



HAL
open science

Localisation et cartographie simultanées par ajustement de faisceaux local : propagation d'erreurs et réduction de la dérive à l'aide d'un odomètre

Alexandre Eudes

► To cite this version:

Alexandre Eudes. Localisation et cartographie simultanées par ajustement de faisceaux local : propagation d'erreurs et réduction de la dérive à l'aide d'un odomètre. Autre. Université Blaise Pascal - Clermont-Ferrand II, 2011. Français. NNT : 2011CLF22112 . tel-00662438

HAL Id: tel-00662438

<https://theses.hal.science/tel-00662438>

Submitted on 24 Jan 2012

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.

N° d'ordre : D.U. 2112
EDSPIC : 516

UNIVERSITÉ BLAISE PASCAL - CLERMONT-FERRAND II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée par :
Alexandre EUDES

Formation Doctorale CSTI :
Composants et Systèmes pour le Traitement de l'Information

en vue de l'obtention du grade de

DOCTEUR D'UNIVERSITÉ

spécialité : Vision pour la robotique

Localisation et cartographie simultanées par ajustement
de faisceaux local : propagation d'erreurs et réduction
de la dérive à l'aide d'un odomètre.

Soutenue publiquement le 14 mars 2011 devant le jury :

M. Roland Chapuis LASMEA (CLERMONT-FERRAND)	Président
Mme. Marie-Odile Berger INRIA (NANCY)	Rapporteur
M. Simon Lacroix LAAS (TOULOUSE)	Rapporteur
M. Philippe Bonnifait UTC/HEURIADIS (COMPIÈGNE)	Examineur
Mme. Sylvie Naudet-Colette CEA, LIST (SACLAY)	Encadrante
M. Maxime Lhuillier LASMEA (CLERMONT-FERRAND)	Directeur de thèse

Remerciements

Je souhaite remercier tout d’abord Marie-Odile Berger, Simon Lacroix, Philippe Bonnifait et Roland Chapuis de m’avoir fait l’honneur d’accepter de faire partie de mon jury de thèse.

Je tiens à remercier Michel Dhome et François Gaspard pour m’avoir fait confiance en me proposant d’effectuer cette thèse au sein de leurs laboratoires.

Je tiens à remercier les gens qui m’ont encadré durant ce travail, et ce souvent à distance ce qui ne rend pas les choses simples : Maxime Lhuillier, mon directeur de thèse, pour ses idées et sa rigueur scientifique, sa disponibilité et son soutien tout au long de ces trois ans ; Sylvie Naudet-Colette, mon encadrante, pour nos nombreuses discussions passionnées et sa bonne humeur. Je souhaite notamment leurs adresser tous mes remerciements pour leurs patiences à la relecture de ce mémoire. Enfin, je souhaite remercier Michel Dhome pour ses conseils et ses interventions avisées. J’ai beaucoup appris à vos cotés.

Je tiens à remercier le projet ANR ODIACC, pour avoir financé ma thèse et m’avoir permis d’ancrer mes travaux sur une réelle problématique et m’avoir fourni des données expérimentales sur lesquelles travailler. Ainsi, j’aimerais remercier l’ensemble des gens avec qui j’ai pu travailler sur ce projet notamment Mosalam Ebrahimi et Gaël Jacob.

Je tiens ensuite à remercier l’ensemble des personnes du LASMEA et du CEA qui m’ont fait un très bon accueil. Plus particulièrement Manu (mon coprof), Laetitia et Bertrand pour m’avoir gentiment offert le poste de secrétaire de la pièce 4016 et Stevens, Pierre, Julien pour les discussions passionnées tant scientifique qu’informatique et la très bonne ambiance de ce bureau.

Enfin, j’aimerais remercier ma famille et mes amis qui m’ont soutenu dans tous mes choix durant toutes ces années.

PierreChaRegisRedaOlivierChristelleMaximeSabriEtienneYoannGoursThierryLionelStevensTejDeboNicolasFabioFrançoisHalaFredRomainDoraMich
elDanielFrançoisJulienAhmedJonathanPatrickDanielaBertrandVincentConstantDidierJuniorManuelLaetitiaIsmaelManueKIRichardMaximeJulienJ
onathanGaelBaptisteCaroSylvieYoucefThomasAmiraPierreEricMohamedAdrienLaetitiaFlorineChristopheCharleneEloBorisRomainXavierPierreMan
uAgnesThierryMickQuocCuongAntoineDattaNicoJeremieChristopheOmarFrankSteveSalemCelinePierreMarionAlanPierreChaRegisRedaOlivierChri
stelleMaximeSabriEtienneYoannGoursThierryLionelStevensTejDeboNicolasFabioFrançoisHalaFredRomainDoraMichelDamienFrançoisJulienAhmedJ
onathanPatrickDanielaBertrandVincentConstantDidierJuniorManuelLaetitiaIsmaelManueKIRichardMaximeJulienJonathanGaelBaptisteCaroSylvie
YoucefThomasAmiraPierreEricMohamedAdrienLaetitiaFlorineChristopheCharleneEloBorisRomainXavierPierreManuAgnesThierryMickQuocCuongA
ntoineDattaNicoJeremieChristopheOmarFrankSteveSalemCelinePierreMarionAlanPierreChaRegisRedaOlivierChristelleMaximeSabriEtienneYoann
GoursThierryLionelStevensTejDeboNicolasFabioFrançoisHalaFredRomainDoraMichelDamienFrançoisJulienAhmedJonathanPatrickDanielaBertrand
VincentConstantDidierJuniorManuelLaetitiaIsmaelManueKIRichardMaximeJulienJonathanGaelBaptisteCaroSylvieYoucefThomasAmiraPierreEric
MohamedAdrienLaetitiaFlorineChristopheCharleneEloBorisRomainXavierPierreManuAgnesThierryMickQuocCuongAntoineDattaNicoJeremieChrist
opheOmarFrankSteveSalemCelinePierreMarionAlanPierreChaRegisRedaOlivierChristelleMaximeSabriEtienneYoannGoursThierryLionelStevensTej
DeboNicolasFabioFrançoisHalaFredRomainDoraMichelDamienFrançoisJulienAhmedJonathanPatrickDanielaBertrandVincentConstantDidierJuniorM
anuelLaetitiaIsmaelManueKIRichardMaximeJulienJonathanGaelBaptisteCaroSylvieYoucefThomasAmiraPierreEricMohamedAdrienLaetitiaFlorine
ChristopheCharleneEloBorisRomainXavierPierreManuAgnesThierryMickQuocCuongAntoineDattaNicoJeremieChristopheOmarFrankSteveSalemC

Résumé

Les travaux présentés ici concernent le domaine de la localisation de véhicule par vision artificielle. Dans ce contexte, la trajectoire d'une caméra et la structure 3D de la scène filmée sont estimées par une méthode d'odométrie visuelle monoculaire basée sur l'ajustement de faisceaux local. Les contributions de cette thèse sont plusieurs améliorations de cette méthode.

L'incertitude associée à la position estimée n'est pas fournie par la méthode d'ajustement de faisceaux local. C'est pourtant une information indispensable pour pouvoir utiliser cette position, notamment dans un système de fusion multi-sensoriel. Une étude de la propagation d'incertitude pour cette méthode d'odométrie visuelle a donc été effectuée pour obtenir un calcul d'incertitude temps réel et représentant l'erreur de manière absolue (dans le repère du début de la trajectoire).

Sur de longues séquences (plusieurs kilomètres), les méthodes monoculaires de localisation sont connues pour présenter des dérives importantes dues principalement à la dérive du facteur d'échelle (non observable). Pour réduire cette dérive et améliorer la qualité de la position fournie, deux méthodes de fusion ont été développées.

Ces deux améliorations permettent de rendre cette méthode monoculaire exploitable dans le cadre automobile sur de grandes distances tout en conservant les critères de temps réel nécessaire dans ce type d'application. De plus, notre approche montre l'intérêt de disposer des incertitudes et ainsi de tirer parti de l'information fournie par d'autres capteurs.

Mots-clé : SLAM Visuel, Localisation temps réel, Monoculaire, Ajustement de faisceaux local, Calcul d'incertitude

Abstract

The present work is about localisation of vehicle using computer vision methods. In this context, the camera trajectory and the 3D structure of the scene is estimated by a monocular visual odometry method based on local bundle adjustment. This thesis contributions are some improvements of this method.

