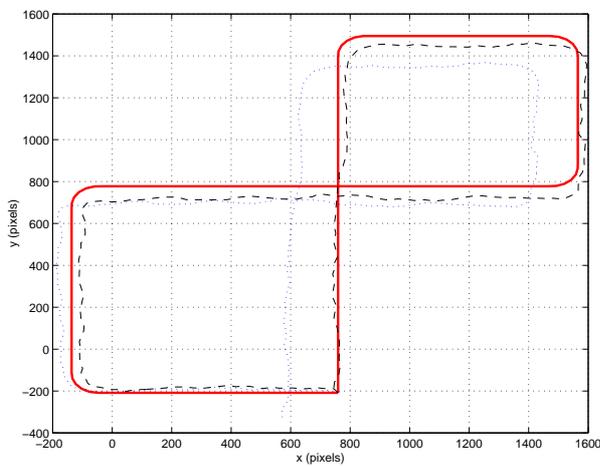


FIG. 6.6 – Comparaison trajectoires bruitée et auto-estimée pour la trajectoire "alpha". En trait plein, la trajectoire programmée pour le robot, en pointillés '.' et en '.' les résultats obtenus pour la localisation A/(C,V), en pointillés '- -' et en '+' les résultats obtenus pour la localisation AR/AR(C,V).

(a) Positions en x (b) Positions en y 

(c) Positions

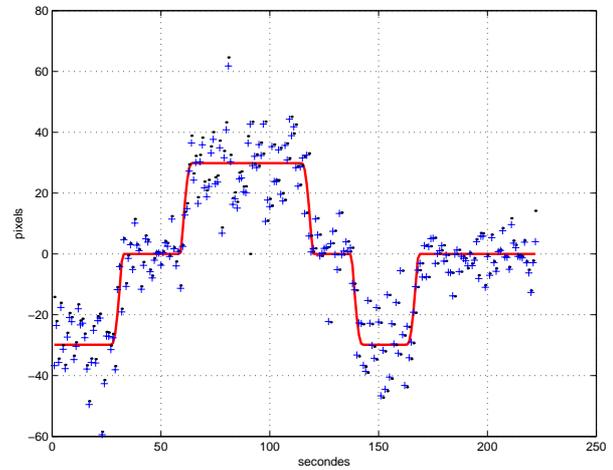
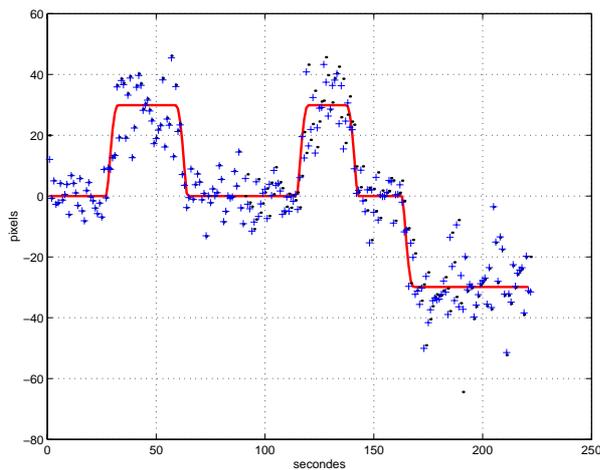
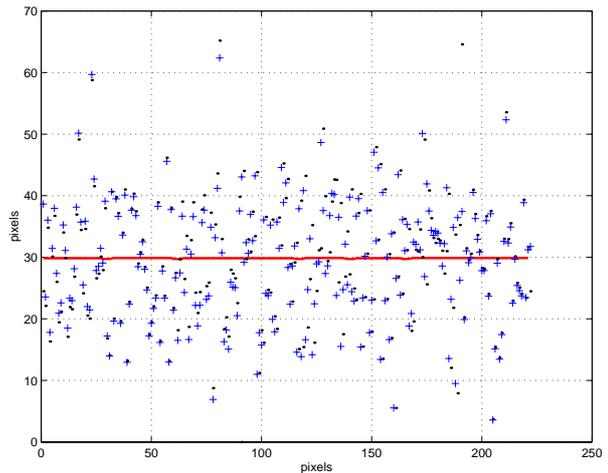
(d) Déplacements estimés Δ_x (e) Déplacements estimés Δ_y (f) Déplacements $\sqrt{\Delta_x^2 + \Delta_y^2}$

FIG. 6.7 – Comparaison trajectoires bruitée et auto-estimée pour la trajectoire à deux boucles. En trait plein, la trajectoire programmée pour le robot, en pointillés '...' et en '.' les résultats obtenus pour la localisation AR/(C,V), en pointillés '- -' et en '+' les résultats obtenus pour la localisation AR/I/(C,V).

6.4.4 Comparaison des localisations AR/(C,V) et AR/I

Dans cette partie, nous présentons une comparaison de la localisation AR/(C,V) (donnée par l'auto-estimation) et de la localisation AR/I (obtenue à partir de la méthode RMR) pour la trajectoire "alpha" à un instant de rendez-vous et pour la trajectoire à deux boucles, comportant deux instants de rendez-vous.

Rappelons que la méthode d'auto-estimation estime une localisation AR/(C,V) en mètres, tandis que la localisation AR/I fournit des positions en pixels. Pour pouvoir comparer ces résultats, il a été nécessaire de connaître la transformation repère scène 3D (mètres) et repère image 2D (pixels). Cette partie a été vue en 6.2.2 et expérimentée en 6.4.1. Le référentiel commun choisi est donc le repère lié à l'image.

Les positions ont donc été comparées en pixels, ainsi que les déplacements. Les déplacements ont été déterminés en faisant une simple différence entre deux positions estimées par localisation AR/(C,V) ou par localisation AR/I.

Tous les résultats relatifs à la localisation AR/(C,V) sont représentés par des pointillés "..." ou par des points ".", tandis que les résultats relatifs à la localisation AR/I sont représentés par des pointillés "—" ou par des "+".

Sont représentés pour la trajectoire "alpha" ainsi que pour la trajectoire à deux boucles :

1. les positions pour l'abscisse x , en pixels (fig. 6.8(a), et 6.9(a)),
2. les positions pour l'ordonnée y , en pixels (fig. 6.8(b), et 6.9(b)),
3. les positions dans le plan 2D de la caméra, en pixels (fig. 6.8(c), et 6.9(c)),
4. les déplacements calculés à partir des trajectoires estimées Δ_x , en pixels (fig. 6.8(d) et 6.9(d)),
5. les déplacements calculés à partir des trajectoires estimées Δ_y , en pixels (fig. 6.8(e) et 6.9(e)),
6. la distance entre deux points consécutifs $\sqrt{\Delta_x^2 + \Delta_y^2}$, en pixels (fig. 6.8(f) et 6.9(f)).

Du point de vue des **positions**, nous pouvons remarquer que localisation AR/(C,V) rend un meilleur résultat que la localisation AR/I. En effet, la localisation AR/I présente une dérive au niveau des positions, par rapport à la trajectoire vraie. Cependant, du point de vue des **déplacements**, la localisation AR/I fournit des résultats plus lisses et plus proches de la trajectoire vraie que la localisation AR/(C,V).

L'idée serait donc de fusionner les positions issues de la localisation AR/(C,V) avec les déplacements estimés par la localisation AR/I pour garantir à la fois une bonne estimation de la position ainsi qu'une bonne estimation des déplacements. Cependant, les variances des paramètres du RMR ne sont pas disponibles. Ainsi, une méthode de fusion comme le filtrage de Kalman est peu envisageable.

La comparaison a été effectuée sur les **mosaïques** également. Les mosaïques exposées sur les figures 6.10(a) (resp. 6.11(a)) à 6.10(d) page 147 (resp. 6.11(d)) sont construites de la façon suivante. Les déplacements déduits de la trajectoire vraie (resp. de localisation AR/(C,V), resp. de la localisation AR/I) sont transmis au logiciel de mosaïques MATISSE® pour juxtaposer les images.

Nous pouvons remarquer tout d'abord que les tailles des mosaïques reconstruites à partir des localisation AR/(C,V) sont proches des tailles des mosaïques reconstruites à partir des trajectoires vraies. En effet, on

voit que les mosaïques reconstruites à partir des localisations AR/I sont de taille plus réduite.

Cependant, la mosaïque obtenue à partir de la localisation AR/(C,V) est moins lisse au niveau de la transition inter-image que la mosaïque déduite de la localisation AR/I. On revient à la remarque précédente qui consistait à dire que l'auto-estimation sur les mesures de cap et de vitesse ne lissait pas les déplacements mais en revanche, estimait bien les positions de l'engin.

D'autre part, pour la localisation AR/(C,V) et AR/I, la dérive au niveau des points initial et final est réduite par rapport à la mosaïque initiale, construite à partir des images seules (mosaïque RMR) : les instants de rendez-vous sont bien gérés. On observe clairement cet effet sur la figure 6.11(c) : la localisation AR/(C,V) aussi bien que la localisation AR/I tiennent bien compte des instants de rendez-vous (contrairement à la mosaïque RMR initiale où l'image de Lucky Luke est recouverte par l'image de Rantanplan) : Lucky Luke est assez bien reconstruit, grâce à la prise en compte des instants de rendez-vous.

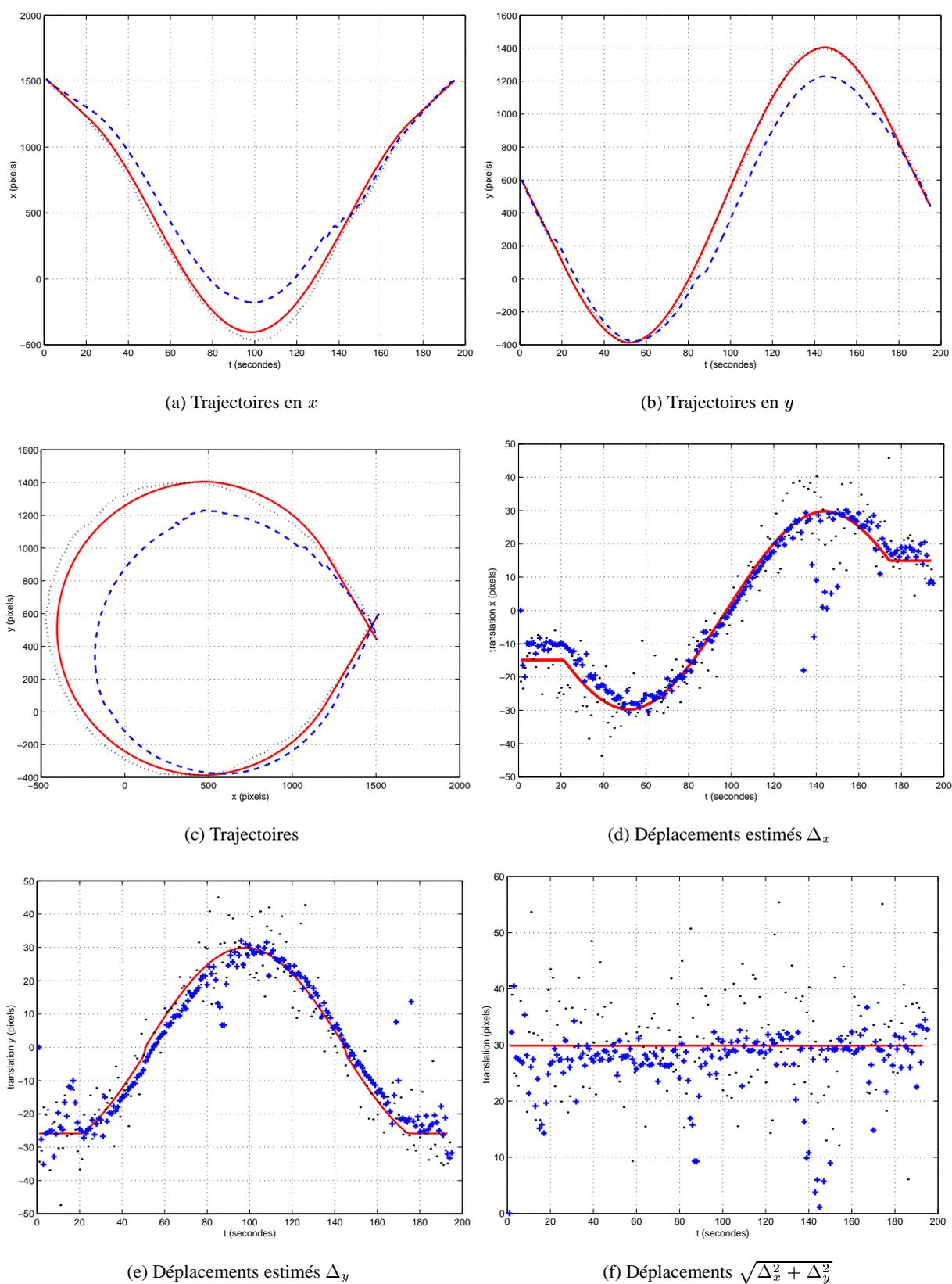
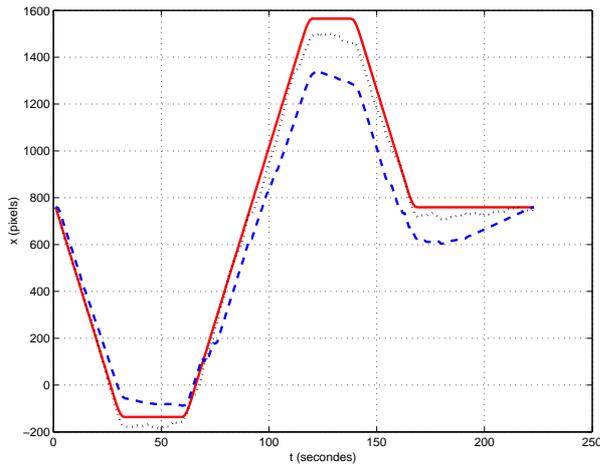
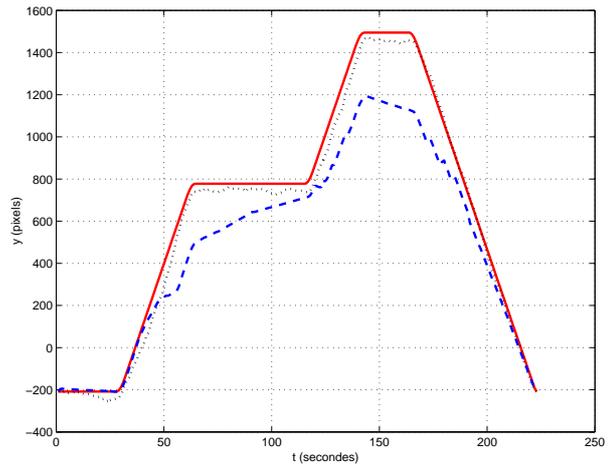


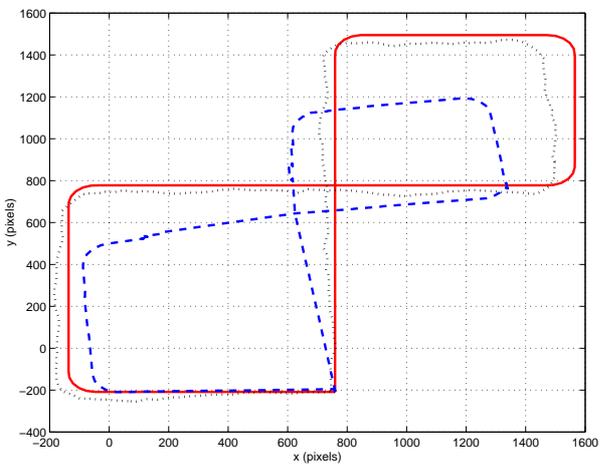
FIG. 6.8 – Comparaison trajectoires RMR et auto-estimée pour la trajectoire "alpha".



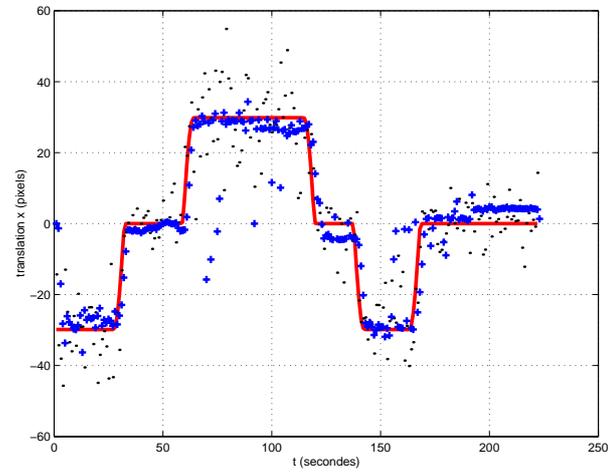
(a) Trajectoires en x



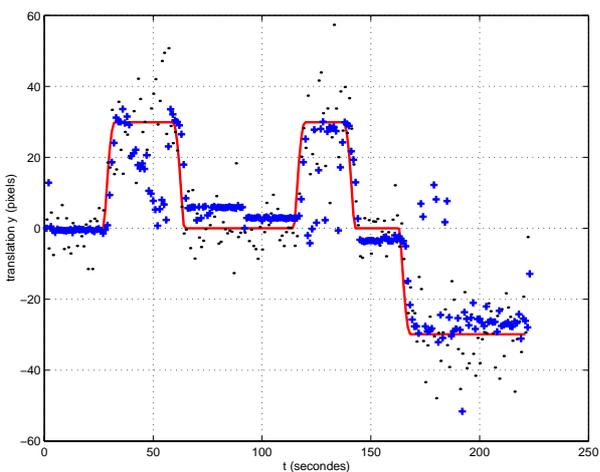
(b) Trajectoires en y



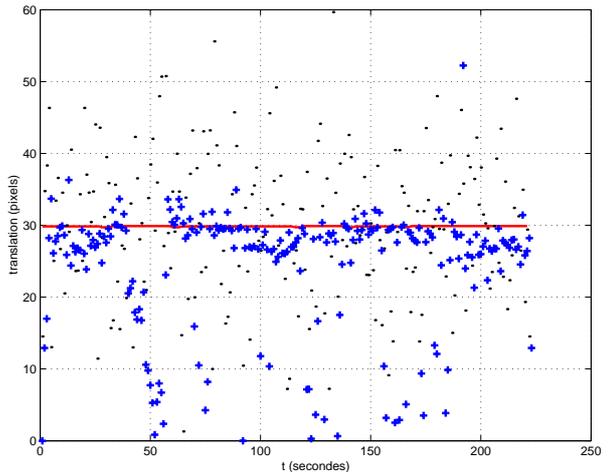
(c) Trajectoires



(d) Déplacements estimés Δ_x

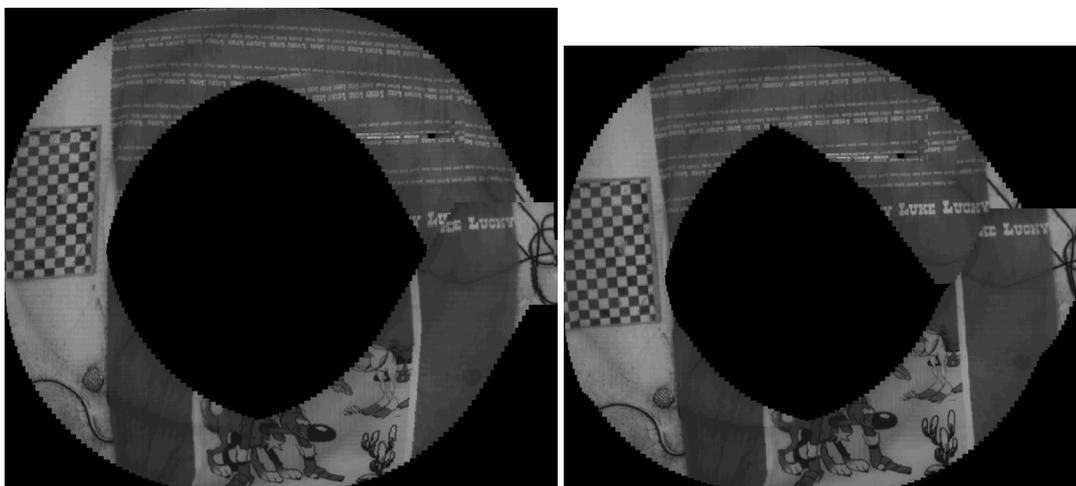


(e) Déplacements estimés Δ_y



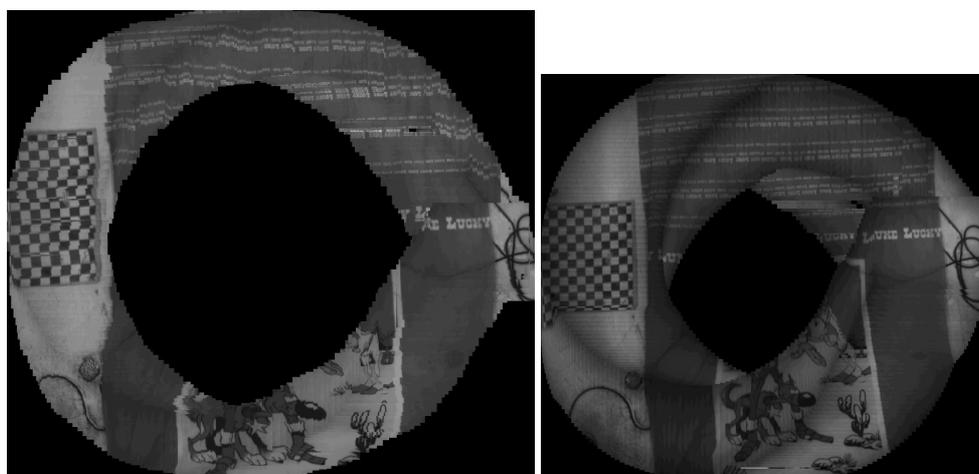
(f) Déplacements $\sqrt{\Delta_x^2 + \Delta_y^2}$

FIG. 6.9 – Comparaison trajectoires RMR et auto-estimée pour la trajectoire à deux boucles.



(a) Mosaïque construite à partir de la trajectoire vraie

(b) Mosaïque initiale, construite à partir du RMR



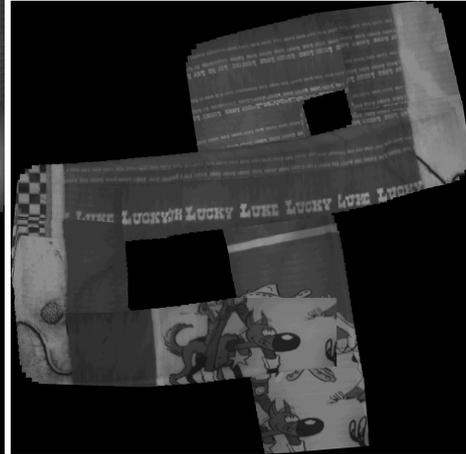
(c) Mosaïque reconstruite à partir de la localisation AR/(C,V)

(d) Mosaïque reconstruite à partir de la localisation AR/I

FIG. 6.10 – Comparaison des mosaïques construites par la trajectoire vraie, le RMR et par les localisations AR/(C,V) et AR/I pour la trajectoire "alpha".



(a) Mosaïque construite à partir de la trajectoire vraie



(b) Mosaïque initiale, construite à partir du RMR



(c) Mosaïque construite à partir de la localisation AR/(C,V)



(d) Mosaïque reconstruite à partir de la localisation AR/I

FIG. 6.11 – Comparaison des mosaïques construites par la trajectoire vraie, le RMR et par les localisations AR/(C,V) et AR/I pour la trajectoire à deux boucles.

6.4.5 Fusion localisation AR/(C,V) et localisation AR/I

La fusion des localisation AR/(C,V) et AR/I a été réalisée ici avec un rapport de variances ρ constant. Plusieurs expériences ont été menées pour la fusion, en faisant varier le rapport ρ de 0.25 à 4 par pas de 0.125.

Sur la figure 6.12 est présentée la fusion entre la localisation AR/(C,V) et la localisation AR/I (RMR) (voir partie 6.3). La trajectoire "alpha" programmée pour le robot est représentée en trait plein gras, tandis que les trajectoires fusionnées pour les ρ sont représentées en trait plein plus fin.

Les résultats montrent que les trajectoires estimées grâce au ρ ne sont pas proches de la trajectoire vraie. Ainsi, cette fusion n'est pas optimale, ce qui est normal puisque nous avons fait l'hypothèse d'un rapport de variances constant entre les localisations AR/(C,V) et AR/I. Ceci est dû en grande partie à la méconnaissance des variances des paramètres du RMR. En revanche, sous réserve que ces variances soient accessibles, une fusion serait possible.

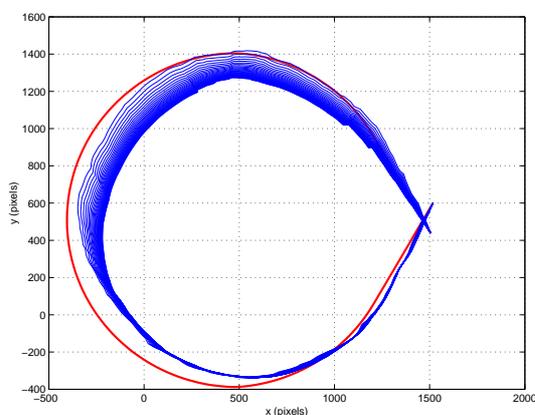


FIG. 6.12 – Fusion des localisations AR/(C,V) et AR/I pour la trajectoire "alpha".

6.5 Conclusion

Lors des expérimentations réalisées en bassin, la mesure de l'altitude de la caméra a été d'une précision relative. Les résultats qui permettent de passer d'une trajectoire en mètres à une trajectoire en pixels montrent bien que la précision sur l'altitude de la caméra est nécessaire pour réaliser cette transformation.

D'autre part, nous avons évoqué dans la partie 5.8.2 l'importance de la déformation radiale. Or, ces déformations ne sont pas prises en compte dans l'estimation des paramètres de la caméra. C'est une évolution possible de la thèse de N. Pessel [Pes03].

Nous avons également remarqué, en comparant a posteriori les positions et déplacements des localisations A/(C,V) et AR/(C,V), que l'estimation de la trajectoire en position à partir de mesures de cap et vitesse, présentée dans le chapitre 3, ne permettait pas d'estimer les déplacements alors qu'elle permet d'estimer correctement les positions. Ces constatations rejoignent la problématique du départ qui était une probléma-

tique de localisation d'un véhicule porteur d'une caméra, c'est-à dire une estimation de ses positions et non de ses déplacements. En revanche, la méthode d'auto-estimation permet d'estimer efficacement les positions et d'améliorer la construction d'une mosaïque en terme de localisation (comparaison des mosaïques reconstruites à partir de la localisation AR/(C,V) et de la localisation AR/I).

Enfin, dans ce chapitre, nous avons proposé une première fusion non-optimale des localisation AR/(C,V) et localisation AR/I. Sous réserve que les biais et variances des paramètres du RMR soient disponibles, une fusion serait possible, par une méthode des moindres carrés pondérés par exemple.

Chapitre 7

Alternative : déformation de la mosaïque initiale par localisation AR/(C,V)

Dans le chapitre précédent, une fusion entre les localisation AR/I (estimée à partir des estimations de déplacements du le RMR) et localisation AR/(C,V) (estimée par l'auto-estimateur à partir des caps et vitesses bruités) a été proposée mais elle n'est pas optimale, en raison de la non-connaissance des biais et variances des paramètres du RMR.

Si cette fusion n'est pas réalisable ou si la séquence d'images n'est plus disponible pour reconstruire les mosaïques à partir des localisations AR, nous proposons une alternative à cette solution, afin de corriger tout de même la mosaïque. Il s'agit de construire la mosaïque par la méthode RMR, de réaliser la localisation AR/(C,V) puis à déformer cette mosaïque initiale grâce à la localisation AR/(C,V), selon des techniques dites de *warping* [BJR04].

Ci-après, sont présentées quelques généralités et quelques exemples d'utilisation de la déformation d'images, puis l'application du warping à notre problématique est exposée.

7.1 Introduction à la déformation d'images

Lors de l'acquisition d'images, des erreurs dues au capteur optique apparaissent. Elles peuvent être de deux types : déformation/distorsion géométrique ou radiométrique (identifiées par la NASA en 1971 [BS71]). Les déformations radiométriques (distribution d'intensité non-uniforme) ne seront pas traitées ici. Les déformations géométriques très visibles en télédétection peuvent être dues au relief du terrain imagé, à l'angle de vue, à l'altitude... Des exemples de déformation dues au capteur optique sont présentées figure 7.1.

A l'origine, le problème posé en imagerie télédétection était de mettre en correspondance plusieurs images d'une même scène provenant de capteurs différents, donc de distorsions différentes. Il s'agissait donc de compenser les distorsions des capteurs d'images. Plus récemment, ces techniques de warping ont été utilisées en synthèse d'images, pour appliquer une texture sur un objet 3D.

La déformation d'image numérique consiste en :

1. une transformation spatiale pour définir le réarrangement des pixels,
2. une interpolation pour calculer les valeurs des pixels.

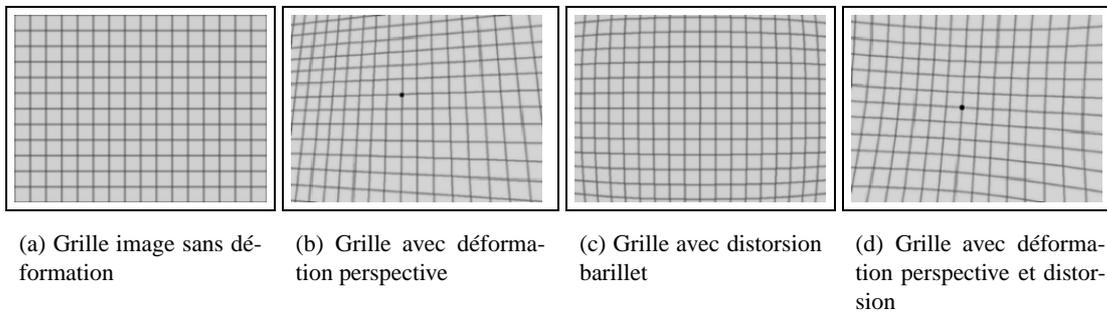


FIG. 7.1 – Exemples de déformation.

7.2 Quelques exemples d'utilisation du warping en traitement d'images

Les techniques de déformation d'images sont très utilisées dans le cas de correction due au système d'acquisition [YC03, Alt03]. En effet, la lentille d'un appareil photo ou d'une caméra induit une distorsion sur l'image. L'idée est donc de rechercher la fonction qui relie l'image déformée à l'image telle qu'elle devrait être. Cette étape de calibration consiste à déterminer pratiquement comment les coordonnées sont transformées en passant du monde réel à l'image (et vice versa).

Le principe de la calibration est le suivant :

1. Acquérir l'image d'une mire dont on connaît parfaitement les coordonnées
2. Extraire des points de contrôle par une méthode de traitement d'images
3. Déterminer les paramètres d'un modèle choisi
4. Une fois le modèle choisi, on peut calculer les coordonnées image de tout point du monde. On peut également fournir une image corrigée grâce au modèle réciproque.

La déformation d'images peut également se retrouver dans une phase de suivi (tracking) d'objet déformable ou lors d'un déplacement de la caméra [HHN02]. En effet, l'objet pouvant se déformer, l'auteur réalise une étape de prédiction de la déformation.

Afin de quantifier le biais dû à l'accumulation de l'erreur d'estimation de mouvement pour la construction de mosaïques, Garcia a mis en place des expériences en laboratoire dans [GBC01]. Le bras d'un robot mobile est programmé pour suivre une trajectoire donnée, au-dessus d'un poster donné. Une caméra est fixée à ce bras et visualise le poster. Une première phase de calibration est réalisée. Dans ce but, le premier poster représente une mire comportant une matrice de disques noirs sur un fond blanc. Les dimensions et la répartition des disques noirs de cette mire sont connues parfaitement. Une mosaïque virtuelle représentant

la mire est construite en connaissant les paramètres exacts des transformations inter-image, à partir des dimensions et de la répartition des disques de la mire.

Une séquence d'images est acquise. Les disques noirs de la mire sont repérés et détectés. Une transformation projective 2D est alors déterminée pour relier les disques noirs détectés aux disques noirs de la mosaïque virtuelle dont la position est connue parfaitement. A partir de ces transformations, on connaît alors le biais dû à la construction de mosaïque pour cette trajectoire.