The uncertainty of the estimated position was not provided by the local bundle adjustment method. Indeed, this uncertainty is crucial in a multi-sensorial fusion system to use optimally the estimated position. A study of the uncertainty propagation in this visual odometry method has been done and an uncertainty calculus method has been designed to comply with real time performance.

By the way, monocular visual localisation methods are known to have serious drift issues on long trajectories (some kilometers). This error mainly comes from bad propagation of the scale factor. To limit this drift and improve the quality of the given position, we proposed two data fusion methods between an odometer and the visual method.

Finally, the two improvements presented here allow us to use visual localisation method in real urban environment on long trajectories under real time constraints.

Key-words : Visual SLAM, Real time localisation, Monocular vision, Local bundle adjustment, Uncertainty calculus

Table des matières

0.1	Notations	xiii
	Introduction	1
1	Notions de base	5
1.1	Outils de vision par ordinateur	6
1.1.1	La caméra et son modèle	6
1.1.2	Triangulation	10
1.1.3	Calcul de pose	10
1.2	Outils mathématiques et statistiques	13
1.2.1	Estimateur	13
1.2.2	Optimisation	15
1.2.3	Covariance	17
1.2.4	Gaussienne, loi normale et modèle gaussien	18
1.2.5	Complément de Schur, inversion par bloc et probabilité . .	19
1.3	Propagation d'erreur	20
1.3.1	Introduction	20
1.3.2	Principe	20
1.3.3	Linéarisation d'une fonction définie implicitement	25
1.3.4	Définition d'une contrainte de jauge	27
2	Etat de l'art : SLAM et propagation d'erreur en vision	29
2.1	Introduction	30
2.2	localisation et reconstruction simultanées	30
2.2.1	Ajustement de faisceau et Full-SLAM	31
2.2.2	Approches utilisant le filtrage	32

2.2.3	Slam relatif	33
2.3	Propagation d'erreur en vision 3D	34
3	Ajustement de faisceaux global	37
3.1	Introduction	38
3.2	Formulation	40
3.3	Résolution du système creux	41
3.4	Modèle de bruit	45
3.5	Propagation d'erreur	46
3.6	Résumé de l'ajustement de faisceau global	49
4	Ajustement de faisceaux local	51
4.1	Introduction	52
4.2	Formulation	53
4.3	Propagation d'erreur	55
4.3.1	Différence par rapport à l'ajustement de faisceaux global	55
4.3.2	Hypothèses et modèles de bruit	55
4.3.3	Covariance résultante	56
4.3.4	Cas avec corrélation	57
4.3.5	Cas avec indépendance	59
4.4	Résumé de l'ajustement de faisceaux local	64
5	Ajustement de faisceaux local pondéré	65
5.1	Introduction	66
5.2	Formulation	67
5.3	Différence par rapport à l'AFG et l'AFL	68
5.4	Propagation d'erreur	69
5.4.1	Cas avec corrélation	70
5.4.2	Cas avec hypothèse d'indépendance	73
5.5	Résumé de l'ajustement de faisceaux local pondéré	78
6	Validation expérimentale du calcul d'incertitude	79
6.1	Introduction	80
6.2	Présentation des séquences	81

6.3	Comparaison avec l'incertitude de l'AFG	81
6.3.1	Critères de comparaison	82
6.3.2	Vérifications préliminaires sur le comportement de l'incertitude	82
6.3.3	Résultat sur des séquences de simulation	84
6.3.4	Résultat sur la séquence de synthèse	89
6.3.5	Résultat sur une séquence réelle	92
6.3.6	Temps de calcul	96
6.3.7	Conclusion	96
6.4	Monte-Carlo	97
6.4.1	Méthodologie pour le test d'une application de l'AFL	97
6.4.2	Présentation des résultats	97
6.4.3	Analyses des résultats	102
6.4.4	Complément d'analyse	102
6.5	Test de Consistance	103
6.5.1	Définition	103
6.5.2	Résultat	104
6.6	Correction de la méthode AFLP	107
6.7	Conclusion	107
7	Application au SLAM Visuel	109
7.1	SLAM Visuel	110
7.1.1	Schéma général	110
7.1.2	Localisation rapide	111
7.1.3	Choix de l'image clef	113
7.1.4	Mise à jour du nuage de points 3D	114
7.1.5	Initialisation	115
7.2	Utilisation de l'ajustement de faisceau local pondéré	116
7.2.1	Ajustement de faisceaux local pondéré	116
7.2.2	Évaluation expérimentale	117
7.2.3	Résultats sur la séquence de synthèse	118
7.2.4	Résultat sur une séquence réelle	122
7.3	Conclusion	125

8	Vers la fusion	127
8.1	Introduction	128
8.2	Fusion avec un odomètre	128
8.3	Intégration simple	130
8.3.1	Résultats	132
8.4	Fusion par AFLP	140
8.4.1	Processus de correction odométrique	141
8.4.2	Intégration du processus de fusion	143
8.4.3	Résultats	144
8.5	Conclusion	150
	Conclusion	151
	Annexes	155
	Bibliographie	158
	Liste des figures	168
	Liste des tableaux	173

0.1 Notations

Type

\mathbf{v}	un vecteur
M	une matrice
f	une fonction
\mathcal{U}	un ensemble

Opération sur les vecteurs

$ \mathbf{v} $	dimension du vecteur \mathbf{v}
$[\mathbf{u}\mathbf{v}] = [\mathbf{u}, \mathbf{v}] = [\mathbf{u}^T \mathbf{v}^T]^T$	concaténation des vecteurs \mathbf{u} et \mathbf{v}
$\mathbf{u} \setminus \mathbf{v}$	vecteur des coefficients de \mathbf{u} qui n'appartiennent pas à \mathbf{v}
$\mathbf{u} \cap \mathbf{v}$	vecteur des coefficients de \mathbf{u} qui appartiennent à \mathbf{v}
$\ \mathbf{v}\ _2$	norme euclidienne du vecteur \mathbf{v}
$E(\mathbf{v})$	espérance du vecteur aléatoire \mathbf{v}

Opération sur les matrices et matrices particulières

M^T	transposée de la matrice M
I_n	la matrice identité de dimension $n \times n$
0_n	la matrice nulle de dimension $n \times n$
$0_{n,m}$	la matrice nulle de dimension $n \times m$
$J_{\mathbf{v}}^f$	la jacobienne de f par rapport à \mathbf{v}
$C_{\mathbf{v}}$	covariance du vecteur \mathbf{v}
$C_{\mathbf{u}/\mathbf{v}}$	covariance croisée du vecteur \mathbf{u} par rapport à \mathbf{v}

Éléments géométriques

Point 3D	
$\mathbf{s}^T = (\dots \mathbf{s}_i^T \dots)$	le vecteur des points 3D
\mathbf{s}_i	le $i^{\text{ème}}$ point 3D du vecteur point 3D \mathbf{s}
Caméra	
$\mathbf{c}^T = (\dots \mathbf{c}_i^T \dots)$	le vecteur des caméras
\mathbf{c}_i	les paramètres de la $i^{\text{ème}}$ caméra du vecteur caméra \mathbf{c}
$\Delta \mathbf{c}_i$	la translation de la caméra i .
$\omega \mathbf{c}_i$	la rotation de la caméra i .
Points 2D	
$\mathbf{m} = (\dots \mathbf{m}_{i,j}^T \dots)$	vecteur des points 2D
$\mathbf{m}_{i,j}$	point 2D vu par la caméra \mathbf{c}_i observation du point 3D \mathbf{s}_j
\mathfrak{M}	l'ensemble des couples d'indices (i,j) tels que $\mathbf{m}_{i,j}$ existe
Fonction de projection	
$P(\mathbf{c}_i, \mathbf{s}_j)$	la projection de \mathbf{s}_j dans la caméra \mathbf{c}_i

Fonction Ajustement de faisceaux

Global

f^g	fonction Ajustement de Faisceaux Global
F^g	fonction de projection/observation de l'Ajustement de Faisceaux Global

Local	
f^l	fonction Ajustement de Faisceaux local
F^l	fonction de projection/observation de l'Ajustement de faisceaux local
Local pondéré	
f^{lp}	fonction Ajustement de Faisceaux local pondéré
F^{lp}	fonction de projection/observation de l'Ajustement de faisceaux local pondéré
Paramètres des fonctions Ajustement de faisceaux	
\mathbf{y}	vecteur des données (mesures)
\mathbf{x}	vecteur des variables estimées
\mathbf{p}	vecteur des paramètres (variables supposées connues et non bruitées dans l'estimation)
Autres objets et notations	
\mathbf{x}^-	la valeur de \mathbf{x} à l'instant précédent
$\bar{\mathbf{m}}$	valeur vraie/exacte de \mathbf{m}
$\hat{\mathbf{m}}$	valeur estimée de \mathbf{m}
$\mathcal{N}(\mathbf{m}, \mathbf{C})$	une loi normale de moyenne \mathbf{m} et de variance/covariance \mathbf{C}
$ \mathcal{U} $	cardinal de l'ensemble \mathcal{U}

Dans tout le mémoire, nous confondrons les objets avec leurs paramètres. Ainsi on se référera à la caméra de paramètre c simplement par la caméra c .