Puis la mire est remplacée par un poster représentant un fond marin. La trajectoire choisie est réexécutée. On peut alors améliorer la construction de la mosaïque en déformant chaque image avant de rechercher les transformations inter-image.

NB : La solution proposée est valable uniquement en laboratoire, en utilisant une mire dont les motifs sont connus parfaitement.

7.3 Transformations spatiales

La transformation spatiale est l'étape de base du warping. Elle relie un système de coordonnées à un autre. Cela peut être une transformation matricielle (rotation, translation, zoom, ...), analytique (polynomiale, ...) ou qui repose sur la mise en correspondance de deux ensembles de points de contrôle (voir ci-après section 7.5).

Définitions Une transformation spatiale définit une relation entre deux systèmes de coordonnées (relatif à deux images). La fonction qui relie ces deux systèmes de coordonnées peut être exprimée de deux façons : soit elle relie le système de coordonnées de l'image de sortie avec celui de l'image d'entrée, soit l'inverse.

$$[x, y] = [X(u, v), Y(u, v)] \quad (7.1)$$

$$[u, v] = [U(x, y), V(x, y)] \quad (7.2)$$

où $[u, v]$ est la coordonnée dans l'image d'entrée, correspondant au pixel observée (image de sortie) $[x, y]$. X, Y, U, V sont les fonctions qui définissent la transformation spatiale. X et Y représentent le "forward-mapping" (relie image d'entrée à image de sortie), tandis que U et V représentent le "inverse-mapping".

Transformations matricielles De nombreuses transformations géométriques peuvent s'exprimer grâce à une matrice T_1 3×3 . Celle-ci peut prendre en compte le changement d'échelle, la rotation, la translation et la perspective en 2D.

$$[x', y', w'] = [u, v, w] T_1 \quad (7.3)$$

avec

$$T_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (7.4)$$

Cette matrice peut se décomposer en quatre sous-matrices qui correspondent chacune à une transformation particulière.

$$T_2 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (7.5)$$

correspond à une transformation linéaire pour le changement d'échelle et la rotation. La matrice $[a_{31} a_{32}]$ représente la translation, $[a_{13} a_{23}]^T$ la perspective. L'élément a_{33} représente le changement d'échelle global.

Transformation bilinéaire La représentation générale d'une transformation bilinéaire est la suivante :

$$[x, y] = [uv, u, v, 1] \begin{bmatrix} a_3 & b_3 \\ a_2 & b_2 \\ a_1 & b_1 \\ a_0 & b_0 \end{bmatrix} \quad (7.6)$$

Il existe également d'autres types de transformations comme les transformations polynômiales, qui sont utilisées pour inverser une distorsion inconnue. On peut utiliser des polynômes par morceaux, des splines, etc...

7.4 La déformation d'images en pratique

Une image numérique est par construction discrète. Il faut donc prendre en compte le fait que les transformations géométriques sont appliquées à l'image dans le plan continu.

L'image acquise est donc interpolée (c'est la phase dite d'"image reconstruction"), la transformation est appliquée sur l'image interpolée. Enfin, on rééchantillonne l'image continue transformée (phase dite d'"image resampling").

7.5 Application du warping à la déformation de la mosaïque

Nous allons exposer ci-après la méthode imaginée pour fusionner la mosaïque avec la localisation AR/(C,V). Des premiers résultats seront présentés, utilisant une séquence d'images déjà acquise pour la mosaïque. En effet, nous n'avons pas eu à notre disposition un jeu de données réelles complet (séquence d'images et mesures de cap et vitesse pour une même plongée).

7.5.1 Principe de base de la méthode

Les techniques de warping sont très utilisées dans le cinéma, notamment dans des scènes de morphing, telles la transformation du visage (loup-garou), transformation d'un visage en un autre (clip "black or white" de Michael Jackson).

C'est le cas de la méthode proposée par Beier et Neely [BN92], qui permet de déformer une image à partir d'une paire de segments correspondants, se situant l'un dans l'image dite "source" et l'autre dans l'image dite "de destination". Dans tout ce qui suit, l'image *source* représente l'image acquise dont on veut déformer

les segments, l'image de *destination* représente l'image de référence, dont les segments sont utilisés pour déformer les segments de l'image source.

Le principe de cette méthode est décrit dans le tableau 7.1 et est illustré sur les figures 7.2.

| |
|---|
| <p>Pour chaque pixel X de l'image de destination :</p> <p>Calculer u et v, selon ^a :</p> $u = \frac{(X-P) \cdot (Q-P)}{\ Q-P\ ^2} \quad (7.7)$ $v = \frac{(X-P) \cdot (Q-P)^\perp}{\ Q-P\ }$ <p>Calculer la position du pixel X' dans l'image acquise pour ces u, v, selon</p> $X' = P' + u \cdot (Q' - P') + \frac{v \cdot (Q' - P')^\perp}{\ Q' - P'\ } \quad (7.8)$ <p>$DestinationImage(X) = SourceImage(X')$</p> <hr/> <p>^a$V^\perp$ est le vecteur perpendiculaire au vecteur V et de même norme.</p> |
|---|

TAB. 7.1 – Algorithme de déformation pour une seule paire de segments.

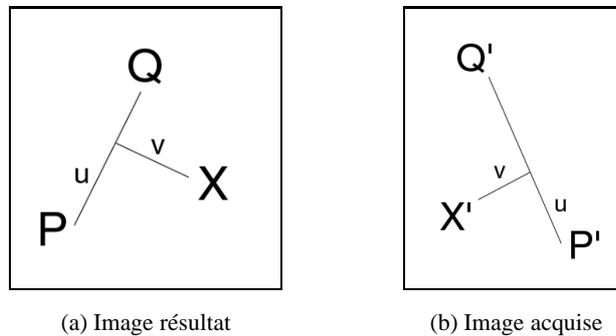


FIG. 7.2 – Principe du warping.

Nous pouvons remarquer que pour obtenir :

- une translation de l'images, il suffit que $\overrightarrow{P'Q'} = \overrightarrow{PQ} + \vec{n}$.
- une rotation de l'image, il suffit que $\|Q' - P'\| = \|Q - P\|$ et $(\widehat{\overrightarrow{P'Q'}, \overrightarrow{PQ}}) = \theta$.
- un changement d'échelle, il suffit que $\overrightarrow{P'Q'} = k \cdot \overrightarrow{PQ}$ avec $k \in \mathbb{R}$.

Sur la figure 7.3 page suivante est représentée en haut à gauche un "F" initial, avec le segment par lequel on veut transformer l'image. Sur les autres figures sont représentés quelques exemples de déformations (rotation, translation, changement d'échelle) à partir de la correspondance entre le segment du "F" initial et le segment tel qu'on a voulu le transformer (arête verticale du "F"). On constate que le "F" n'est pas déformé.

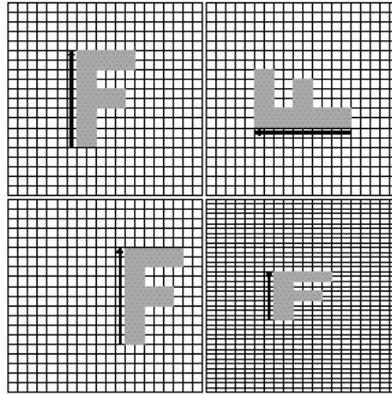


FIG. 7.3 – Exemples de déformations à partir d'une paire de segments.

7.5.2 Extension à plusieurs couples de segments

Il peut être intéressant de tenir compte de plusieurs paires de segments. C'est le cas lors du morphing de deux visages : on sélectionne par exemple les deux arêtes du nez et on les met en correspondance. Dans ce cas, la méthode est quasiment la même que pour un seul segment. On introduit simplement un poids supplémentaire pour chaque segment, qui dépend de la longueur du segment et de l'éloignement du point considéré au segment.

Le principe en est détaillé dans [BN92]. Soit X_i la position du pixel considéré. Une position est X'_i est calculée pour chaque segment. Le déplacement $D_i = X'_i - X$ est la différence entre les positions du pixel dans l'image source et l'image destination. Une moyenne pondérée de ces déplacements est calculée. Chaque poids est déterminé en fonction de la distance entre X et le segment considéré. Le déplacement moyen est ajouté à la position du pixel X pour déterminer la position du pixel X' de l'image source.

On introduit les constantes a , b et p qui permettent de changer les effets relatifs des segments. Le paramètre *longueur* est la longueur d'un segment et *dist* est la distance du pixel au segment considéré. Le poids assigné à chaque segment est le plus important quand le pixel est exactement sur le segment.

$$poids = \left(\frac{longueur^p}{a + dist} \right)^b \quad (7.9)$$

En théorie, si a est positif mais proche de 0 et que la distance entre le pixel et le segment est nulle, le poids est infini. Avec a grand, l'utilisateur sait que les pixels sur le segment iront exactement où il veut. Pour des valeurs de a plus grandes, le résultat sera une déformation plus douce mais avec moins de contrôle. La variable b détermine à quel point le poids pour chaque segment diminue avec la distance. Si b est grand, chaque pixel sera affecté par le segment le plus proche et lui seul. Si b est nul, chaque pixel sera affecté de la même façon par tous les segments. Les valeurs de b les plus communément employées sont comprises entre 0.5 et 2. Les valeurs de p sont comprises dans l'intervalle $[0, 1]$. Si p est nul, tous les segments ont le même poids. S'il vaut 1, les segments les plus longs ont plus de poids que les segments plus petits.

L'algorithme est présenté sur le tableau 7.2.

| |
|--|
| <p>Pour chaque pixel X de l'image de destination :</p> $DSOM = (0, 0)$ $SommePoids = 0$ <p>Pour chaque segment $P_i Q_i$</p> <p>calculer u et v à partir de P_i et Q_i</p> <p>calculer X'_i à partir de u, v et $P'_i Q'_i$</p> <p>calculer le déplacement $D_i = X'_i - X_i$ pour ce segment</p> <p>$dist =$ distance entre X et $P_i Q_i$</p> $poids = \left(\frac{longueur^p}{a + dist} \right)^b$ $DSOM+ = D_i * poids$ $SommePoids+ = poids$ $X' = X + \frac{DSOM}{SommePoids}$ $DestinationImage(X) = SourceImage(X')$ |
|--|

TAB. 7.2 – Algorithme de déformation pour des segments multiples.

Sur la figure 7.4 sont représentés quelques exemples de déformations à partir d'une paire d'arêtes verticales et d'une paire d'arêtes horizontales du "F". On remarque une déformation du "F" et de la grille pixellique.

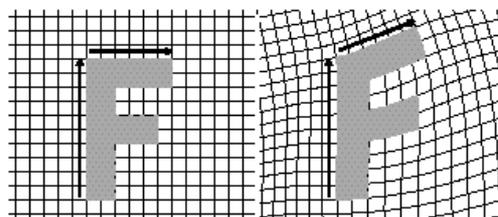


FIG. 7.4 – Exemples de déformations à partir de plusieurs paires de segments.

7.5.3 Déformation de la mosaïque à partir de la localisation AR/(C,V)

L'idée est de mettre en correspondance les transformations estimées par le RMR et les déplacements déduits de la localisation AR/(C,V) (trajectoire auto-estimée), afin de déformer la mosaïque déjà construite et considérée comme une image, pour la faire "coller" le mieux possible avec la localisation AR/(C,V), en laquelle nous avons davantage confiance [BJR04].

Ainsi, les P_i, Q_i sont les segments représentant les déplacements du centre de l'image, estimés par le RMR à partir de la séquence d'images seule. Les P'_i, Q'_i représentent les déplacements obtenus par la localisation AR/(C,V).

NB : La localisation AR/(C,V) est évaluée en mètres. Or, les techniques de warping se situent au niveau image, c'est-à-dire au niveau des pixels et non des mètres. Il convient donc de transformer les coordonnées en mètres en coordonnées dans l'image. Pour cela, nous utilisons la matrice fondamentale (voir partie 6.2.4), qui représente la relation pixel-mètre.

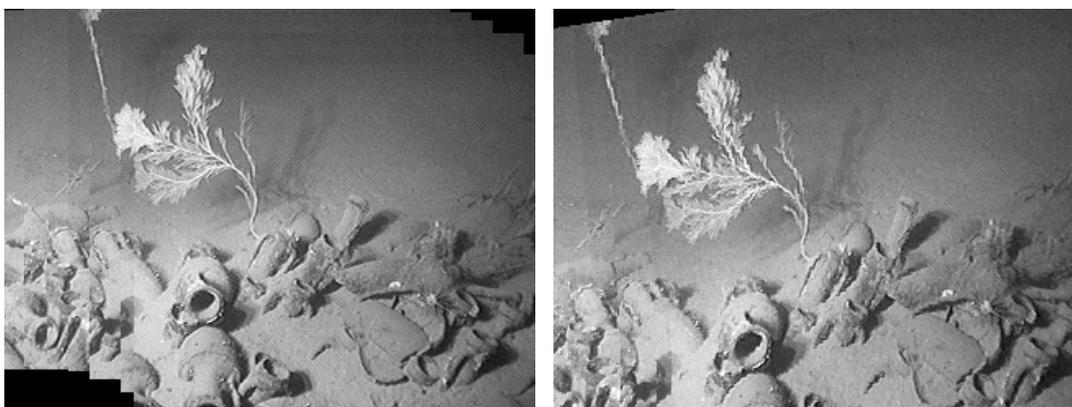
7.6 Expérimentations

Les expérimentations ont été réalisées sur une image de taille 324×248 , représentant une mosaïque de 7 images seulement et ceci, pour deux raisons. La première est due au fait que l'algorithme de déformation est très lent. En effet, le nombre de calculs effectués est très grand car la place de chaque pixel de l'image est recalculée à partir des distances à tous les segments de déformation. La déformation appliquée à la mosaïque a pris près de 3 minutes.

La deuxième raison est que dans le logiciel MATISSE®, la mosaïque est gérée à ce jour comme une grande image. Ainsi, les parties correspondant aux instants de rendez-vous se chevauchent et de l'information est perdue pour les images qui se retrouvent chevauchées. On ne peut donc pas appliquer l'algorithme de déformation multi-segments car de l'information est perdue. Ce problème pourrait être contourné si la mosaïque était gérée comme un objet et non comme une image. On aurait alors accès à l'information située en dessous, au niveau des instants de rendez-vous. C'est une évolution envisageable et envisagée pour MATISSE®.

7.6.1 Un exemple de déformation de la mosaïque

La figure 7.5(a) représente la mosaïque d'un champ d'amphores. Nous avons associé les déplacements inter-image (résultats de translation et rotation donnés par le RMR) à des segments aléatoires, représentant la trajectoire obtenue par la localisation AR/(C,V). Le résultat obtenu est représenté sur la figure 7.5(b). Nous pouvons constater que certaines parties ont été étirées, agrandies ou ont subi une rotation. L'exemple proposé a nécessité environ 3 minutes de calcul, pour un algorithme programmé en C++ (en Matlab c'est encore plus long) et exécuté sur un Pentium 4 à 1.6 GHz, avec 512Mo de RAM.



(a) Mosaïque initiale

(b) Mosaïque déformée

FIG. 7.5 – Exemple de déformation sur une mosaïque de 7 images.

Pourtant, la mosaïque a été formée à partir de seulement 7 images. Elle fait une taille relativement réduite : 324×248 pixels. La déformation ne doit traiter que 6 paires de segments (6 déplacements entre les 7 images). Aussi, pour une mosaïque d'environ 200 images et de taille 2400×2000 pixels comme celle construite dans le chapitre 5, la déformation prend en compte 200 paires de segments pour calculer la position de chaque pixel. Aussi, le temps de calcul pour effectuer la déformation est très élevé.

7.6.2 Etude des paramètres de la déformation à partir de plusieurs paires de segments

Une étude a été réalisée sur l'influence des différents paramètres a , b et p . L'image originale est la même que précédemment, elle représente une mosaïque de 7 images d'un champ d'amphores. Les segments '-*-' représentent les segments initiaux (translations inter-images déterminées par l'algorithme RMR) tandis que les segments '-□-' représentent les segments tels que désirés après déformation.

L'influence du paramètre a est illustrée sur la figure 7.6 page suivante qui représente la mosaïque initiale avec en surimposition, les paires de segments, et sur les 7.7 page suivante, qui représentent les résultats de la déformation obtenue pour $a = 1$ et $a = 100$. Les paramètres p et b sont fixés respectivement à 0 et 0.5. On ne remarque pas de différence flagrante. Toutefois, on peut remarquer que le passage de $a = 1$ à $a = 100$ se traduit par le fait que certains éléments sont étirés vers le haut ou vers le bas (par exemple au niveau de l'amphore dont on voit l'ouverture).

L'influence des paramètres p et b est illustrée sur la figure 7.8 page 161 qui représente la mosaïque initiale avec en surimposition, les paires de segments, et sur les 7.9 page 161, qui représentent les résultats de la déformation obtenue pour $p = 0$ ou 1 et $b = 0.5$ ou 2. Le paramètre a est fixé à 1. On peut remarquer tout d'abord en prenant comme référence l'image pour $p = 0$, $b = 0.5$, que l'amphore la plus longue paraît étirée lorsque $b = 2$. D'autre part, lorsque p passe de 0 à 1, on remarque que l'image paraît distendue vers le haut et vers le bas. Pour $b = 0.5$ ou 2, prendre $p = 1$ déforme encore plus l'image.

Il est très difficile de conclure à l'influence des différents paramètres de la déformation par l'étude des déformations d'une image donnée (la mosaïque ou toute autre image plus simple), car les déformations ne sont pas simples : ce ne sont ni des translations, ni des rotations, ni des transformations affines ou polynômiales. On peut seulement conclure sur la nécessité d'adapter les paramètres aux transformations que l'on veut appliquer à l'image, en suivant les remarques théoriques sur l'influence des paramètres, présentées dans la partie 7.5.2.

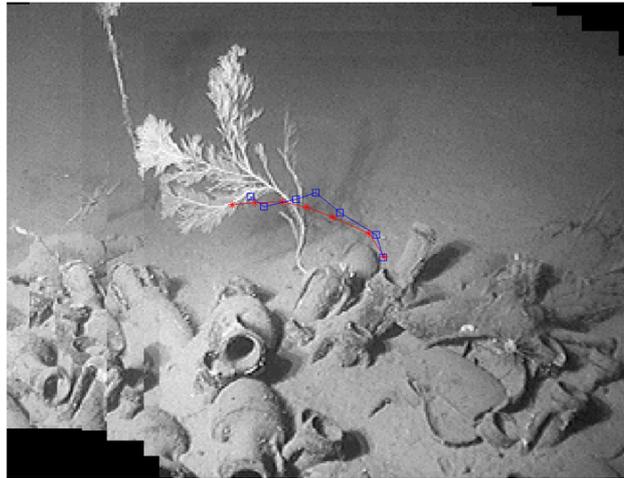
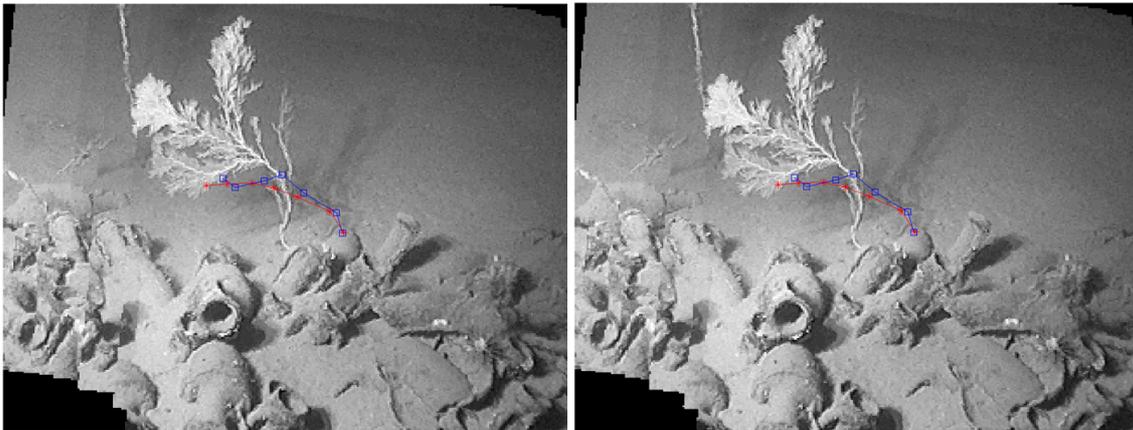


FIG. 7.6 – Image initiale, avec les paires de segments surexposés, utilisée pour l'étude du paramètre a .



(a) $a = 1, p = 0, b = 0.5$.

(b) $a = 100, p = 0, b = 0.5$.

FIG. 7.7 – Influence du paramètre a sur la déformation d'une image, pour $p = 0$ et $b = 0.5$.

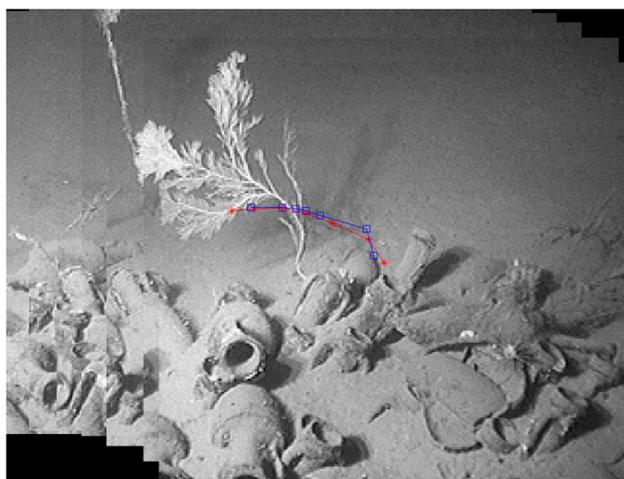


FIG. 7.8 – Image initiale, avec les paires de segments surexposés, utilisée pour l'étude des paramètres p et b .

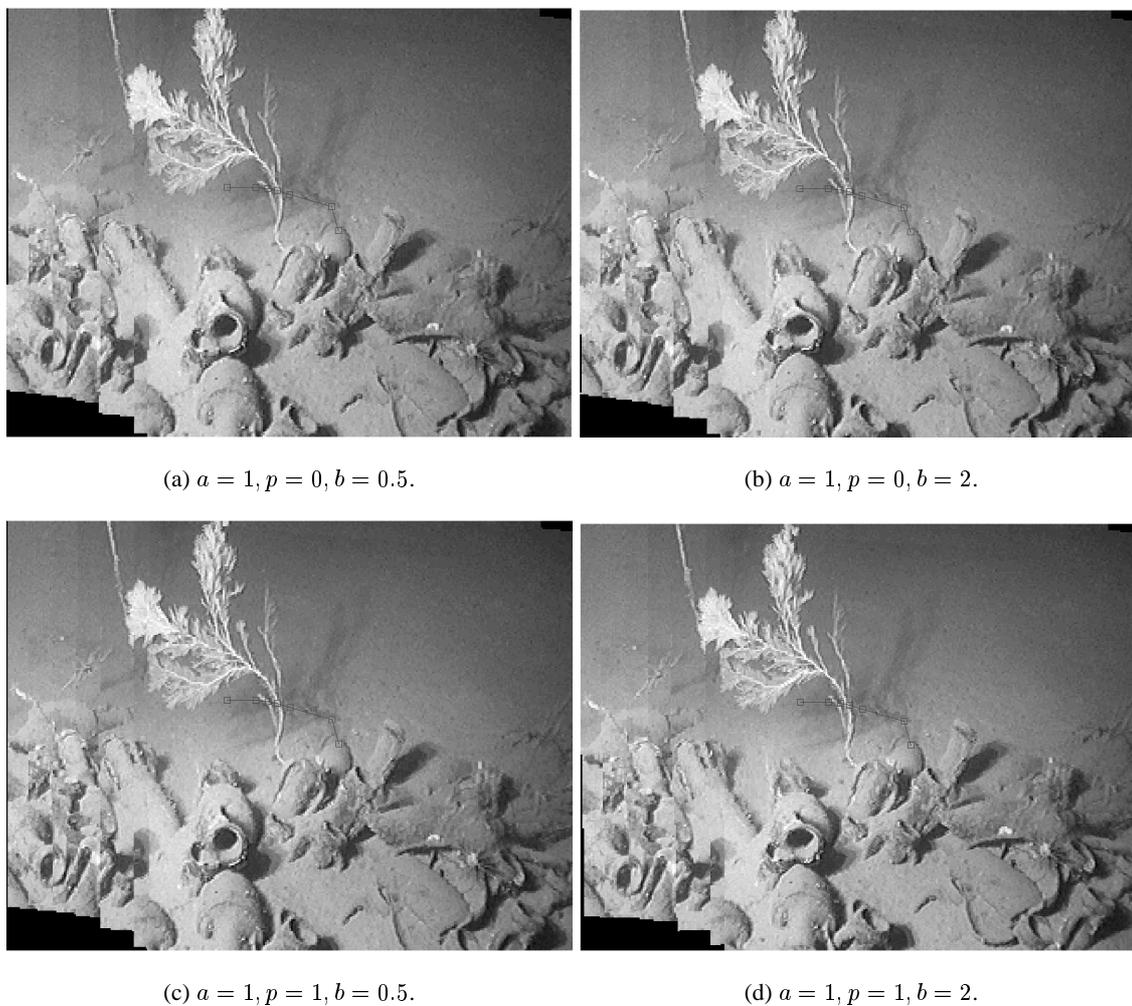


FIG. 7.9 – Influence des paramètres p et b sur la déformation d'une image, pour $a = 1$.

7.7 Conclusion

Dans ce chapitre, nous avons proposé une alternative à la fusion des localisations AR/(C,V) et AR/I, qui est réalisable sous réserve que les biais et variances des paramètres du RMR soient disponibles. Cette méthode consiste à déformer la mosaïque initialement construite, à partir des déplacements déduits de la localisation AR/(C,V).

Cette méthode est prometteuse car elle semble bien adaptée à la déformation d'une mosaïque d'images, au travers de la manipulation des segments reliant le centre de chaque imagerie constituant la mosaïque. Cependant, à l'heure actuelle, il existe des limitations, la principale étant un coût algorithmique très important.

Chapitre 8

Conclusion et perspectives

8.1 Récapitulatif : le système de correction de mosaïques

Dans cette partie, nous allons décrire le système complet d'amélioration de la mosaïque, à partir des données initiales, qui sont un jeu de mesures de cap et de vitesse, les positions des points de départ et d'arrivée, et une séquence d'images sous-marines. Ces recherches ont permis de fusionner deux systèmes (le système d'acquisition d'images et le système gyro-doppler) afin d'améliorer la localisation de la mosaïque.

Pour réaliser ce système, nous avons utilisé des algorithmes de construction de mosaïques pré-existants (le RMR [OB95] et le KLT [ST94], décrits dans respectivement les parties 5.4.3 et 5.4.2), pour estimer les positions de l'engin à partir de la séquence d'images. Puis nous avons étendu l'algorithme PISA [Sis00] de mise en correspondance d'images (décrit dans la partie 5.3.1) et avons mis au point une méthode originale d'estimation de trajectoire, l'auto-estimation [BRL03], décrite dans la partie 3.4. Nous avons également proposé une fusion des estimations des positions à partir des mesures de cap et vitesse et des instants de rendez-vous estimés (localisation AR/(C,V)) et à partir des images et des instants de rendez-vous (localisation AR/I). Une alternative à cette fusion a été proposée : il s'agit d'une déformation de la mosaïque.

Le synoptique du traitement complet des mesures et images pour l'amélioration de la mosaïque d'images est présenté figure 8.1 page suivante.

Pour réaliser la localisation A/(C,V), il suffit d'intégrer les caps et vitesses mesurés à partir de la position initiale de l'engin. A partir de cette localisation A/(C,V), on estime grossièrement les instants où la trajectoire se recoupe. Cela donne une première approximation des instants de rendez-vous.

Ceux-ci sont alors utilisés par la PISA pour estimer plus finement les couples d'instants de rendez-vous à partir des images. Le principe est le suivant (voir pour plus de détail voir partie 5.3.3) : un fenêtrage autour des couples est considéré. La transformation entre chaque couple d'images dans le fenêtrage est alors estimée par le biais de six méthodes possibles, dont quatre développées par Sistiaga (moindres carrés pondérés ou non, EKF pondérés ou non) et deux originales (moindres carrés linéaires pondérés ou non), basées sur des critères linéaires (l'apport de ces dernières méthodes est essentiellement leur rapidité et leur simplicité d'implémentation : pas de problème de minima locaux).

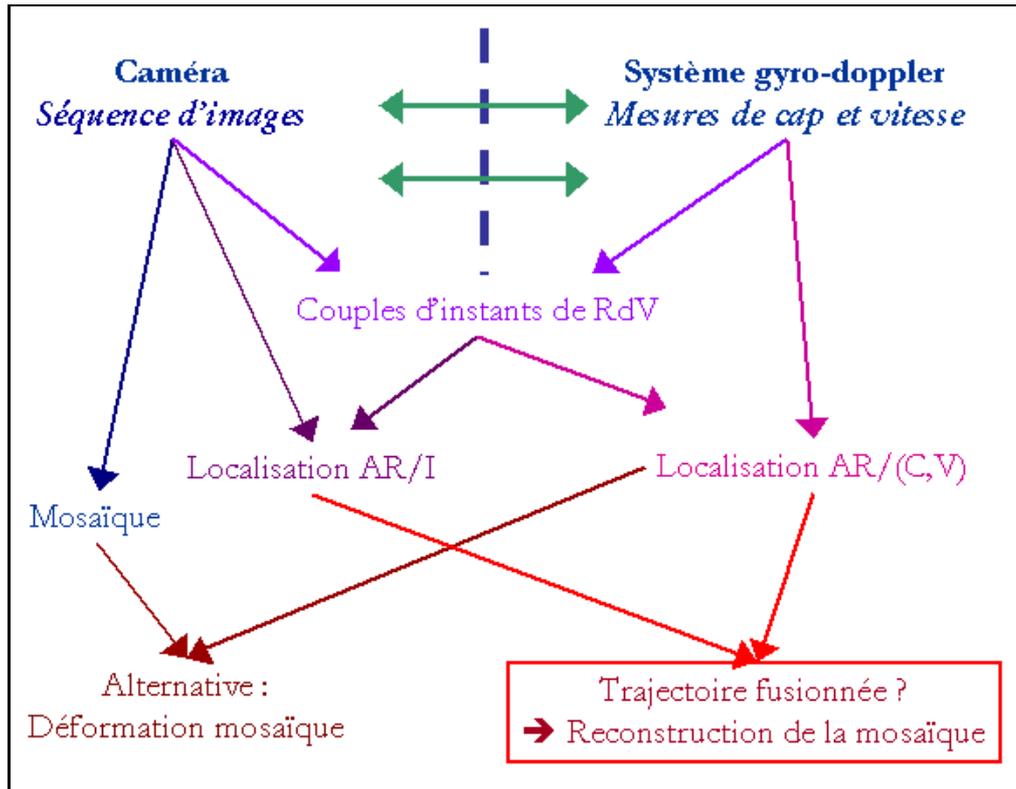


FIG. 8.1 – Synoptique de l'amélioration de mosaïques d'images proposée.

Les indices des images comportant la plus petite translation sont alors désignés comme un couple d'*instants de rendez-vous*.

A partir des instants de rendez-vous et des $N - 1$ mesures de cap et vitesse, on réalise une *localisation AR/(C,V)* par la méthode que nous avons proposée dans [BJR03], l'auto-estimation. Cette méthode repose sur le concept d'aller-retour et sur une estimation par les moindres carrés pondérés. Pour estimer la trajectoire à l'instant k , une trajectoire aller (mettant en jeu les caps et vitesses des instants 1 à k) et une trajectoire retour (mettant en jeu les mesures des instants $N - 1$ à $k + 1$) sont estimées. Après une réduction de biais, ces deux trajectoires estimées sont fusionnées par la méthode des moindres carrés. Les performances de cet estimateur sont un biais quasi-nul et des variances proches de celles calculées à partir de la Borne de Cramer-Rao.