Introduction

L'automatisation de la conduite automobile est un sujet de recherche actuel, comme en témoignent par exemple les résultats des différents concours du DARPA (grand challenge et urban challenge). Certaines des technologies impliquées dans ces problèmes sont devenues suffisamment matures pour des expériences en route ouverte, voire une industrialisation.

La conduite automatique de véhicule requiert une localisation précise du véhicule ainsi qu'une connaissance de l'environnement dans lequel il évolue.

Le capteur le plus utilisé pour la localisation est sans nul doute le GPS. Il a l'avantage de fournir une position sur l'ensemble de la planète, avec une précision suffisante d'environ 5m en moyenne, pour des applications de positionnement classique (guidage d'usagers, trafic maritime et aérien, ...).

Mais, le guidage de véhicule automatique en milieu urbain ne peut se satisfaire de la précision métrique d'un GPS standard en condition optimale. Il est par exemple insuffisant pour connaître sur quelle voie de circulation se trouve le véhicule. De plus, la précision du GPS peut être dégradée par le masquage du signal GPS (tunnel, bâtiments, végétations) et par les phénomènes de multi-chemin dus aux réflexions des ondes satellites sur les bâtiments ou les objets environnants. Ces dégradations sont par nature imprévisibles et ont malheureusement un impact significatif sur la qualité du positionnement. Pour toutes ces raisons, le GPS est un capteur avec des propriétés intéressantes, mais il y a des applications de localisation de véhicule où l'utilisation du GPS à lui seul ne suffit pas pour obtenir une position du véhicule fiable et suffisamment précise.

Les constructeurs automobiles veulent de plus en plus développer les applications d'aide à la conduite (détection d'obstacles, de piétons, détection de la signalisation routière et guidage du conducteur). L'un des capteurs utilisé pour ces tâches est la caméra. En effet, avec l'augmentation de la puissance de calcul embarquée, il devient possible de traiter le flux vidéo en temps réel et d'en extraire l'information voulue. Ici, l'image est à la fois une bonne source d'information

pour le système et un moyen pratique de rendre compte du résultat en surimposant de l'information additionnelle sur l'image initiale (réalité augmentée). Ainsi, les caméras sont des capteurs de plus en plus disponibles dans le domaine automobile. Ces caméras étant disponibles sur les véhicules récents ou futurs, il serait judicieux d'utiliser aussi ce capteur pour obtenir la position du véhicule.

En effet, une caméra peut aussi servir comme capteur de localisation, et ce, de trois manières différentes :

- **Estimation du déplacement relatif** : le déplacement est calculé entre deux images. Ce type de méthodes est appelé odométrie visuelle ou egomotion. Cette technique a d'ores et déjà été utilisée avec succès pour des applications de guidage de véhicule. Citons par exemple son utilisation dans le cadre de l'exploration martienne par les rovers Spirit et Opportunity. L'odométrie visuelle a remplacé l'odométrie standard pour limiter les erreurs dues aux glissements [74].
- **La relocalisation** : le système localise la caméra dans un environnement déjà connu.
C'est ce principe qui est utilisé au LASMEA pour effectuer le guidage d'un véhicule. Le trajet est appris et une carte de l'environnement est reconstruite, puis le véhicule se sert de cette carte pour se relocaliser et suivre le chemin appris [89]. Notons que ce processus est en cours d'industrialisation pour le guidage de véhicule automatique à destination du transport public sur de courtes distances (projet VIPA).
- **SLAM visuel** : ce type de méthode estime conjointement l'environnement et la position des caméras. C'est dans le cadre de ces méthodes que nous nous plaçons.

Notre problématique est résumée de la façon suivante. Nous nous plaçons dans le cadre d'un véhicule qui se déplace en milieu urbain et dont on souhaite connaître la position à chaque instant. Pour cela nous disposons des images fournies par une unique caméra placée derrière le pare-brise. Dans un second temps, nous utiliserons aussi l'odométrie fournie par le véhicule.

Contexte de la thèse

Cette thèse est issue d'une longue collaboration entre le CEA et le LASMEA. Dans ce cadre, les travaux de thèse d'Etienne Mouragnon ont permis de développer une méthode de localisation temps réel et précise en utilisant une unique caméra calibrée.

Dans la continuité de cette thèse, le projet ANR ODIAAC, a eu pour objectif de pallier les déficiences du GPS, principalement les erreurs de localisation commises en ville lors de la traversée de canyon urbain où les signaux GPS sont

fortement perturbés. Pour cela, l'idée a été de coupler le GPS avec un système de vision pour permettre d'obtenir une position plus robuste et fiable, et ce, en utilisant uniquement des capteurs à bas coût. Ce projet est une collaboration entre Renault, DotMobil, le CEA et le LASMEA. En début de projet, l'approche retenue était d'utiliser la donnée fournie par le SLAM visuel comme un capteur absolu et d'effectuer la fusion de cette donnée avec un GPS en utilisant un filtre de Kalman.

Une donnée est manquante pour effectuer correctement ce travail. L'incertitude globale (relative au début de la trajectoire) associée à la vision est en effet inconnue. Cette donnée est pourtant essentielle pour effectuer une fusion dans des conditions optimales. Un deuxième problème inhérent à la méthode de SLAM visuel utilisée est la dérive du facteur d'échelle. Cette dérive provient de l'utilisation d'une unique caméra. Dans ce contexte, le facteur d'échelle de la reconstruction n'est pas observable, et, expérimentalement on observe une dérive de ce facteur. Cette dérive, négligeable sur de petites distances devient importante (et même prépondérante) sur de grandes trajectoires.

Les travaux de cette thèse s'inscrivent donc dans le cadre des problématiques soulevées par ce projet. Elle a été réalisée pendant 3 ans, et c'est déroulée pour moitié au LASMEA à Clermont-Ferrand et pour moitié au CEA (centre de Saclay). Ces travaux ont été effectués sous la direction de Maxime Lhuillier (LASMEA) et encadrés par Michel Dhome (LASMEA) et Sylvie Naudet (CEA).

Réalisation

Les contributions de cette thèse sont plusieurs améliorations d'une méthode de SLAM visuel monoculaire basée sur l'ajustement de faisceaux local.

Tout d'abord, une étude de la propagation d'incertitude pour cette méthode de SLAM itérative a été effectuée. En effet, contrairement à d'autres méthodes comme le filtre de Kalman, l'ajustement de faisceau local ne fournissait pas d'incertitude associée à la position estimée. Pour cela, quatre méthodes de calcul d'incertitude de la pose de la caméra ont été étudiées et expérimentées sous deux hypothèses de bruits. L'objectif est d'obtenir une méthode de calcul d'incertitude temps réel et représentant l'erreur de manière globale (dans le repère du début de la trajectoire).

Ensuite, deux méthodes ont été développées pour réduire la dérive du facteur d'échelle propre aux méthodes monoculaires. Ces deux méthodes utilisent l'information fournie par un odomètre pour obtenir ce résultat.

Ces travaux ont fait l'objet de plusieurs publications :

- CVPR'09 [33], "Error propagation for local bundle adjustment" ;

- RFIA'10 [35] & ORASIS'09 [34], "Propagation d'erreur pour l'ajustement de faisceaux local" ;
- ICPR'10 [36], "Fast Odometry Integration in Local Bundle Adjustment-Based Visual SLAM" ;
- BMVC'10 [37], "Weighted Local Bundle Adjustment and Application to Odometry and Visual SLAM Fusion" ;

Plan du mémoire

Ce mémoire est organisé de la manière suivante. Nous commencerons par un rappel de quelques notions scientifiques et un état de l'art. Puis nous aborderons les techniques de propagation pour différents ajustements de faisceaux, enfin nous effectuerons une vérification expérimentale de ces méthodes. Nous présenterons ensuite la méthode de SLAM utilisée, ainsi que ses limites. Enfin, nous proposerons des méthodes permettant de limiter les problèmes de la méthode originale. Plus précisément, le plan est le suivant.

- Le chapitre 1 présente les différents outils utilisés au cours de cette thèse.
- Le chapitre 2 est un état de l'art sur les techniques de SLAM et sur l'utilisation de techniques de propagation d'erreur en vision 3D.
- Le chapitre 3 est un rappel sur la méthode de l'ajustement de faisceaux global et la propagation d'erreur associée.
- Le chapitre 4 présente l'ajustement de faisceaux local développé durant la thèse d'Etienne Mouragnon et montre comment propager l'incertitude dans ce cas.
- Le chapitre 5 présente un nouvel ajustement de faisceaux local, l'ajustement de faisceaux local pondéré et la propagation d'erreur associée.
- Le chapitre 6 contient la validation expérimentale des méthodes de propagation d'erreur formalisées dans les chapitres 4 et 5.
- Le chapitre 7 est consacré à la méthode de SLAM utilisée durant cette thèse et montre les performances de l'ajustement de faisceaux local pondérés introduits dans le chapitre 5.
- Le chapitre 8 présente deux méthodes permettant d'utiliser l'odomètre avec le SLAM visuel monoculaire de manière à limiter la dérive du facteur d'échelle observée dans le chapitre 7.

Chapitre 1

Notions de base

Sommaire

1.1 Outils de vision par ordinateur	6
1.1.1 La caméra et son modèle	6
1.1.2 Triangulation	10
1.1.3 Calcul de pose	10
1.2 Outils mathématiques et statistiques	13
1.2.1 Estimateur	13
1.2.2 Optimisation	15
1.2.3 Covariance	17
1.2.4 Gaussienne, loi normale et modèle gaussien	18
1.2.5 Complément de Schur, inversion par bloc et probabilité	19
1.3 Propagation d'erreur	20
1.3.1 Introduction	20
1.3.2 Principe	20
1.3.3 Linéarisation d'une fonction définie implicitement	25
1.3.4 Définition d'une contrainte de jauge	27

Ce chapitre présente les différentes notions de vision et de mathématiques utilisées au cours de cette thèse. Il n'est pas destiné à une lecture exhaustive mais est un complément au reste de ce tapuscrit. Le lecteur averti pourra donc commencer sa lecture au chapitre suivant.

1.1 Outils de vision par ordinateur

1.1.1 La caméra et son modèle

Une caméra est un dispositif permettant de produire une image 2D de l'environnement 3D. C'est un capteur très riche, elle apporte en effet à la fois une information photométrique et géométrique sur la scène observée. Ici, nous allons nous intéresser à l'aspect géométrique de la caméra.

Une caméra est constituée de deux éléments : un objectif, interface entre le monde extérieur et la caméra, et un capteur plan sensible à la lumière. Dans les premiers appareils photographiques, l'objectif était réduit à un simple trou (appelé sténopé) et le plan sensible était un papier photosensible (le réactif change de couleur lors de son exposition à la lumière grâce à une réaction chimique).

Dans ce système, tous les rayons lumineux passent donc par un unique point (le sténopé) avant de frapper le plan sensible à la lumière. C'est encore le même principe qui est utilisé aujourd'hui pour la plupart des appareils photographiques. Le sténopé a été remplacé par un objectif qui permet d'augmenter la quantité de lumière transmise et la plaque photosensible par un capteur numérique.

Une caméra va être modélisée comme une transformation du monde (espace 3D) en une image (espace 2D). Le modèle des caméras décrites précédemment est le modèle sténopé pour lequel tous les rayons lumineux de la scène passent par un même point puis se projettent sur le capteur.

Cette transformation se modélise par une projection centrale. La géométrie projective va nous offrir un cadre mathématique permettant de simplifier l'expression de ce type de transformation.

Transformations et géométrie projective

L'espace projectif est construit à partir d'un espace vectoriel. Avec l'espace vectoriel \mathbb{R}^{n+1} , on construit la relation d'équivalence suivante :

$$\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n+1}, \mathbf{u} \sim \mathbf{v} \Leftrightarrow \exists \lambda \in \mathbb{R}^*, \mathbf{u} = \lambda \mathbf{v} \quad (1.1)$$

L'ensemble des classes d'équivalence pour cette relation ' \sim ' est appelé espace projectif. Cet espace est un espace de dimension n et sera noté \mathbb{P}^n . Ce n'est pas l'espace projectif qui nous intéresse ici, mais la notion de géométrie projective qui étudie les transformations projectives dans les espaces affines en utilisant l'espace projectif. L'introduction de cet espace projectif permet de plus de formaliser la notion de point à l'infini dans l'espace affine.

En effet, si $\mathbf{u}^+ \in \mathbb{P}^n$, alors $\mathbf{u}^+ = (u_1 \dots u_{n+1})$. Si u_{n+1} est non nul, u^+ représente le vecteur $\mathbf{u} \in \mathbb{R}^n / \mathbf{u} = (\frac{u_1}{u_{n+1}} \dots \frac{u_n}{u_{n+1}})$. Dans le cas contraire (si $u_{n+1} = 0$), \mathbf{u} est un point à l'infini ou une direction dans l'espace \mathbb{R}^n . Les coordonnées de \mathbf{u}^+ sont appelées coordonnées homogènes de \mathbf{u} . Nous appelons π , la transformation qui permet de passer de \mathbb{P}^n à \mathbb{R}^n .

$$\mathbb{P}^n \rightarrow \mathbb{R}^n \quad (1.2)$$

$$\pi : \mathbf{u}^+ = (u_1 \dots u_{n+1}) \mapsto \mathbf{u} = \left(\frac{u_1}{u_{n+1}} \dots \frac{u_n}{u_{n+1}} \right) \quad (1.3)$$

Cas du plan projectif, dualité point/droite

L'espace projectif \mathbb{P}^2 est appelé plan projectif. C'est l'espace projectif associé au plan 2D \mathbb{R}^2 . Dans \mathbb{P}^2 , un point 2D sera représenté par les coordonnées $\mathbf{m} = (m_1, m_2, m_3)$. De même, il existe une représentation des droites de \mathbb{R}^2 . Une droite définie implicitement par l'équation $ax_1 + bx_2 + c = 0$, $\mathbf{x} \in \mathcal{R}^2$, $a, b, c \in \mathcal{R}$ aura pour représentation $\mathbf{l} = (a, b, c)$. Dans cet espace, le point et la droite sont duaux, il en découle les relations suivantes : un point \mathbf{m} appartiendra à une droite \mathbf{l} si $\mathbf{l}^T \mathbf{m} = 0$. De plus, si \mathbf{m}, \mathbf{m}' sont deux points alors $\mathbf{l} = \mathbf{m} \wedge \mathbf{m}'$ est la droite passant par ces deux points et si \mathbf{l}, \mathbf{l}' sont deux droites alors $\mathbf{m} = \mathbf{l} \wedge \mathbf{l}'$ est l'intersection des deux droites.

Caméra perspective

Une caméra perspective (présentée en introduction) peut être modélisée par une transformation projective de l'espace 3D (\mathbb{P}^3) vers l'espace 2D (\mathbb{P}^2). Cette transformation surjective va nous permettre de modéliser la projection effectuée par la caméra lors de l'acquisition de l'image et ainsi pour chaque point \mathbf{s} de l'espace 3D, on va chercher le point 2D \mathbf{m} correspondant dans le plan image. On a :

$$\mathbf{m}^+ = \mathbf{P} \mathbf{s}^+ \quad (1.4)$$

avec \mathbf{P} une matrice 3×4 .