D'autre part, deux stratégies de parcours de trajectoire pour inclure les informations données par les instants de rendez-vous ont été proposées. La meilleure stratégie consiste à considérer la trajectoire comme un graphe orienté (voir définition partie 3.5.2). L'algorithme repose sur un parcours de graphe et une recherche itérative des plus courts chemins dans ce graphe.

En parallèle, une *mosaïque d'images* est construite à partir des images de la séquence acquise, par les méthodes de KLT [ST94] et du RMR développée par Odobez et Bouthémy [OB95]. Une comparaison des deux méthodes a été effectuée sur plusieurs séquences d'images et il s'avère que le RMR est plus robuste (il

estime mieux la trajectoire) et plus rapide. Cette méthode repose sur les notions de flot optique et d'estimation multi-échelle (ou pyramidale). L'algorithme retourne une estimation des déplacements inter-image. Connaissant les positions extrêmes et les estimations des instants de rendez-vous, une **localisation AR/I** peut être réalisée. Comme pour la localisation AR/(C,V), on considère la trajectoire dans son ensemble (trajectoire complexe). Chaque trajectoire simple est obtenue en fusionnant (par une méthode des moindres carrés pondérés) d'une part, la localisation A/I déduite de l'intégration des déplacements estimés par le RMR et la position initiale, et d'autre part, la localisation R/I déduite de l'intégration rétrograde des déplacements estimés à partir de la position finale.

Une fusion des localisations AR/(C,V) et AR/I a été proposée pour l'amélioration de l'estimation de la trajectoire (moindres carrés pondérés). On remarque que les déplacements de la localisation AR/I sont très lisses alors que ceux de la localisation AR/(C,V) le sont moins (le bruit n'est pas éliminé lors de l'auto-estimation). En revanche, la position est mieux estimée par la localisation AR/(C,V). Aussi, l'idée était de fusionner les deux localisations AR/(C,V) et AR/I afin de tirer avantage de chacune. Néanmoins, la fusion que nous proposons est très simple et n'est pas optimale, en partie pour des raisons de non-connaissance de variances de certains paramètres liés à l'image.

Enfin, une alternative à cette fusion a été proposée. Elle vise à améliorer la qualité visuelle de la mosaïque d'images ainsi que sa localisation. Il s'agit d'une méthode de déformation d'images (ou warping) qui recalcule la position d'un pixel à partir d'une liste de couples de déplacements. Ces couples de déplacements sont fournis d'une part, par la trajectoire estimée au mieux (en l'occurrence, la trajectoire fusionnée s'il en existe une, sinon, la localisations AR/(C,V)) et d'autre part, par les déplacements inter-image estimés par la méthode de construction de mosaïques (localisation A/I). Cette méthode a été mise en jeu sur l'exemple d'une mosaïque de 7 images seulement, essentiellement pour des contraintes de coût calculatoire. Il serait intéressant d'obtenir un résultat sur une mosaïque entière, sur un ordinateur assez puissant pour réaliser les calculs.

8.2 Discussion et perspectives

Les résultats que nous avons présentés dans les chapitres précédents reposent sur plusieurs hypothèses. La première est l'hypothèse consistant à connaître les positions d'arrivée et de départ de l'engin lors de l'exploration. Cette hypothèse peut être avérée par la pose d'un amer artificiel au fond marin, ce qui est réalisable opérationnellement.

La deuxième hypothèse consiste à considérer que le véhicule parcourt une trajectoire à profondeur constante. Cette hypothèse était nécessaire pour initier la problématique d'amélioration de mosaïques. Elle est très simplificatrice puisqu'elle permet de raisonner en deux dimensions au lieu de trois. Malheureusement elle n'a pas ou peu lieu d'être en conditions réelles. Il reste donc du travail à réaliser dans ce sens.

La troisième hypothèse a été que nous ne prenions pas en considération l'orientation de la caméra (tan-

gage et roulis) lors des prises de vue. Or ces effets ne sont pas négligeables (ils changent notamment la taille du pixel exprimée mètres) et influencent la construction de la mosaïque. Il serait donc bienvenu de tenir compte de ces effets.

D'autre part, toutes les méthodes que nous avons développées n'ont pas été réfléchies pour être appliquées en temps réel. En effet, l'objectif de ces recherches est d'améliorer la qualité globale d'une carte visuelle du fond marin. Pour cela, il n'est pas utile que les méthodes soient temps réel. De plus, ceci peut être réalisé plus efficacement si toutes les données externes aux images sont prises en compte.

Enfin, les résultats présentés pour les méthodes développées dans ce manuscript ont été obtenus sur des jeux de données acquis en bassin intérieur. Il serait intéressant de faire des expérimentations sur un jeu de données complet (caps, vitesses, images) acquis lors de missions en mer.

Ces travaux de recherche ont initié l'amélioration de la localisation de mosaïques d'images en proposant une méthode combinant des données de trajectographie (mesures de cap et de vitesse) et une séquence d'images, en tenant compte de la géométrie particulière d'exploration de la zone d'intérêt.

Bibliographie

- [AB86] H. Asada and M. Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1) :2–14, January 1986.
- [ABO⁺04] A.-G. Allais, M. Borgetto, J. Opderbecke, N. Pessel, and V. Rigaud. Seabed video mosaicking with matisse : a technical overview and cruise results. In *Proceedings of the 14th ISOPE Conference*, pages 417–421, Toulon, May 2004.
- [Ade98] H.G. Adelman. Butterworth equations for homomorphic filtering of images. *Computers in Biology and Medicine*, 28 :169–181, 1998.
- [Alt03] Yucel Altunbasak. A fast parametric motion estimation algorithm with illumination and lens distortion correction. *IEEE Transactions on Image Processing*, 12(4) :395–408, April 2003.
- [Aya03] N. Ayache. Epidaure : a research project in medical image analysis, simulation and robotics at inria. *IEEE Trans. on Medical Imaging*, 22(10) :1185–1201, October 2003.
- [Bea78] P. R. Beaudet. Rotationally invariant image operators. In *4th International Joint Conference on Pattern Recognition, Kyoto, Japan*, pages 578–583, 1978.
- [BEFW97] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe. Mobile robot positioning - sensors and techniques. *Journal of Robotic Systems, special issue on mobile robots*, 14(4) :231–249, April 1997. Invited paper.
- [Ber83] Claude Berge. *Graphes*. Gauthier-Villars, 1983.
- [BJR03] Manon Borgetto, Claude Jauffret, and Vincent Rigaud. Auto-localisation d’un véhicule sous-marin quadrillant une zone. In *Actes du GRETSI 2003, Paris*, Septembre 2003.
- [BJR04] M. Borgetto, C. Jauffret, and V. Rigaud. Underwater image mosaic correction. In *Proceedings of the 14th ISOPE Conference*, volume II, pages 427–432, Toulon, France, May 2004.
- [BN92] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *Proc. of the 19th SIGGRAPH*, pages 35–42. ACM Press, 1992.
- [BPDR04] F. Bretar, M. Pierrot-Desseilligny, and M. Roux. Fusion lidar/photogrammétrie : Une mise en cohérence globale. In *Actes du 14^{ème} Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, France, Janvier 2004.

- [BRL03] M. Borgetto, V. Rigaud, and J.F. Lots. Lighting correction for underwater mosaicking enhancement. *Proceedings of the 16th International Conference on Vision Interface, Halifax, Canada*, June 2003.
- [Bro92] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4) :325–376, 1992.
- [Bro94] N.A. Brokloff. Matrix algorithm for doppler sonar navigation. In *Proceedings of the IEEE OCEAN's conference*, volume III, pages 378–383, 1994.
- [BS71] R. Bernstein and H. Silverman. Digital techniques for earth resource image data processing. In *Proceedings of the american institute of aeronautics and astronautics 8th annual meeting*, volume C21. AIAA, october 1971.
- [BS00] Jan C. Becker and Andreas Simon. Sensor and Navigation Data Fusion for an Autonomous Vehicle. In *Procs. IEEE Intelligent Vehicles Symposium 2000*, pages 156–161, Detroit, USA, october 2000.
- [BSL93] Yaakov Bar-Shalom and Xiao-Rong Li. *Estimation and Tracking, Principles, Techniques and Software*. Artech House, 1993.
- [BT02] Camille Béchaz and Hubert Thomas. Gib portable tracking systems - the underwater use of gps. *Hydro INTERNATIONAL*, 6(8), October 2002.
- [Cal04] O. Le Calvé. Propriétés physiques du milieu marin. Cours en ligne d'introduction à l'océanographie physique, May 2004. <http://isitv.univ-tln.fr/lecalve/oceano/plan.htm>.
- [Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8 :679–698, 1986.
- [CC87] C.K. Chui and G. Chen. *Kalman Filtering with Real-Time Applications*. Springer-Verlag, 1987.
- [CGH96] Irène Charon, Anne Germa, and Olivier Hudry. *Méthodes d'optimisation combinatoire*. Masson, 1996.
- [Cha95] R. Chatila. Deliberation and reactivity in autonomous mobile robots. *Robotics and Autonomous Systems*, 16 :197–211, 1995.
- [Cha02] Raja Chatila. Slam : representation issues. Présentation orale - SLAM Summer School, Stockholm, August 2002. CNRS/LAAS.
- [CZ96] E. Chong and S. Zak. *An introduction to ompimization*. John Wiley and Sons - New-York, 1996.
- [DCF⁺95] M. Devy, R. Chatila, P. Fillatreau, S. Lacroix, and F. Nashashibi. On autonomous navigation in a natural environment. *Robotics and Autonomous Systems*, 16 :5–16, 1995.
- [Dev03] F. Devernay. Vision par ordinateur. Cours dispensés au Mastère Photogrammétrie numérique, ENSG, 2001-2003.
- [DG93] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2) :101–124, 1993.

- [Dij59] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 :269–271, 1959.
- [DPNDWC01] G. Dissanayake, S. Clark P. Newman, H. Durrant-Whyte, and 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.
- [DSB96] M. Deffenbaugh, H. Schmidt, and J.G. Bellingham. Acoustic positioning in a fading multipath environment. In *Proceedings of the MTS/IEEE OCEANS*, volume 2, pages 596–600, 1996.
- [DW02] Hugh Durrant-Whyte. Localisation, mapping and the simultaneous localisation and mapping problem. Présentation orale - SLAM Summer School, Stockholm, August 2002. Australian Center for Field Robotics, University of Sydney.
- [Edy02] C. Edy. Logiciel caraibes, recalage de la navigation regina. Dossier des spécifications fonctionnelles, IFREMER, 2002.
- [FB81] M.A. Fischler and R.C. Bolles. Random sample consensus : a paradigm for model fitting with application to image analysis and automated cartography. *Communication Association and Computing Machine*, 24(6) :381–395, 1981.
- [Gal02] Jean Gallice, editor. *Images de profondeur*. Hermes, 2002.
- [GBC01] R. Garcia, J. Batlle, and X. Cufi. A system to evaluate the accuracy of a visual mosaicking methodology. In *Proceedings of the MTS/IEEE OCEANS conference*, volume 4, pages 2570–2576, Honolulu, Hawaii, 2001.
- [GBCA01] R. Garcia, J. Batlle, X. Cufi, and J. Amat. Positioning an underwater vehicle through image mosaicking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2279–2784, Seoul, Korea, May 2001.
- [GBLHT02] D.A. Grejner-Brzezinska, R. Li, N. Haala, and C. Toth. Multi-sensor systems for land-based and airborne mapping : technology of the future ? *International Archives on Photogrammetry and Remote Sensing*, XXXIV(II) :157–162, August 2002.
- [GCP00] R. Garcia, X. Cufi, and L. Pacheco. Image mosaicking for estimating the motion of an underwater vehicle. In *Proceedings of the 5th IPAC Conference on Manoeuvring and Control of Marine Crafts (MCMC)*, Aalborg, Denmark, 2000.
- [GHM91] M.S. Grewal, V.D. Henderson, and R.S. Miyasako. Application of kalman filtering to the calibration and alignment of inertial navigation systems. *IEEE Transactions on Automatic Control*, 36(1) :4–13, January 1991.
- [GRC01] P. Gros, A. Remazeilles, and F. Chaumette. Les invariants : introduction et application à la vision et à la robotique. In *Actes des troisièmes journées nationales de la recherche en robotique*, Giens, France, October 2001.
- [GSV98] N. Gracias and J. Santos-Victor. Automatic mosaic creation of the ocean floor. In *Proc. of the IEEE OCEANS'98*, Nice, France, September 1998.
- [GSV99] N. Gracias and J. Santos-Victor. Trajectory reconstruction using mosaic registration, 1999.

- [GSV00] N. Gracias and J. Santos-Victor. Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding*, 79(1) :66–91, July 2000.
- [GW92] R.C. Gonzales and R.E. Woods. *Digital image processing*. Addison Wesley, 1992.
- [Har95] R. Hartley. In defence of the 8-points algorithm. In *Proceedings of the fifth International Conference on Computer Vision (ICCV)*, Cambridge, Massachusetts, June 1995.
- [HHN02] Yu Huang, T.S. Huang, and H. Niemann. Segmentation-based object tracking using image warping and kalman filtering. In *Proceedings of the 2002 International Conference on Image Processing*, volume 3, pages 601–604, June 2002.
- [Hil62] M.N. Hill. *The Sea - Ideas and Observations on Progress in teh Study of the seas*, volume 1. Interscience publishers, 1962.
- [HS88] C. Harris and M. Stephens. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference, Manchester, GB*, pages 147–151, August 1988.
- [JL04] I.-K. Jung and S. Lacroix. Cartographie haute résolution à partir d’images stéréoscopiques acquises à basse altitude. In *Actes du 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, France, Janvier 2004.
- [JO04] J. Jouffroy and J. Opderbecke. Underwater vehicle trajectory estimation using contracting pde observers. In *Proc. of the American Control Conference (ACC)*, Boston, Massachusetts, USA, 2004.
- [JSV01] Sjoerd van der Zwaan José Santos-Victor, Nuno Gracias. Using vision for underwater robotics : video mosaics and station keeping. 1st International Workshop on Underwater robotics for Sea Exploitation and Environmental Monitoring, Rio de Janeiro, Brasil, October 2001.
- [Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(D) :35–45, 1960.
- [KHE96] J. Kilian, N. Haala, and M. Englich. Capture and evaluation of airborne laser scanner data. *International Archives on Photogrammetry and Remote Sensing*, XXXI(B3) :383–388, 1996.
- [KHH95] D. Kang, H. Hashimoto, and F. Harashima. Position estimation for mobile robot using sensor fusion. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, volume 1, pages 271–276, Nagoya, Aichi, Japan, 1995.
- [KR82] L. Kitchen and A. Rosenfeld. Gray level corner detection. *Pattern Recognition Letters*, 1 :95–102, 1982.
- [KW02] J.C. Kinsey and L.L. Whitcomb. Towards in-situ calibration of gyro and doppler navigation sensors for precision underwater vehicle navigation. *Proceedings of the IEEE Conference on Robotics and Automation*, pages 4016–4023, May 2002. Washington DC.
- [KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331, 1988.

- [LBSF98] J.J. Leonard, A.A. Bennett, C.M. Smith, and H.J.S. Feder. Autonomous underwater vehicle navigation. Technical Memorandum 98-1, Massachusetts Institute of Technology (MIT), Marine Robotics Laboratory, 1998.
- [Lem04] Thomas Lemaire. Robotique mobile : Localisation et cartographie simultanées. In *Colloque Ecole Doctorale Informatique et Télécommunication (EDIT)*, mars 2004.
- [LK81] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop*, pages 121–130, 1981.
- [LLFZ03] M. Legris, K. Lebart, F. Fohanno, and B. Zerr. Les capteurs d'imagerie en robotique sous-marine : tendances actuelles et futures. *Traitement du signal*, 20(2) :137–164, 2003.
- [LPOR96] L. Lucido, B. Popescu, J. Opderbecke, and V. Rigaud. Segmentation of bathymetric profiles and terrain matching for underwater vehicle navigation. In *Proceedings of the second annual World Automation Conference*, Montpellier, France, May 1996.
- [Maj01] S. Majumder. *Sensor Fusion and Feature Based Navigation for Subsea Robots*. PhD thesis, School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, Australia, Australian Center for Field Robotics, August 2001.
- [MC99] P. R. S. Mendonça and R. Cipolla. A simple technique for self-calibration. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 500–505, Fort Collins, Colorado, 1999.
- [MH80] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London*, B(207) :187–217, 1980.
- [MRL94] R. Marks, S. Rock, and M. Lee. Real-time video mosaicking of the ocean floor. In *Proceedings of the IEEE Symposium on autonomous underwater vehicle technology*, July 1994.
- [MRL95] R. Marks, S. Rock, and M. Lee. Real-time video mosaicking of the ocean floor. *IEEE Journal of Oceanic Engineering*, 20(3) :229–241, July 1995.
- [MS98] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12) :1376–1381, December 1998.
- [MTKW03] S.T.M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam 2.0 : an improved particle filtering algorithm for simultaneous localization and mapping that probably converges. *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1151–1156, August 2003. Acapulco, Mexico.
- [NLR03] P.M. Newman, J.J. Leonard, and R.J. Rikoski. Towards constant-time slam on an autonomous underwater vehicle using synthetic aperture sonar. In *Proceedings of the International Symposium on Robotics Research*, Sienna, Italy, October 2003.
- [NX02] S. Negahdaripour and X. Xu. Mosaic-based positioning and improved motion-estimation methods for automatic navigation of submersible vehicles. *IEEE Journal of Oceanic Engineering*, 27(1) :79–99, January 2002.