Cette matrice de projection peut se décomposer en 3 transformations projectives :

$$\mathbf{P} = \mathbf{K} \mathbf{\Pi} \mathbf{T} \quad (1.5)$$

avec :

- \mathbf{T} la matrice de pose, c'est la matrice de transformation associée à une homographie de \mathbb{P}^3 (transformation de \mathbb{P}^3 dans \mathbb{P}^3), \mathbf{T} est une matrice 4×4 ;

- Π la matrice représentant la projection centrale π de \mathbb{P}^3 dans \mathbb{P}^2 autour du centre de la caméra, Π est une matrice 3×4 ;
 - K la matrice de calibrage, c'est la transformation associée à une homographie de \mathbb{P}^2 (transformation de \mathbb{P}^2 dans \mathbb{P}^2), K est une matrice 3×3 .
- Nous allons maintenant détailler le rôle de chacune de ces transformations.

Paramètres extrinsèques La première transformation est une transformation de l'espace 3D. Elle a pour but de se placer dans le repère associé à la caméra de manière à effectuer la projection du monde tel qu'il est observé par cette caméra. Les paramètres associés à cette transformation sont appelés paramètres extrinsèques.

La caméra a 6 degrés de liberté et sa position est paramétrée ainsi :

- la position 3D du centre de la caméra, 3 degrés de liberté, est paramétré par un vecteur 3D \mathbf{t} ;
- l'orientation de la caméra, 3 degrés de liberté, est paramétrée par une matrice de rotation R de taille 3×3 qui peut être obtenue à partir des angles d'Euler ou d'un quaternion.

La connaissance de la pose de la caméra permet de définir les relations de passage entre le repère monde \mathfrak{w} et le repère caméra \mathfrak{c} .

$$\mathbf{s}_{\mathfrak{w}}^+ = \begin{pmatrix} R & \mathbf{t} \\ 0_{1 \times 3} & 1 \end{pmatrix} \mathbf{s}_{\mathfrak{c}}^+ \quad (1.6)$$

$$\mathbf{s}_{\mathfrak{c}}^+ = \begin{pmatrix} R^T & -R^T \mathbf{t} \\ 0_{1 \times 3} & 1 \end{pmatrix} \mathbf{s}_{\mathfrak{w}}^+ \quad (1.7)$$

Et donc la matrice des paramètres extrinsèques est définie par :

$$T = \begin{pmatrix} R^T & -R^T \mathbf{t} \\ 0_{1 \times 3} & 1 \end{pmatrix}. \quad (1.8)$$

Projection centrale La seconde transformation est la projection centrale de l'espace 3D dans le repère caméra, vers le plan 2D appelé rétine (\mathfrak{r}).

Grâce à l'utilisation de la géométrie projective, la projection centrale de \mathbb{P}^3 dans \mathbb{P}^2 s'exprime linéairement par la relation suivante :

$$\mathbf{m}_{\mathfrak{r}}^+ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{s}_{\mathfrak{c}}^+ \quad (1.9)$$

Ceci nous donne l'expression de la matrice Π définie précédemment.

Paramètres intrinsèques La dernière transformation permet de transformer les coordonnées du point du repère de la rétine au repère image (i). Cette transformation modélise les effets dus à la position et les caractéristiques du capteur de la caméra. Elle permet de passer aux coordonnées pixels telles qu'utilisées dans l'image. Les paramètres de cette transformation sont appelés paramètres intrinsèques de la caméra.

La transformation projective associée a une matrice de cette forme :

$$K = \begin{pmatrix} \alpha f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.10)$$

Cinq paramètres permettent de définir le capteur.

- La distance focale f : c'est la distance orthogonale entre le centre de la caméra et le plan image, elle est exprimée en pixels par unité de mesure.
- Le point principal (u_0, v_0) en pixel, il est défini comme le point d'intersection de l'axe optique avec le plan rétinien. Utiliser le centre du capteur comme point principal est souvent une bonne approximation.
- α est le rapport d'aspect des pixels, ce paramètre traduit le fait que les pixels ne sont pas carrés, mais rectangulaires.
- s est le "skew ratio", ce paramètre traduit le fait que les pixels ne sont pas tout à fait rectangulaires mais trapézoïdaux. Ce défaut des capteurs est souvent négligeable et alors s est nul.

Dans le cas de courte focale (grand angle), une distorsion importante de l'image obtenue peut apparaître. Cette distorsion n'est pas modélisée par le modèle précédent. Pour corriger ce problème, des coefficients de correction de la distorsion sont souvent ajoutés au modèle précédent pour corriger les images ou les points 2D obtenus. Les paramètres intrinsèques sont obtenus lors d'une opération appelée calibrage. Pour plus d'informations, on pourra se référer à [65]. Dans toute la suite de la thèse, on considère les caméras calibrées et les images corrigées de la distorsion.

En résumé, la projection obtenue est la suivante :

$$\begin{aligned} m_i^+ &= P s_w^+ \\ m_i^+ &= K \Pi T s_w^+ \\ m_i^+ &= \begin{pmatrix} \alpha f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R^T & -R^T t \\ 0_{1 \times 3} & 1 \end{pmatrix} s_w^+ \quad (1.11) \end{aligned}$$

$$m_i^+ = K (R^T \quad -R^T t) s_w^+ \quad (1.12)$$

Finalement, nous notons P la fonction :

$$\mathbb{R}^6 \times \mathbb{R}^3 \rightarrow \mathbb{R}^2 \quad (1.13)$$

$$(\mathbf{c}, \mathbf{s}) \mapsto \mathbf{m} = \pi \left(\begin{array}{c} KR^T \\ -R^T \mathbf{t} \end{array} \right) \begin{pmatrix} \mathbf{s} \\ 1 \end{pmatrix}. \quad (1.14)$$

avec (R, \mathbf{t}) la pose de la caméra \mathbf{c} où le paramétrage de R choisi est minimal (angle d'Euler), K sa matrice de calibrage, \mathbf{s} les coordonnées d'un point 3D dans le repère monde (\mathbf{w}) .

Nous allons voir maintenant comment la modélisation de la caméra peut être utilisée pour retrouver les paramètres de la caméra ou des points observés.

1.1.2 Triangulation

La triangulation est une opération consistant à retrouver la position d'un point 3D à partir de ses observations par au moins deux caméras dont les poses sont connues. Le problème est simple dans le cas où il n'y a pas de bruit. Il suffit de trouver l'intersection des deux rayons rétroprojetés définis par chaque couple (caméra, point 2D). En pratique, en présence de bruit, ces rayons ne se coupent plus. Il faut donc trouver "la moins mauvaise des solutions" à ce problème. Plusieurs méthodes ont été proposées, on peut citer les trois suivantes :

- la méthode du point milieu. C'est la méthode la plus rapide, elle consiste à garder le point équidistant des deux rayons. Elle peut être étendue à plus de deux vues en calculant la triangulation par paire de caméras puis en conservant l'isobarycentre comme résultat de la triangulation ;
- une méthode linéaire basée sur la DLT (direct linear transform)[49]. Elle a l'avantage d'être linéaire et peut s'étendre facilement à plus de 2 observations. Elle conduit à la résolution d'un système d'équations linéaires ;
- une méthode optimale dans le cas de bruit gaussien [51]. Cette méthode a pour désavantage de devoir trouver les racines d'un polynôme de degrés 6.

Le détail et une comparaison de ces trois méthodes peuvent être trouvés dans [51] et récemment une autre méthode optimale de triangulation plus rapide a été proposée [59].

1.1.3 Calcul de pose

Le calcul de pose permet de déterminer la position et l'orientation d'une caméra à partir de connaissances sur les points 2D observés.

Il y a plusieurs cas possibles suivant que l'on connaisse les relations 2D/2D ou 2D/3D associés à ces points.

A partir de correspondances 2D/2D

Ici, les correspondances entre les points 2D de deux caméras sont connues et on veut déterminer la pose relative entre les deux caméras.