- [OB95] J.-M. Odobez and P. Bouthémy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4) :348–365, December 1995.
- [Opd97] J. Opderbecke. At-sea calibration of a usbl underwater vehicle positioning system. *Proceedings of MTS/IEEE OCEANS Conference*, 1 :721–726, 1997.
- [Pes03] N. Pessel. *Auto-calibrage d'une caméra en milieu sous-marin*. Thèse de doctorat, Université Montpellier II, 2003.
- [ROSP01] V. Rigaud, J. Opderbecke, P. Siméoni, and C. Pitout. Real time operational mosaics with the deep sea rosvictor. In *Proceedings of the International Workshop on Underwater Robotics for Sea Exploitation and Environmental Monitoring - IARP 2001*, Rio de Janeiro, Brasil, October 2001.
- [RR04] S. Rolfes and M.-J. Rendas. Statistical snakes : application to tracking of benthic contours. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, September 2004.
- [RRCC03] D. Ribas, P. Ridao, M. Carreras, and X. Cufi. An ekf vision-based navigation of an uuv in a structured environment. In *6th IFAC Conference on Manoeuvring and Control of Marine Crafts*, Girona, Spain, September 2003.
- [RTS65] H.E. Rauch, F. Tung, and C.T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8) :1145–1450, August 1965.
- [SAN⁺04] L. Soler, N. Ayache, S. Nicolau, X. Pennec, C. Forest, H. Delingette, and J. Marescaux. Traitement d'images médicales pour la planification, la simulation et l'aide intra-opératoire des actes chirurgicaux. *La Revue de l'Electricité et de l'Electronique*, 2004.
- [SB97] S.M. Smith and J.M. Brady. Susan : a new approach to low level image processing. *International Journal of Computer Vision*, pages 45–78, 1997.
- [Sch98] Cordelia Schmid. Comparing and evaluating interest points. *Proceedings of the Sixth International Conference on Computer Vision, Bombay, India*, pages 230–235, January 4-7 1998.
- [Sis00] M. Sistiaga. *Navigation référencée images de terrain pour engins sous-marins*. PhD thesis, IFREMER/ LIRMM, Septembre 2000.
- [Smi97] S.M. Smith. Reviews of optic flow, motion segmentation, edge finding and corner finding. Technical Report TR97SMS1, Oxford Centre for Functional Magnetic Resonance Imaging of the Brain (FMRIB), Oxford, UK, 1997.
- [ST94] Jianbo Shi and Carlo Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition, Seattle*, June 1994.
- [Tar95] J.P. Tarel. Calibration radiométrique de caméra. Technical report, INRIA, Mars 1995.
- [VF01] B. Vik and T.I. Fossen. A nonlinear observer for gps and ins integration. In *Proc. of the 40th IEEE Conference on Decision and Control*, pages 2956–2961, Orlando, Florida, USA, December 2001.

- [VG99] A. Visser and F. Groen. Organisation and design of autonomous systems. Organisation and Design of Autonomous Systems Course, M.Sc., Faculty of Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, August 1999.
- [vL90] J. van Leeuwen. Graph algorithms. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science : Volume A : Algorithms and Complexity*, pages 525–631. Elsevier, Amsterdam, 1990.
- [VLB96] Jerome Vaganay, John J. Leonard, and James G. Bellingham. Outlier rejection for autonomous acoustic navigation. *Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, USA*, pages 2174–2181, April 1996.
- [WB01] Greg Welch and Gary Bishop. An introduction to the kalman filter. SIGGRAPH 2001, course 8, 2001.
- [WYS99] L.L. Whitcomb, D.R. Yoeger, and H. Singh. Combined doppler/lbl based navigation of underwater vehicles. In *Proc. of the 11th International Symposium on Unmanned Untethered Submersible Technology*, Durham, New Hampshire, USA, August 1999.
- [XN01] X. Xu and S. Negahdaripour. Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwater vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2759–2766, Seoul, Korea, May 2001.
- [YC03] Wonpil Yu and Tunkoo Chung. A calibration-free lens distortion correction method for low cost digital imaging. In *2003 International Conference on Image Processing*, volume 1, pages 813–816, Sept. 14-17 2003.
- [Zha93] Zhengyou Zhang. Le problème de la mise en correspondance : l'état de l'art. Rapport de recherche RR-2146, INRIA, Décembre 1993.
- [Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11) :1330–1334, 2000.
- [ZRH01] Z. Zhu, E. M. Riseman, and A. R. Hanson. Parallel-perspective stereo mosaics. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, pages 345–352, Vancouver, Canada, July 2001.

Annexe A

Détails du calcul des biais, variances et covariances

A.1 Démonstration de la formule (3.36) de la partie 3.4.4

Soit X une variable aléatoire gaussienne, d'espérance μ_X et de variance σ_X^2 . Pour calculer $\mathbf{E}(\sin X)$, on utilise la fonction caractéristique de X :

$$\phi_X(t) = \mathbf{E}(e^{itX}) = e^{\mu_X it - \frac{\sigma_X^2 t^2}{2}} \quad (\text{A.1})$$

si $X \sim G(\mu_X, \sigma_X^2)$. Évaluons la fonction caractéristique en $t = 1$, on obtient :

$$\mathbf{E}(e^{iX}) = \mathbf{E}(\cos X + i \sin X) = (\cos \mu_X + i \sin \mu_X) e^{-\frac{\sigma_X^2}{2}} \quad (\text{A.2})$$

Soit en prenant les parties réelle et imaginaire :

$$\mathbf{E}(\cos X) = \cos \mu_X e^{-\frac{\sigma_X^2}{2}} \quad (\text{A.3})$$

$$\mathbf{E}(\sin X) = \sin \mu_X e^{-\frac{\sigma_X^2}{2}} \quad (\text{A.4})$$

NB : L'hypothèse de gaussiennité des bruits additifs est indispensable pour effectuer ce calcul. Tous les calculs qui suivent reposent sur cette hypothèse.

A.2 Détails du calcul des biais, variances, covariances dans la navigation à l'estime

Pour mener ce calcul, nous allons tout d'abord supposer que les mesures de cap \tilde{q}_k et de vitesse \tilde{v}_k sont entachées d'un bruit gaussien centré.

$$\begin{aligned} \tilde{c}_k &= c_k + \varepsilon_{c_k} \\ \tilde{v}_k &= v_k + \varepsilon_{v_k} \end{aligned} \quad (\text{A.5})$$

Dans cette partie, nous allons dans un premier temps déterminer le biais de la position estimée à l'estime $\hat{X}_{k|k}^+$ (intégration simple des caps et vitesses). Puis nous décomposerons le calcul de la matrice de variance-covariance en établissant les termes diagonaux (variances des abscisses et ordonnées), puis en établissant les termes extra-diagonaux.

A.2.1 Calcul des biais

On rappelle ci-dessous les équations 3.31 de la navigation à l'estime (qui correspond à la navigation "aller" dans notre algorithme).

$$\begin{aligned}\hat{x}_{k|k}^+ &\triangleq x_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \sin \tilde{c}_{l-1} \\ \hat{y}_{k|k}^+ &\triangleq y_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \cos \tilde{c}_{l-1}\end{aligned}\tag{A.6}$$

Pour simplifier les notations, posons $\hat{X}_k = \hat{X}_{k|k}^+$, soit $(\hat{x}_k, \hat{y}_k)^T = (\hat{x}_{k|k}^+, \hat{y}_{k|k}^+)^T$. Rappelons la définition du biais :

$$b_{\hat{X}}(k) = X_k - \mathbf{E}(\hat{X}_k)\tag{A.7}$$

Soit encore

$$b_{\hat{X}}(k) = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} - \mathbf{E} \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} = \begin{bmatrix} b_x(k) \\ b_y(k) \end{bmatrix}\tag{A.8}$$

Le calcul du biais peut donc être effectué sur chaque composante indépendamment. C'est ce que nous allons présenter dans ce qui suit.

A.2.1.1 Espérance de \hat{x}_k et biais

$$\mathbf{E}(\hat{x}_k) = x_1 + \mathbf{E} \left(\sum_{l=2}^k (t_l - t_{l-1}) \tilde{v}_{l-1} \sin \tilde{c}_{l-1} \right)\tag{A.9}$$

Or, le cap et la vitesse sont indépendants puisque les mesures ne proviennent pas des mêmes capteurs. On a donc :

$$\mathbf{E}(\hat{x}_k) = x_1 + \sum_{l=2}^k (t_l - t_{l-1}) \mathbf{E}(\tilde{v}_{l-1}) \mathbf{E}(\sin \tilde{c}_{l-1})\tag{A.10}$$

Or $\mathbf{E}(\tilde{v}_{l-1}) = v_{l-1}$ et $\mathbf{E}(\sin \tilde{c}_{l-1}) = \sin c_{l-1} e^{-\sigma_c^2/2}$. D'où on obtient :

$$\mathbf{E}(\hat{x}_k) = x_1 + \sum_{l=2}^k (t_l - t_{l-1}) v_{l-1} \sin c_{l-1} e^{-\sigma_c^2/2}\tag{A.11}$$

Pour finir, le biais vaut

$$b_x(k) = x_k - \mathbf{E}(\hat{x}_k)\tag{A.12}$$

Soit

$$b_x(k) = \sum_{l=2}^k (t_l - t_{l-1}) v_{l-1} \sin c_{l-1} \left(1 - e^{-\sigma_c^2/2}\right) \quad (\text{A.13})$$

A.2.1.2 Espérance de \hat{y}_k et biais

$$\mathbf{E}(\hat{y}_k) = y_1 + \mathbf{E}\left(\sum_{l=2}^k (t_l - t_{l-1}) \tilde{v}_{l-1} \cos \tilde{c}_{l-1}\right) \quad (\text{A.14})$$

Nous faisons les mêmes calculs que pour $\hat{x}(k)$, en ajoutant que $\mathbf{E}(\cos \tilde{c}_{l-1}) = \cos c_{l-1} e^{-\sigma_c^2/2}$. D'où on obtient :

$$\mathbf{E}(\hat{y}_k) = y_1 + \sum_{l=2}^k (t_l - t_{l-1}) v_{l-1} \cos c_{l-1} e^{-\sigma_c^2/2} \quad (\text{A.15})$$

Pour finir, le biais vaut

$$b_y(k) = y_k - \mathbf{E}(\hat{y}_k) \quad (\text{A.16})$$

Soit

$$b_y(k) = \sum_{l=2}^k (t_l - t_{l-1}) v_{l-1} \cos c_{l-1} \left(1 - e^{-\sigma_c^2/2}\right) \quad (\text{A.17})$$

A.2.2 Calcul des variances

Dans cette partie, nous allons déterminer les termes diagonaux de la matrice de variance-covariance de \hat{X}_k .

A.2.2.1 Variance de x

$$\mathbf{Var}(\hat{x}_k) = \sum_{l=2}^k (t_l - t_{l-1})^2 \mathbf{Var}(\tilde{v}_{l-1} \sin \tilde{c}_{l-1}) \quad (\text{A.18})$$

Commençons par calculer chaque terme de la somme :

$$\mathbf{Var}(\tilde{v}_{l-1} \sin \tilde{c}_{l-1}) = \mathbf{E}(\tilde{v}_{l-1}^2 \sin^2 \tilde{c}_{l-1}) - \mathbf{E}(\tilde{v}_{l-1} \sin \tilde{c}_{l-1})^2 \quad (\text{A.19})$$

Calculons le premier terme de la soustraction :

$$\begin{aligned} \mathbf{E}(\tilde{v}_{l-1}^2 \sin^2 \tilde{c}_{l-1}) &= (v_{l-1}^2 + \sigma_v^2) \mathbf{E}(\sin^2 \tilde{c}_{l-1}) \\ &= (v_{l-1}^2 + \sigma_v^2) \mathbf{E}\left(\frac{1 - \cos 2\tilde{c}_{l-1}}{2}\right) \\ &= (v_{l-1}^2 + \sigma_v^2) \frac{1 - \cos 2c_{l-1} e^{-2\sigma_c^2}}{2} \end{aligned} \quad (\text{A.20})$$

Le deuxième terme de la soustraction est donné par :

$$\begin{aligned} \mathbf{E}(\tilde{v}_{l-1} \sin \tilde{c}_{l-1})^2 &= \mathbf{E}(\tilde{v}_{l-1})^2 \mathbf{E}(\sin \tilde{c}_{l-1})^2 \\ &= v_{l-1}^2 \sin^2 c_{l-1} e^{-\sigma_c^2} \\ &= v_{l-1}^2 \frac{1 - \cos 2c_{l-1}}{2} e^{-\sigma_c^2} \end{aligned} \quad (\text{A.21})$$

Donc l'équation A.19 revient à :

$$\begin{aligned} \mathbf{Var}(\tilde{v}_{l-1} \sin \tilde{c}_{l-1}) &= (v_{l-1}^2 + \sigma_v^2) \frac{1 - \cos 2c_{l-1} e^{-2\sigma_c^2}}{2} - v_{l-1}^2 \frac{1 - \cos 2c_{l-1}}{2} e^{-\sigma_c^2} \\ &= \frac{v_{l-1}^2}{2} \left(1 - e^{-\sigma_c^2} - \cos 2c_{l-1} e^{-2\sigma_c^2} + \cos 2c_{l-1} e^{-\sigma_c^2} \right) + \frac{\sigma_v^2}{2} \left(1 - \cos 2c_{l-1} e^{-2\sigma_c^2} \right) \\ &= \frac{v_{l-1}^2}{2} \left(1 - e^{-\sigma_c^2} + 2 \cos 2c_{l-1} e^{-\frac{3\sigma_c^2}{2}} \sinh \frac{\sigma_c^2}{2} \right) + \frac{\sigma_v^2}{2} \left(1 - \cos 2c_{l-1} e^{-2\sigma_c^2} \right) \end{aligned} \quad (\text{A.22})$$

Soit, après sommation de tous les termes :

$$\mathbf{Var}(\hat{x}_k) = \sum_{l=2}^k \left[\frac{v_{l-1}^2}{2} \left(1 - e^{-\sigma_c^2} + 2 \cos 2c_{l-1} \sinh \frac{\sigma_c^2}{2} e^{-3\sigma_c^2/2} \right) + \frac{\sigma_v^2}{2} \left(1 - \cos 2c_{l-1} e^{-2\sigma_c^2} \right) \right] (t_l - t_{l-1})^2 \quad (\text{A.23})$$

A.2.2.2 Variance de y

$$\mathbf{Var}(\hat{y}_k) = \sum_{l=2}^k (t_l - t_{l-1})^2 \mathbf{Var}(\tilde{v}_{l-1} \cos \tilde{c}_{l-1}) \quad (\text{A.24})$$

Commençons par calculer chaque terme de la somme :

$$\mathbf{Var}(\tilde{v}_{l-1} \cos \tilde{c}_{l-1}) = \mathbf{E}(\tilde{v}_{l-1}^2 \cos^2 \tilde{c}_{l-1}) - \mathbf{E}(\tilde{v}_{l-1} \cos \tilde{c}_{l-1})^2 \quad (\text{A.25})$$

Calculons le premier terme de la soustraction :

$$\begin{aligned} \mathbf{E}(\tilde{v}_{l-1}^2 \cos^2 \tilde{c}_{l-1}) &= (v_{l-1}^2 + \sigma_v^2) \mathbf{E}(\cos^2 \tilde{c}_{l-1}) \\ &= (v_{l-1}^2 + \sigma_v^2) \mathbf{E}\left(\frac{1 + \cos 2\tilde{c}_{l-1}}{2}\right) \\ &= (v_{l-1}^2 + \sigma_v^2) \frac{1 + \cos 2c_{l-1} e^{-2\sigma_c^2}}{2} \end{aligned} \quad (\text{A.26})$$

Le deuxième terme de la soustraction est donné par :

$$\begin{aligned} \mathbf{E}(\tilde{v}_{l-1} \cos \tilde{c}_{l-1})^2 &= \mathbf{E}(\tilde{v}_{l-1})^2 \mathbf{E}(\cos \tilde{c}_{l-1})^2 \\ &= v_{l-1}^2 \cos^2 c_{l-1} e^{-\sigma_c^2} \\ &= v_{l-1}^2 \frac{1 + \cos 2c_{l-1}}{2} e^{-\sigma_c^2} \end{aligned} \quad (\text{A.27})$$

Donc l'équation A.25 revient à :

$$\begin{aligned} \mathbf{Var}(\tilde{v}_{l-1} \cos \tilde{c}_{l-1}) &= (v_{l-1}^2 + \sigma_v^2) \frac{1 + \cos 2c_{l-1} e^{-2\sigma_c^2}}{2} - v_{l-1}^2 \frac{1 + \cos 2c_{l-1}}{2} e^{-\sigma_c^2} \\ &= \frac{v_{l-1}^2}{2} \left(1 - e^{-\sigma_c^2} - 2 \cos 2c_{l-1} e^{-\frac{3\sigma_c^2}{2}} \sinh \frac{\sigma_c^2}{2} \right) + \frac{\sigma_v^2}{2} \left(1 + \cos 2c_{l-1} e^{-2\sigma_c^2} \right) \end{aligned} \quad (\text{A.28})$$

Soit, après sommation de tous les termes :

$$\mathbf{Var}(\hat{y}_k) = \sum_{l=2}^k \left[\frac{v_{l-1}^2}{2} \left(1 - e^{-\sigma_c^2} - 2 \cos 2c_{l-1} \sinh \frac{\sigma_c^2}{2} e^{-3\sigma_c^2/2} \right) + \frac{\sigma_v^2}{2} \left(1 + \cos 2c_{l-1} e^{-2\sigma_c^2} \right) \right] (t_l - t_{l-1})^2 \quad (\text{A.29})$$

A.2.3 Calcul des covariances

Nous allons à présent déterminer la covariance de \hat{X}_k . Ci-dessus, nous avons calculé les termes diagonaux de la matrice. Calculons maintenant les termes extra-diagonaux de la matrice de variance-covariance. Ceux-ci ne sont a priori pas nuls car les termes \hat{x}_k et \hat{y}_k , déterminés à partir des mêmes mesures, ne sont pas indépendants.

$$\mathbf{Cov}(\hat{x}_k, \hat{y}_k) = \mathbf{E}(\hat{x}_k \cdot \hat{y}_k) - \mathbf{E}(\hat{x}_k) \mathbf{E}(\hat{y}_k) \quad (\text{A.30})$$

Calculons le premier terme de la soustraction :

$$\mathbf{E}(\hat{x}_k \cdot \hat{y}_k) = \mathbf{E} \left(\left(x_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \sin \tilde{c}_{l-1} \right) \cdot \left(y_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \cos \tilde{c}_{l-1} \right) \right) \quad (\text{A.31})$$

$$\mathbf{E}(\hat{x}_k \cdot \hat{y}_k) = \mathbf{E} \left(\begin{aligned} &x_1 \cdot \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \cos \tilde{c}_{l-1} + y_1 \cdot \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \sin \tilde{c}_{l-1} + x_1 y_1 \\ &+ \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \sin \tilde{c}_{l-1} \cdot \sum_{l=2}^{l=k} (t_l - t_{l-1}) \tilde{v}_{l-1} \cos \tilde{c}_{l-1} \end{aligned} \right) \quad (\text{A.32})$$

$$\begin{aligned}
\mathbf{E}(\hat{x}_k \cdot \hat{y}_k) &= x_1 \sum_{l=2}^{l=k} \mathbf{E}(\tilde{v}_{l-1} \cos \tilde{c}_{l-1})(t_l - t_{l-1}) + y_1 \cdot \sum_{l=2}^{l=k} \mathbf{E}(\tilde{v}_{l-1} \sin \tilde{c}_{l-1})(t_l - t_{l-1}) \\
&+ x_1 y_1 + \sum_{l=2}^{l=k} \sum_{n=1}^{n=k} \mathbf{E}(\tilde{v}_{l-1} \sin \tilde{c}_{l-1} \tilde{v}_{n-1} \cos \tilde{c}_{n-1})(t_l - t_{l-1})(t_n - t_{n-1})
\end{aligned} \tag{A.33}$$

$$\begin{aligned}
\mathbf{E}(\hat{x}_k \cdot \hat{y}_k) &= x_1 \sum_{l=2}^{l=k} v_{l-1} \cos c_{l-1} \exp\left(-\frac{\sigma_c^2}{2}\right)(t_l - t_{l-1}) + y_1 \sum_{l=2}^{l=k} v_{l-1} \sin c_{l-1} e^{-\frac{\sigma_c^2}{2}}(t_l - t_{l-1}) \\
&+ x_1 y_1 + \sum_{l=2}^{l=k} \sum_{n=1}^{n=k} \mathbf{E}(\tilde{v}_{l-1} \tilde{v}_{n-1}) \mathbf{E}(\sin \tilde{c}_{l-1} \cos \tilde{c}_{n-1})(t_l - t_{l-1})(t_n - t_{n-1})
\end{aligned} \tag{A.34}$$

Pour calculer le dernier élément de la somme, considérons deux cas : $n = l$ ou $n \neq l$. Rappelons que les caps et vitesses sont indépendants dans leur ensemble et mutuellement indépendants. Cette propriété nous permet de terminer les calculs.

– $n = l$. Alors

$$\mathbf{E}(\tilde{v}_{l-1}^2) = \sigma_v^2 + v_{l-1}^2 \tag{A.35}$$

$$\text{car } \sigma_v^2 = \mathbf{E}(\tilde{v}_{l-1}^2) - \mathbf{E}(v_{l-1})^2.$$

$$\mathbf{E}(\sin \tilde{c}_{l-1} \cos \tilde{c}_{l-1}) = \mathbf{E}\left(\frac{1}{2} \sin 2\tilde{c}_{l-1}\right) = \frac{1}{2} \sin 2c_{l-1} \exp(-2\sigma_c^2) \tag{A.36}$$

– $n \neq l$. Alors

$$\mathbf{E}(\tilde{v}_{l-1} \tilde{v}_{n-1}) = \mathbf{E}(\tilde{v}_{l-1}) \mathbf{E}(\tilde{v}_{n-1}) = v_{l-1} v_{n-1} \tag{A.37}$$

$$\begin{aligned}
\mathbf{E}(\sin \tilde{c}_{l-1} \cos \tilde{c}_{n-1}) &= \mathbf{E}(\sin \tilde{c}_{l-1}) \mathbf{E}(\cos \tilde{c}_{n-1}) \\
&= \sin \tilde{c}_{l-1} e^{-\frac{\sigma_c^2}{2}} \cos \tilde{c}_{n-1} e^{-\frac{\sigma_c^2}{2}} \\
&= \sin \tilde{c}_{l-1} \cos \tilde{c}_{n-1} e^{-\sigma_c^2}
\end{aligned} \tag{A.38}$$

On obtient donc pour la dernière partie de la somme :

$$\begin{aligned}
(1) &= \left(x_1 \sum_{l=2}^{l=k} v_{l-1} \cos c_{l-1} (t_l - t_{l-1}) + y_1 \sum_{l=2}^{l=k} v_{l-1} \sin c_{l-1} (t_l - t_{l-1}) \right) e^{-\frac{\sigma_c^2}{2}} + x_1 y_1 \\
&+ \sum_{l=2}^{l=k} \sum_{n=1}^{n=k} (t_l - t_{l-1})(t_n - t_{n-1}) \begin{cases} \frac{1}{2} (v_{n-1}^2 + \sigma_v^2) \sin 2c_{n-1} e^{-2\sigma^2} & \text{pour } n = l \\ v_{l-1} v_{n-1} \sin c_{l-1} \cos c_{n-1} e^{-\sigma^2} & \text{pour } n \neq l \end{cases}
\end{aligned} \tag{A.39}$$

Calculons maintenant le deuxième terme de la covariance :

$$\begin{aligned}
\mathbf{E}(\hat{x}_k)\mathbf{E}(\hat{y}_k) &= \mathbf{E}\left(x_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1})\tilde{v}_{l-1} \sin \tilde{c}_{l-1}\right) \mathbf{E}\left(y_1 + \sum_{l=2}^{l=k} (t_l - t_{l-1})\tilde{v}_{l-1} \cos \tilde{c}_{l-1}\right) \\
&= x_1 \mathbf{E}\left(\sum_{l=2}^{l=k} (t_l - t_{l-1})\tilde{v}_{l-1} \cos \tilde{c}_{l-1}\right) + y_1 \mathbf{E}\left(\sum_{l=2}^{l=k} (t_l - t_{l-1})\tilde{v}_{l-1} \sin \tilde{c}_{l-1}\right) \\
&+ x_1 y_1 + \sum_{l=2}^{l=k} \sum_{n=1}^{n=k} \mathbf{E}(\hat{v}_{l-1} \sin \hat{c}_{l-1}) \mathbf{E}(\hat{v}_{n-1} \cos \hat{c}_{n-1}) (t_l - t_{l-1})(t_n - t_{n-1}) \\
&= x_1 \sum_{l=2}^{l=k} (t_l - t_{l-1}) v_{l-1} \cos c_{l-1} e^{-\frac{\sigma_c^2}{2}} + y_1 \sum_{l=2}^{l=k} (t_l - t_{l-1}) v_{l-1} \sin c_{l-1} e^{-\frac{\sigma_c^2}{2}} \quad (\text{A.40}) \\
&+ x_1 y_1 + \sum_{l=2}^{l=k} \sum_{n=1}^{n=k} v_{l-1} \sin c_{l-1} e^{-\frac{\sigma_c^2}{2}} v_{n-1} \cos c_{n-1} e^{-\frac{\sigma_c^2}{2}} (t_l - t_{l-1})(t_n - t_{n-1}) \\
&= \left(x_1 \sum_{l=2}^{l=k} (t_l - t_{l-1}) v_{l-1} \cos c_{l-1} + y_1 \sum_{l=2}^{l=k} (t_l - t_{l-1}) v_{l-1} \sin c_{l-1}\right) e^{-\frac{\sigma_c^2}{2}} \\
&+ x_1 y_1 + \sum_{l=2}^{l=k} \sum_{n=1}^{n=k} v_{l-1} \sin c_{l-1} v_{n-1} \cos c_{n-1} e^{-\sigma_c^2} (t_l - t_{l-1})(t_n - t_{n-1})
\end{aligned}$$

Tous les termes s'annulant, sauf les termes pour $n = l$, on obtient donc comme expression de la covariance :

$$\begin{aligned}
\mathbf{Cov}(\hat{x}_k, \hat{y}_k) &= \mathbf{E}(\hat{x}_k \hat{y}_k) - \mathbf{E}(\hat{x}_k) \mathbf{E}(\hat{y}_k) \\
&= \sum_{n=1}^{n=k} \frac{1}{2} (t_n - t_{n-1})^2 (v_{n-1}^2 + \sigma_v^2) \sin 2c_{n-1} e^{-2\sigma_c^2} - \frac{1}{2} v_{n-1}^2 \sin 2c_{n-1} e^{-\sigma_c^2} \quad (\text{A.41})
\end{aligned}$$

$$\boxed{\mathbf{Cov}(\hat{x}_k, \hat{y}_k) = \sum_{n=1}^{n=k} \frac{1}{2} (t_n - t_{n-1})^2 \left((v_{n-1}^2 + \sigma_v^2) e^{-2\sigma_c^2} - \frac{1}{2} v_{n-1}^2 e^{-\sigma_c^2} \right) \sin 2c_{n-1}} \quad (\text{A.42})$$

A.3 Biais, variances et covariances de l'estime après réduction du biais

Dans cette partie, nous allons présenter les résultats obtenus pour la trajectoire $\tilde{X}_{k|k}^+$, obtenue d'après la trajectoire à l'estime, après réduction du biais. Nous allons tout d'abord présenter la réduction de biais, puis nous expliciterons les calculs de variances et covariances, en fonction des résultats trouvés dans la partie précédente.

A.3.1 Réduction du biais

Pour débiaiser les trajectoires aller et retour, il suffit de remarquer que l'espérance est donnée par la formule :

$$\begin{aligned}\mathbf{E}(\hat{x}_{k|k}^+) &= x_1 + \sum_{l=2}^k \mathbf{E}(\tilde{v}_{l-1}) \mathbf{E}(\sin \tilde{c}_{l-1}) \\ &= x_1 + \sum_{l=2}^k v_{l-1} \sin c_{l-1} e^{-\sigma_c^2/2}\end{aligned}\quad (\text{A.43})$$

Il est alors possible de débiaiser simplement les abscisses estimées (on procède de la même façon pour les ordonnées) en posant

$$\tilde{x}_{k|k}^+ \triangleq e^{\frac{\sigma_c^2}{2}} \hat{x}_{k|k}^+ + (1 - e^{\frac{\sigma_c^2}{2}}) x_1 \quad (\text{A.44})$$

Il en va de même pour les ordonnées :

$$\tilde{y}_{k|k}^+ = e^{\frac{\sigma_c^2}{2}} \hat{y}_{k|k}^+ + \left(1 - e^{\frac{\sigma_c^2}{2}}\right) y_1 \quad (\text{A.45})$$

On obtient donc

$$\tilde{X}_{k|k}^+ = e^{\frac{\sigma_c^2}{2}} \hat{X}_{k|k}^+ + (1 - e^{\frac{\sigma_c^2}{2}}) X_1 \quad (\text{A.46})$$

A.3.2 Calcul des variances après correction du biais

D'après les expressions données au paragraphe précédent, on a pour la variance en x :

$$\begin{aligned}\mathbf{Var}(\tilde{x}_k) &= \mathbf{Var}\left((1 - e^{\frac{\sigma_c^2}{2}})x_1\right) + \mathbf{Var}\left(e^{\frac{\sigma_c^2}{2}}\hat{x}_k\right) \\ &= e^{\sigma_c^2} \mathbf{Var}(\hat{x}_k)\end{aligned}\quad (\text{A.47})$$

De même pour les variances en y :

$$\mathbf{Var}(\tilde{y}_k) = e^{\sigma_c^2} \mathbf{Var}(\hat{y}_k) \quad (\text{A.48})$$

A.3.3 Calcul de la covariance après correction du biais

La covariance pour le couple \tilde{x}_k, \tilde{y}_k est donnée par :

$$\mathbf{Cov}(\tilde{x}_k, \tilde{y}_k) = \mathbf{E}(\tilde{x}_k \tilde{y}_k) - \mathbf{E}(\tilde{x}_k) \mathbf{E}(\tilde{y}_k) \quad (\text{A.49})$$

Calculons la première partie de la soustraction :

$$\begin{aligned}\mathbf{E}(\tilde{x}_k \tilde{y}_k) &= \mathbf{E}\left[\left(e^{\frac{\sigma_c^2}{2}} \hat{x}_k + (1 - e^{\frac{\sigma_c^2}{2}})x_1\right) \left(e^{\frac{\sigma_c^2}{2}} \hat{y}_k + (1 - e^{\frac{\sigma_c^2}{2}})y_1\right)\right] \\ &= \mathbf{E}\left(e^{\sigma_c^2} \hat{x}_k \hat{y}_k\right) + \left(1 - e^{\frac{\sigma_c^2}{2}}\right) \left(y_1 \mathbf{E}(\hat{x}_k) e^{\frac{\sigma_c^2}{2}} + x_1 \mathbf{E}(\hat{y}_k) e^{\frac{\sigma_c^2}{2}}\right) + (1 - e^{\frac{\sigma_c^2}{2}})^2 x_1 y_1 \\ &= e^{\sigma_c^2} \mathbf{E}(\hat{x}_k \hat{y}_k) + (1 - e^{\frac{\sigma_c^2}{2}}) e^{\frac{\sigma_c^2}{2}} (x_1 \mathbf{E}(\hat{y}_k) + y_1 \mathbf{E}(\hat{x}_k)) + (1 - e^{\frac{\sigma_c^2}{2}})^2 x_1 y_1\end{aligned}\quad (\text{A.50})$$