Géométrie épipolaire La géométrie épipolaire résume dans l'image les contraintes dues au fait que deux points 2D sont issus de la même observation 3D. Nous avons vu précédemment que dans le cas d'une caméra projective, il existe un faisceau de points 3D qui se projettent tous en un même point 2D. Maintenant, si nous projetons ce faisceau dans la deuxième caméra, nous obtenons une droite 2D qui est la projection de tous les points 3D associés au point 2D de la première caméra. Ainsi, le point 2D, observation du point 3D dans la deuxième caméra, doit donc se trouver sur cette droite. Cette droite est appelée droite épipolaire. Pour la première caméra, le centre optique est un point particulier. En effet d'après la définition de la caméra perspective, ce point appartient à tous les faisceaux de points 3D issus d'un point 2D de l'espace image. De ce fait, la projection de ce point dans la deuxième caméra est aussi un point particulier, il est l'intersection de toutes les droites épipolaires. Il est nommé épipole.

Les contraintes de la géométrie épipolaire peuvent être résumée sous cette forme :

$$\mathbf{m}_1^{+T} \mathbf{F} \mathbf{m}_2^+ = 0 \quad (1.15)$$

Ici, $\mathbf{F} \mathbf{m}_2^+$ sont les coordonnées homogènes de la droite épipolaire associée à \mathbf{m}_2^+ dans la caméra 1 et l'équation précédente induit bien la contrainte d'appartenance. La matrice \mathbf{F} est appelée la matrice fondamentale ([48],[39]), c'est une matrice 3×3 de rang 2.

Cette matrice peut se décomposer de la manière suivante :

$$\mathbf{F} = \mathbf{K}_1^T \mathbf{E} \mathbf{K}_2 \quad (1.16)$$

avec $\mathbf{K}_1, \mathbf{K}_2$ les matrices des paramètres intrinsèques des deux caméras.

La matrice \mathbf{E} est appelée la matrice essentielle, c'est une matrice 3×3 ayant deux valeurs propres égales et la troisième nulle. On peut montrer que la matrice

essentielle s'exprime en fonction de la transformation entre les deux caméras.

$$E = R [R^T \mathbf{t}]_{\times} \quad (1.17)$$

$$[\mathbf{t}]_{\times} = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} \quad (1.18)$$

avec R la matrice de rotation entre 1 et 2, \mathbf{t} le vecteur de translation entre 1 et 2, et $[\mathbf{t}]_{\times}$ la matrice de produit vectoriel associé à \mathbf{t} . Le vecteur \mathbf{t} et la matrice R peuvent être retrouvés en décomposant la matrice E par SVD [49].

De même, la matrice fondamentale peut s'exprimer en fonction de K_1, K_2, R et \mathbf{t} :

$$F = K_1^{-T} R K_2 [K_1 R^T \mathbf{t}]_{\times} \quad (1.19)$$

Cas non calibré Dans le cas où les paramètres intrinsèques des caméras est inconnu, le calcul de la transformation relative entre la caméra 1 et la caméra 2 se fait en estimant la matrice fondamentale à partir de correspondances entre les deux images. Cela peut être fait en utilisant l'algorithme des 8 points [50]. A partir de cette estimation de la matrice fondamentale, il est possible de retrouver les matrices de projection de chaque caméra [49]. En effet, la transformation entre les deux caméras est connue à une transformation projective de l'espace 3D près. Les matrices de calibrage K_1 et K_2 peuvent être retrouvées par une procédure appelée autocalibrage [9]. La reconstruction projective pourra alors être transformée vers une reconstruction euclidienne.

Cas calibré Dans le cas où le calibrage des caméras est connu, on utilise plutôt la matrice essentielle E pour diminuer les erreurs d'estimations. Elle sera estimée à partir de 5 correspondances en utilisant l'algorithme des 5 points [83]. Puis R et \mathbf{t} peuvent être estimés à partir de la matrice essentielle par SVD. Cet algorithme permet de retrouver la pose relative des deux caméras. Il est à noter que la translation sera seulement connue à une échelle près.

A partir de correspondances 2D/3D

Nous cherchons cette fois-ci à déterminer la pose de la caméra à partir de correspondances entre les points 2D et des points 3D de position connues. Ici, on va chercher à localiser la caméra dans le repère défini par ces points 3D.

Il est nécessaire de connaître la position d'au moins 3 points 3D dans le cas calibré et 6 dans le cas non calibré.

De nombreuses méthodes ont été proposées pour résoudre ce problème. On peut trouver une comparaison des méthodes de base dans Haralick *et al.* [46]. Plus récemment, Lepetit *et al.* [68] ont présenté une méthode plus rapide et plus précise. Il est à noter que le calcul de pose à partir de correspondances 2D-3D est une méthode rapide et contrairement à l'approche précédente, il n'y a pas d'ambiguïté sur l'échelle.

Utilisation de ces algorithmes

Les algorithmes présentés précédemment sont des algorithmes minimaux : l'estimation est faite à partir d'un minimum de données. Ces algorithmes ne sont donc pas résistants aux données aberrantes. Pour pallier ce problème, ces algorithmes sont souvent utilisés conjointement avec des techniques d'estimation robuste comme RANSAC [42]. Elles permettent, en estimant à partir de plusieurs ensembles de données, de trouver une solution filtrant les données aberrantes.

Une étape de minimisation non linéaire est souvent effectuée après cette première estimation par les algorithmes minimaux. En effet, ces algorithmes trouvent la meilleure solution pour un sous ensemble minimal des données et utilisent toutes les données uniquement pour vérifier la validité de l'estimation. Dans cette étape, la minimisation d'un critère ayant un sens physique, par exemple l'erreur de reprojection ou l'adéquation à la géométrie épipolaire, permet de trouver la meilleure solution pour l'ensemble des données considérées.

1.2 Outils mathématiques et statistiques

1.2.1 Estimateur

Maximum de vraisemblance

Le maximum de vraisemblance est une méthode permettant de construire un estimateur pour un problème donné. Pour utiliser cette méthode, il est nécessaire de connaître la loi de probabilité des estimés.

Les avantages de cette méthode sont de fournir un estimateur dans de nombreuses situations (dès qu'une distribution peut être inférée). L'estimateur obtenu par cette méthode a de bonnes propriétés mathématiques[8] : il est convergent, asymptotiquement efficient et asymptotiquement distribué suivant une loi normale. C'est-à-dire qu'il converge vers un estimateur sans biais avec la covariance minimum (il atteint la borne de Cramer-Rao) et ce, en suivant une loi normale.

Les principaux désavantages sont :

- les expressions peuvent être difficiles à obtenir et le calcul de l'estimateur dérivé par la méthode est propre au problème résolu.
- l'estimateur peut être biaisé pour de petits échantillons.

La fonction de vraisemblance est définie comme suit :

$$L_{\mathbf{y}}(\mathbf{x}) = p(\mathbf{y}|\mathbf{x}) \quad (1.20)$$

avec \mathbf{y} un vecteur d'observations, \mathbf{x} un vecteur de paramètres et p la densité de probabilité de \mathbf{y} pour les paramètres \mathbf{x} de la distribution.

Le maximum de vraisemblance consiste à trouver la fonction f^{ML} appelée estimateur qui maximisera le critère :

$$\hat{\mathbf{x}} = f^{ML}(\mathbf{y}) = \arg \max_{\mathbf{x}} L_{\mathbf{y}}(\mathbf{x}) = \arg \max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}). \quad (1.21)$$

Cette fonction sera donc trouvée en résolvant l'équation :

$$\frac{d}{d\mathbf{x}} L_{\mathbf{y}}(\mathbf{x}) = 0 \text{ et vérifiant que } \frac{d^2}{d^2\mathbf{x}} L_{\mathbf{y}}(\mathbf{x}) \text{ est définie négative.} \quad (1.22)$$

Si les paramètres sont indépendants, $p(\mathbf{y}|\mathbf{x})$ peut se réécrire sous la forme d'un produit de facteur. Pour simplifier les calculs, on utilise plutôt le log-vraisemblance et on cherche à minimiser la fonction $-\ln L_{\mathbf{y}}(\mathbf{x})$, et ce d'autant plus dans le cas d'un modèle gaussien ou exponentiel.