La deuxième partie de la soustraction :

$$\begin{aligned} \mathbf{E}(\tilde{x}_k)\mathbf{E}(\tilde{y}_k) &= \mathbf{E}\left(e^{\frac{\sigma_c^2}{2}}\hat{x}_k + (1 - e^{\frac{\sigma_c^2}{2}})x_1\right)\mathbf{E}\left(e^{\frac{\sigma_c^2}{2}}\hat{y}_k + (1 - e^{\frac{\sigma_c^2}{2}})y_1\right) \\ &= e^{\sigma_c^2}\mathbf{E}(\hat{x}_k)\mathbf{E}(\hat{y}_k) + (1 - e^{\frac{\sigma_c^2}{2}})e^{\frac{\sigma_c^2}{2}}(x_1\mathbf{E}(\hat{y}_k) + y_1\mathbf{E}(\hat{x}_k)) + (1 - e^{\frac{\sigma_c^2}{2}})^2 x_1 y_1 \end{aligned} \quad (\text{A.51})$$

Après soustraction des deux termes, on en déduit finalement la covariance :

$$\mathbf{Cov}(\tilde{x}_k, \tilde{y}_k) = e^{\sigma_c^2} \mathbf{Cov}(\hat{x}_k, \hat{y}_k) \quad (\text{A.52})$$

A.4 Détails du calcul des biais, variances, covariances pour la trajectoire auto-estimée

Rappelons la formule de calcul de la trajectoire auto-estimée à l'instant k , $\tilde{X}_{k|N}$, à partir des aller et retour intégrés $\tilde{X}_{k|k}^+$ et $\tilde{X}_{k|N-k}^-$ obtenus après réduction de biais :

$$\tilde{X}_{k|N} \triangleq \left[\mathbf{P}\left(\tilde{X}_{k|k}^+\right)^{-1} + \mathbf{P}\left(\tilde{X}_{k|N-k}^-\right)^{-1} \right]^{-1} \cdot \left[\mathbf{P}\left(\tilde{X}_{k|k}^+\right)^{-1} \tilde{X}_{k|k}^+ + \mathbf{P}\left(\tilde{X}_{k|N-k}^-\right)^{-1} \tilde{X}_{k|N-k}^- \right] \quad (\text{A.53})$$

A.4.1 Biais de l'estimateur

$$\begin{aligned} b_{\tilde{X}_{k|N}} &= X_k - \mathbf{E}(\tilde{X}_{k|N}) \\ &= X_k - \left[\mathbf{P}\left(\tilde{X}_{k|k}^+\right)^{-1} + \mathbf{P}\left(\tilde{X}_{k|N-k}^-\right)^{-1} \right]^{-1} \cdot \left[\mathbf{P}\left(\tilde{X}_{k|k}^+\right)^{-1} \mathbf{E}\left(\tilde{X}_{k|k}^+\right) + \mathbf{P}\left(\tilde{X}_{k|N-k}^-\right)^{-1} \mathbf{E}\left(\tilde{X}_{k|N-k}^-\right) \right] \end{aligned} \quad (\text{A.54})$$

Soit, en réduisant au même dénominateur,

$$b_{\tilde{X}_{k|N}} = \left[\mathbf{P}\left(\tilde{X}_{k|k}^+\right)^{-1} + \mathbf{P}\left(\tilde{X}_{k|N-k}^-\right)^{-1} \right]^{-1} \cdot \left[\mathbf{P}\left(\tilde{X}_{k|k}^+\right)^{-1} b_{\tilde{X}_{k|k}^+}(k) + \mathbf{P}\left(\tilde{X}_{k|N-k}^-\right)^{-1} b_{\tilde{X}_{k|N-k}^-}(k) \right] \quad (\text{A.55})$$

Les biais $b_{\tilde{X}_{k|k}^+}(k)$ et $b_{\tilde{X}_{k|N-k}^-}(k)$ étant nuls, l'estimateur que nous proposons est non-biaisé.

A.4.2 Matrice de variance-covariance

Les deux positions aller et retour sont calculées à partir de caps et vitesses différents et mutuellement indépendants. Ces deux estimateurs $\tilde{X}_{k|k}^+$ et $\tilde{X}_{k|N-k}^-$ sont donc indépendants. Ainsi, la matrice de variance-

covariance, notée P , de $\tilde{X}_{k|N}$ est donnée par :

$$\begin{aligned}
P(\tilde{X}_{k|N}) &= \left[\mathbf{P} \left(\tilde{X}_{k|k}^+ \right)^{-1} + \mathbf{P} \left(\tilde{X}_{k|N-k}^- \right)^{-1} \right]^{-1} \cdot \left[\mathbf{P} \left(\tilde{X}_{k|k}^+ \right)^{-1} \mathbf{P} \left(\tilde{X}_{k|k}^+ \right) + \mathbf{P} \left(\tilde{X}_{k|N-k}^- \right)^{-1} \mathbf{P} \left(\tilde{X}_{k|N-k}^- \right) \right] \\
&= \left[\mathbf{P} \left(\tilde{X}_{k|k}^+ \right)^{-1} + \mathbf{P} \left(\tilde{X}_{k|N-k}^- \right)^{-1} \right]^{-1} \\
&= e^{\sigma_c^2} \left[\mathbf{P} \left(\hat{X}_{k|k}^+ \right)^{-1} + \mathbf{P} \left(\hat{X}_{k|N-k}^- \right)^{-1} \right]^{-1}
\end{aligned} \tag{A.56}$$

A.4.3 Exemple de calculs pour le cas d'une trajectoire en ligne droite

Calculons la matrice de variance-covariance dans le cas d'une droite, à vitesse et cap constants, pour $t_l - t_{l-1} = 1s$.

$$\mathbf{Var}(\hat{x}_{k|k}^+) = \sum_{l=2}^k \left[\frac{v^2}{2} \left(1 - e^{-\sigma_c^2} + 2 \cos 2c \sinh \frac{\sigma_c^2}{2} e^{-3\sigma_c^2/2} \right) + \frac{\sigma_v^2}{2} \left(1 - \cos 2ce^{-2\sigma_c^2} \right) \right] (t_l - t_{l-1})^2 = A(k-1) \tag{A.57}$$

avec A constante.

$$\mathbf{Var}(\hat{x}_{k|N-k}^-) = A(N-k) \tag{A.58}$$

avec A constante. Il en va de même pour l'ordonnée :

$$\mathbf{Var}(\hat{y}_{k|k}^+) = B(k-1) \tag{A.59}$$

avec B constante.

$$\mathbf{Var}(\hat{y}_{k|N-k}^-) = B(N-k) \tag{A.60}$$

La covariance du couple est donnée par :

$$\begin{aligned}
\mathbf{Cov}(\hat{x}_{k|k}^+, \hat{y}_{k|k}^+) &= \sum_{n=1}^{n=k} \frac{1}{2} (t_n - t_{n-1})^2 (v^2 + \sigma_v^2) \sin 2ce^{-2\sigma_c^2} - \frac{1}{2} v^2 \sin 2c_{n-1} e^{-\sigma_c^2} \\
&= \sum_{n=1}^{n=k} \frac{1}{2} (t_n - t_{n-1})^2 \left[(v^2 + \sigma_v^2) e^{-2\sigma_c^2} - \frac{1}{2} v^2 e^{-\sigma_c^2} \right] \sin 2c \\
&= C(k-1)
\end{aligned} \tag{A.61}$$

De même,

$$\mathbf{Cov}(\hat{x}_{k|N-k}^-, \hat{y}_{k|N-k}^-) = C(N-k) \tag{A.62}$$

D'où la matrice de variance-covariance de $\tilde{X}_{k|N}$:

$$\begin{aligned}
 P(\tilde{X}_{k|N}) &= e^{\sigma_c^2} \left(\begin{bmatrix} A(k-1) & C(k-1) \\ C(k-1) & B(k-1) \end{bmatrix}^{-1} + \begin{bmatrix} A(N-k) & C(N-k) \\ C(N-k) & B(N-k) \end{bmatrix}^{-1} \right)^{-1} \\
 &= e^{\sigma_c^2} \left((k-1)^{-1} \begin{bmatrix} A & C \\ C & B \end{bmatrix}^{-1} + (N-k)^{-1} \begin{bmatrix} A & C \\ C & B \end{bmatrix}^{-1} \right)^{-1} \quad (\text{A.63}) \\
 &= e^{\sigma_c^2} \begin{bmatrix} A & C \\ C & B \end{bmatrix} \frac{(k-1)(N-k)}{N}
 \end{aligned}$$

Remarquons que si l'on se restreint à une composante de la matrice $P(\tilde{X}_{k|N})$, la variance sur les abscisses par exemple, la variance est représentée par une courbe en cloche.

Annexe B

Critères de confiance de la PISA

B.1 Critère de confiance λ_k sur un appariement k

Pour déterminer la confiance à accorder à chaque appariement, Sistiaga introduit dans [Sis00] plusieurs coefficients dépendant des distances de Mahalanobis sur attributs et sur position ainsi que des seuils. L'appariement considéré est le k^{me} . Le premier coefficient est C_{ak} . Il est calculé à partir de la distance de Mahalanobis sur attributs δ_{ak}^2 et du seuil S_a utilisé (dédit de la loi du χ^2) dans la phase 1 de la PISA. Il traduit la ressemblance relative des vecteurs d'attributs décrivant les deux primitives du couple considéré.

On a

$$C_{ak} = \frac{S_a - \delta_{ak}^2}{S_a} \quad (\text{B.1})$$

Le deuxième coefficient C_{pk} est basé sur la distance de Mahalanobis sur les vecteurs de position δ_{pk}^2 et du seuil S_p . C_{pk} traduit la cohérence relative du couple par rapport à la transformation estimée. On a

$$C_{pk} = \frac{S_p - \delta_{pk}^2}{S_p} \quad (\text{B.2})$$

Le coefficient C_{mk} est basé sur les appariements multiples. En effet, un point de l'image I_1 peut être apparié avec plusieurs points de l'image I_2 . La confiance accordée au couple k est calculée de manière proportionnelle par rapport à l'ensemble des appariements. On a

$$C_{mk} = \frac{\Delta_{ak} - \delta_{ak}^2}{\Delta_{ak}} \quad (\text{B.3})$$

où Δ_{ak} est la somme des distances de Mahalanobis δ_{ak}^2 que le point de l'image I_1 possède avec tous ses n_k appariements. Plus un point a des appariements multiples, moins on fera confiance aux appariements le mettant en jeu. En revanche, si un point n'est apparié qu'une fois, la confiance est maximale. On définit la pondération suivante :

$$P_{mk} = \begin{cases} 1 & \text{si } n_k = 1 \\ \frac{C_{mk}}{n_k - 1} & \text{sinon} \end{cases} \quad (\text{B.4})$$

Enfin, Sistiaga définit le coefficient λ_k par :

$$\lambda_k = P_{mk} \exp (C_{ak} C_{pk} - 1) \quad (\text{B.5})$$

B.2 Critères de confiance pour la qualification de l'estimation

Afin de qualifier le résultat de l'estimation, Sistiaga définit deux critères de confiance sur les attributs et les position.

Critère de confiance sur les vecteurs d'attributs A partir de la liste finale des n appariements utilisés pour estimer la transformation, on définit le critère C_a selon :

$$C_a = 100 \exp \left(- \frac{\sum_{k=1}^n \delta_{ak}^2}{n S_a} \right) \quad (\text{B.6})$$

C_a est un pourcentage représentant l'écart moyen séparant les vecteurs d'attributs.

Critère de confiance sur les vecteurs de position De la même façon, est défini le critère C_p sur les vecteurs de position :

$$C_p = 100 \exp \left(- \frac{\sum_{k=1}^n \delta_{pk}^2}{n S_p} \right) \quad (\text{B.7})$$

Critère de qualification de l'estimation par PISA Le critère de confiance final C_r , également en pourcent, traduit la performance du résultat estimé par rapport aux critères internes de la PISA.

$$C_r = \frac{C_a + C_p}{2} \quad (\text{B.8})$$

Annexe C

Le logiciel MATISSE®

La vidéo sous-marine est présente sur la plupart des missions en mer, puisqu'elle peut permettre de contrôler des engins sous-marins, de mesurer des objets, ou de construire des mosaïques optiques utilisées par exemple pour la cartographie, la reconstruction d'épaves ou l'aide au pilotage.

Le logiciel MATISSE® [ROSP01, ABO⁺04] (Mosaïque Assistée par Traitement d'ImageS et de Sources Externes) permet de construire des mosaïques à partir de deux algorithmes, le KLT développé par [ST94] et le RMR mis au point à l'IRISA par [OB95].

Ce logiciel permet le réglage de plusieurs paramètres, dont l'estimation ou pas du zoom, de la rotation, l'estimation sur les images entières ou sur un fenêtrage de chaque image (paramètres en commun pour le KLT et le RMR), la taille de la fenêtre et le nombre de points sélectionnés pour le KLT et l'estimation de la variation d'intensité lumineuse pour le RMR.

Un exemple d'utilisation du logiciel MATISSE est illustré sur la figure C.1.



FIG. C.1 – Un exemple d'utilisation de MATISSE.

Titre : Contribution à la construction de mosaïques d'images sous-marines géo-référencées par l'introduction de méthodes de localisation

Résumé : En milieu sous-marin, les mosaïques d'images collectées par un véhicule en déplacement permettent d'obtenir des cartes visuelles d'une zone d'intérêt. La trajectoire estimée du véhicule par simple intégration des vecteurs vitesse présente une dérive qui se répercute dans la mosaïque (en particulier lorsque le véhicule recoupe sa propre trajectoire). Le travail de cette thèse a consisté à estimer de la façon la plus précise la trajectoire du véhicule à partir de mesures de cap et de vitesse et d'une séquence d'images, en tenant compte des lieux de recoupement (ou points de rendez-vous). Nous estimons dans un premier temps ces couples d'instant par appariement d'images, puis les introduisons dans l'algorithme de localisation du véhicule. Dans un deuxième temps, la mosaïque est reconstruite. La méthodologie proposée ici devrait permettre de peaufiner encore la cartographie dans la mesure où les variances des paramètres de mouvement estimés par les images seules sont disponibles. Dans ce cas, une fusion des deux trajectoires est réalisable. Si ce n'est pas le cas, la cartographie visuelle est finalement améliorée au moyen de techniques de déformation d'images.

Mots-clés : Localisation, mosaïque, image sous-marine, système gyro-doppler, mise en correspondance, estimation de position, estimation du mouvement, fusion, déformation, amélioration.

Title : Contribution to the underwater image geo-referenced mosaic construction by introducing localization methods

Abstract : In the underwater environment, an image mosaic allows to obtain a panoramic view of a site of interest. The image sequence, used to create such mosaic, is acquired by a moving vehicle. The vehicle trajectory is estimated by a simple speed vector integration. As a consequence, a drift appears. This is directly palpable on the mosaic, especially when the vehicle crosses its own trajectory (at "rendezvous" times). This work consists in estimating as precisely as possible, the trajectory from angle and speed measurements and an image sequence, by taking into account the rendezvous times. First, the couples of rendezvous times are estimated, then they are introduced in the vehicle localization algorithm. Second, the mosaic is created from these estimated positions. The proposed methodology could allow to obtain a more accurate cartography, provided the motion parameter variances are available. In this case, a fusion between the estimated trajectory and the optical trajectory (obtained by the image sequence motion only) is feasible. Otherwise, the visual map is finally enhanced by image warping methods.

Keywords : Localization, mosaic, underwater image, position estimation, gyro-doppler system, motion estimation, fusion, warping, enhancement.