Moindres carrés

La technique des moindres carrés est utilisée dans les cas de problèmes sur-contraints, où l'on dispose de plus de données que d'inconnues.

On cherche à estimer \mathbf{x} tel que $A(\mathbf{x}, \mathbf{z}) = \mathbf{b}$ avec \mathbf{z} un vecteur de paramètres connu. On observe un ensemble bruité de réalisation \mathbf{b}^i de \mathbf{b} . On note $\mathbf{y} = \begin{bmatrix} \dots \\ \mathbf{b}^i \\ \dots \end{bmatrix}$ la concaténation des \mathbf{b}^i et $F : \mathbf{x} \mapsto \begin{bmatrix} \dots \\ A(\mathbf{x}, \mathbf{z}^i) \\ \dots \end{bmatrix}$ est la concaténation des fonctions A .

Dans l'estimation aux moindres carrés, on cherche les paramètres dont le carré de l'erreur aux données sera minimal.

L'estimateur aux moindres carrés est donc le suivant :

$$\hat{\mathbf{x}} = f^{MC}(\mathbf{y}) = \arg \min_{\mathbf{x}} \mathbf{e}^T \mathbf{e} = \arg \min_{\mathbf{x}} \|\mathbf{y} - F(\mathbf{x})\|_2^2 \quad (1.23)$$

$$\mathbf{e} = (\mathbf{y} - F(\mathbf{x})) \quad (1.24)$$

Il est intéressant de noter que dans le cas où F est linéaire et les mesures y sont entachées d'une erreur gaussienne, l'estimateur aux moindres carrés est aussi le maximum de vraisemblance.

Dans le cas où A est linéaire, ce problème revient à résoudre les équations normales suivantes :

$$\begin{pmatrix} J_x^{FT} & J_x^F \end{pmatrix} \hat{\mathbf{x}}^{MC} = -J_x^{FT} \mathbf{e} \quad (1.25)$$

Ce système peut être résolu en utilisant la décomposition de Cholesky de l'approximation (de Gauss-Newton) $\begin{pmatrix} J_x^{FT} & J_x^F \end{pmatrix}$ de la hessienne de $\mathbf{x} \mapsto \frac{1}{2}e^T e$ et en résolvant les deux systèmes triangulaires obtenus.

Dans le cas non linéaire, il faut alors utiliser une procédure d'optimisation pour calculer le résultat. Des méthodes couramment utilisées sont l'algorithme de Levenberg-Marquardt ou la descente de gradient conjuguée si le système est de dimension importante.

1.2.2 Optimisation

Dans cette section, nous présenterons les techniques d'optimisation qui seront utilisées dans la suite de cette thèse, notamment pour l'ajustement de faisceaux. Le but de ces techniques est de trouver un minimiseur d'une fonction de coût g suivant un jeu de paramètres \mathbf{x} :

$$\min_{\tilde{\mathbf{x}}} g(\tilde{\mathbf{x}}) \quad (1.26)$$

Dans le cas des moindres carrés, cette fonction g peut se réécrire sous forme d'une somme de carrés :

$$g(\mathbf{x}) = \frac{1}{2} \mathbf{e}(\mathbf{x})^T \mathbf{e}(\mathbf{x}). \quad (1.27)$$

Descente de gradient

Il s'agit de la plus simple des techniques d'optimisation itérative. Elle consiste, à chaque itération, à trouver la direction qui fait le plus décroître le coût et à avancer d'un pas dans cette direction. Cette direction est donnée par l'opposé du gradient de la fonction de coût au point \mathbf{x}^{t-1} . Ainsi :

$$\mathbf{x}^t = \mathbf{x}^{t-1} - \delta^t J_x^g(\mathbf{x}^{t-1})^T \quad (1.28)$$

La longueur du pas δ est un réel positif modifié au cours des itérations, et qui est obtenu, par exemple, par recherche linéaire [11]. Du choix de ce pas dépend la vitesse de convergence.

Cette méthode itérative à l'avantage de converger vers un minimum local même loin de la solution. Par contre, elle souffre de phénomènes oscillatoires proches de l'optimum. La technique du gradient conjugué [11] est toujours préférée à la technique de la descente de gradient pour limiter ce problème. Dans le cas d'un problème aux moindres carrés, la descente de gradient s'écrit :

$$J_{\mathbf{x}}^g(\mathbf{x}^{t-1})^T = J_{\mathbf{x}}^e(\mathbf{x}^{t-1})^T e(\mathbf{x}^{t-1}) \quad (1.29)$$

$$\mathbf{x}^t = \mathbf{x}^{t-1} - \delta^t J_{\mathbf{x}}^e(\mathbf{x}^{t-1})^T e(\mathbf{x}^{t-1}) \quad (1.30)$$

Newton-Raphson

Cette méthode est basée sur la méthode de Newton servant à trouver un zéro d'une fonction. Ici, on cherche à minimiser une fonction. On va donc chercher un zéro de son gradient. Pour cela, on utilise le processus itératif suivant :

$$\mathbf{x}^t = \mathbf{x}^{t-1} - (\mathbb{H}_{\mathbf{x}}^g(\mathbf{x}^{t-1}))^{-1} J_{\mathbf{x}}^g(\mathbf{x}^{t-1})^T \quad (1.31)$$

Cette méthode a l'avantage de converger rapidement près de l'optimum. Contrairement à la méthode précédente, il n'est plus nécessaire de chercher un pas optimal. Voici deux inconvénients majeurs de cette méthode. Premièrement, elle nécessite de calculer les dérivées secondes de la fonction de coût, ce qui peut être difficile et long à obtenir. Deuxièmement, elle peut diverger si l'on est loin de l'optimum.

Gauss-Newton La méthode de Gauss-Newton est une approximation de la méthode précédente qui utilise l'approximation éponyme pour le calcul du hessien dans le cas des moindres carrés ($\mathbb{H}^g \approx J^e T J^e$). Elle évite ainsi le calcul des dérivées du second ordre.

$$\mathbf{x}^t = \mathbf{x}^{t-1} - \left[J_{\mathbf{x}}^e(\mathbf{x}^{t-1})^T J_{\mathbf{x}}^e(\mathbf{x}^{t-1}) \right]^{-1} J_{\mathbf{x}}^e(\mathbf{x}^{t-1})^T e(\mathbf{x}^{t-1}) \quad (1.32)$$

Cette méthode est particulièrement utilisée dans le cadre des moindres carrés non linéaire, où elle revient à résoudre à chaque itération une équation normale résultante de la linéarisation du problème autour du point courant.

Levenberg-Marquardt

La technique d'optimisation de Levenberg-Marquardt(LM) est utilisée pour la résolution de problème de moindres carrés non linéaire et combine l'avantage

des deux formulations précédentes (la descente de gradient et Gauss-Newton) par l'utilisation d'un coefficient de pondération.

En effet, le Levenberg-Marquardt est une descente de gradient loin de l'optimum et utilise la technique de Gauss-Newton proche de celui-ci afin d'éviter les oscillations dues à la méthode de descente.

L'itération s'écrit :

$$\left(J_{\mathbf{x}}^e(\mathbf{x}^{t-1})^T J_{\mathbf{x}}^e(\mathbf{x}^{t-1}) + \lambda \mathbf{I} \right) \delta_{\mathbf{x}} = -J_{\mathbf{x}}^e(\mathbf{x}^{t-1})^T e(\mathbf{x}^{t-1}) \quad (1.33)$$

$$\mathbf{x}^t = \mathbf{x}^{t-1} + \delta_{\mathbf{x}} \quad (1.34)$$

La valeur de λ permet de régler le poids de la descente par rapport à Gauss-Newton. En général, il est diminué au cours des itérations.

1.2.3 Covariance

La matrice de variance/covariance d'un vecteur aléatoire est définie par :

$$C_{\mathbf{x}} = E((\mathbf{x} - E(\mathbf{x}))(\mathbf{x} - E(\mathbf{x}))^T) \quad (1.35)$$

$$= E(\mathbf{x}\mathbf{x}^T) - E(\mathbf{x})E(\mathbf{x})^T \quad (1.36)$$

avec $E(\mathbf{x})$ l'espérance de \mathbf{x} .

La matrice de covariance croisée entre deux vecteurs aléatoires est définie par :

$$C_{\mathbf{x}/\mathbf{y}} = E((\mathbf{x} - E(\mathbf{x}))(\mathbf{y} - E(\mathbf{y}))^T) \quad (1.37)$$

$$= E(\mathbf{x}\mathbf{y}^T) - E(\mathbf{x})E(\mathbf{y})^T \quad (1.38)$$

Des propriétés sont :

1. $C_{\mathbf{x}/\mathbf{x}} = C_{\mathbf{x}}$
2. $var(a) = C_{a/a}$
3. $C_{\mathbf{x}/\mathbf{y}} = C_{\mathbf{y}/\mathbf{x}}^T$
4. $C_{\mathbf{Ax}/\mathbf{By}} = \mathbf{A}C_{\mathbf{x}/\mathbf{y}}\mathbf{B}^T$
5. $C_{\mathbf{x}}$ est une matrice symétrique positive

L'inverse de la matrice de covariance (si elle existe) est appelé matrice d'information. C'est elle aussi une matrice définie positive.

1.2.4 Gaussienne, loi normale et modèle gaussien

Nous notons $X \sim \mathcal{N}(\bar{X}, \sigma^2)$ le fait qu'une variable aléatoire X suit une loi normale de moyenne \bar{X} et de variance σ^2 et donc sa fonction de densité de probabilité est égale à :

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{X})^2}{2\sigma^2}} \quad (1.39)$$

De même, nous notons $\mathbf{X} \sim \mathcal{N}(\bar{\mathbf{X}}, \Sigma)$ le fait qu'un vecteur aléatoire \mathbf{X} suit une loi normale de moyenne $\bar{\mathbf{X}}$ et de matrice de variance-covariance Σ :

$$p(\mathbf{x}) = \frac{1}{\det(\Sigma)^{\frac{|\mathbf{x}|}{2}} (2\pi)^{\frac{|\mathbf{x}|}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{X}})^T \Sigma^{-1} (\mathbf{x}-\bar{\mathbf{X}})} \quad (1.40)$$

Ellipse d'incertitude

L'ellipse ou ellipsoïde d'incertitude permet de représenter graphiquement les matrices de covariance des vecteurs aléatoires gaussiens. Pour cela, on utilise la notion d'intervalle de confiance.

Si \mathbf{x} est un vecteur aléatoire gaussien de moyenne $\bar{\mathbf{x}}$ et de covariance $C_{\mathbf{x}}$, alors $(\mathbf{x} - \bar{\mathbf{x}})^T C_{\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \sim \chi_{|\mathbf{x}|}^2$ avec $\chi_{|\mathbf{x}|}^2$ la loi de χ^2 à $|\mathbf{x}|$ degrés de liberté (car c'est une somme de $|\mathbf{x}|$ gaussiennes centrées réduites indépendantes).

L'intervalle de confiance définit un intervalle dans l'espace des paramètres qui correspondent à la probabilité qu'un élément tiré au hasard suivant une loi donnée soit dans cette zone. Ainsi, l'intervalle de confiance à 90%, pour le vecteur gaussien \mathbf{x} précédemment défini, sera :

$$(\mathbf{x} - \bar{\mathbf{x}})^T C_{\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) < \chi_{|\mathbf{x}|,0.1}^2 \quad (1.41)$$

Dans le cas d'un vecteur aléatoire de dimension 3, la valeur du seuil $\chi_{|\mathbf{x}|,0.1}^2$ peut être trouvée dans une table et vaut 6.25.

On remarque que l'inégalité 1.41 est l'équation d'un ellipsoïde dont la longueur des demi-axes est la racine carrée des valeurs propres de $\chi_{|\mathbf{x}|,0.1}^2 C_{\mathbf{x}}$ et dont les axes sont orientés par les vecteurs propres de $\chi_{|\mathbf{x}|,0.1}^2 C_{\mathbf{x}}$.

1.2.5 Complément de Schur, inversion par bloc et probabilité

Soit une matrice M symétrique¹ inversible. On partitionne M comme suit :

$$M = \begin{pmatrix} U & W^T \\ W & V \end{pmatrix} \quad (1.42)$$

Le complément de Schur de V est la quantité $S_V = U - WW^{-1}W^T$. Ce complément est important pour l'inversion des matrices par blocs.

Inversion par bloc Si M est symétrique inversible alors :

$$M^{-1} = \begin{pmatrix} U & W^T \\ W & V \end{pmatrix}^{-1} = \begin{pmatrix} S_V^{-1} & -S_V^{-1}W^TV^{-1} \\ -V^{-1}WS_V^{-1} & V^{-1}WS_V^{-1}W^TV^{-1} \end{pmatrix} \quad (1.43)$$

Cette formule d'inversion est très utile en pratique si V est de grande dimension et s'inverse facilement. Dans notre cas, V sera de grande taille et diagonale par blocs. Ces formules permettent aussi de calculer une partie de l'inverse de la matrice.

Lois marginales/conditionnelles, et leurs covariances/matrices d'information

Soient deux vecteurs aléatoires \mathbf{a} et \mathbf{b} tels que :

$$[\mathbf{ab}] \sim \mathcal{N}([\bar{\mathbf{ab}}], C_{[\mathbf{ab}]})$$

Notons :

$$C_{[\mathbf{ab}]} = \begin{pmatrix} C_{\mathbf{a}/\mathbf{a}} & C_{\mathbf{a}/\mathbf{b}} \\ C_{\mathbf{b}/\mathbf{a}} & C_{\mathbf{b}/\mathbf{b}} \end{pmatrix} \quad (1.44)$$

Loi marginale :

\mathbf{a} est une loi marginale de $[\mathbf{ab}]$ et donc $\mathbf{a} \sim \mathcal{N}(\bar{\mathbf{a}}, C_{\mathbf{a}/\mathbf{a}})$. Marginaliser \mathbf{b} revient donc à supprimer les lignes et les colonnes correspondantes à \mathbf{b} .

Dans le cas où la matrice d'information $\Lambda_{[\mathbf{ab}]}$ de $[\mathbf{ab}]$ est connue, on retrouve le résultat précédent par le complément de Schur.

$$\Lambda_{\mathbf{a}} = (\Lambda_{\mathbf{a}/\mathbf{a}} - \Lambda_{\mathbf{a}/\mathbf{b}}\Lambda_{\mathbf{b}/\mathbf{b}}^{-1}\Lambda_{\mathbf{b}/\mathbf{a}}) = S_{\Lambda_{\mathbf{b}/\mathbf{b}}} \quad (1.45)$$

Marginaliser \mathbf{b} revient donc à calculer le complément de Schur $S_{\Lambda_{\mathbf{b}/\mathbf{b}}}$ de $\Lambda_{[\mathbf{ab}]}$.

1. Pour plus de clarté, nous donnons ces résultats pour des matrices symétriques. En effet, bien qu'ils soient généralisables pour des matrices inversibles quelconques, ils sont uniquement utilisés pour des matrices symétriques dans le cadre de cette thèse.

Loi conditionnelle :
la loi de \mathbf{a} sachant \mathbf{b} en connaissant la loi de $[\mathbf{ab}]$ est donnée par :

$$\mathbf{a}|\mathbf{b} \sim \mathcal{N}(\bar{\mathbf{a}} - C_{\mathbf{a}/\mathbf{b}}C_{\mathbf{b}/\mathbf{b}}^{-1}\bar{\mathbf{b}}, C_{\mathbf{a}/\mathbf{a}} - C_{\mathbf{a}/\mathbf{b}}C_{\mathbf{b}/\mathbf{b}}^{-1}C_{\mathbf{b}/\mathbf{a}}) \quad (1.46)$$

La covariance de la probabilité conditionnelle est donnée par le complément de Schur $S_{C_{\mathbf{b}/\mathbf{b}}}$. La probabilité conditionnelle pour la matrice d'information s'exprime directement : $\Lambda_{\mathbf{a}|\mathbf{b}} = \Lambda_{\mathbf{a}/\mathbf{a}}$. L'information de la loi conditionnelle est donnée directement en supprimant les lignes et les colonnes de $\Lambda_{[\mathbf{ab}]}$ correspondant à \mathbf{b} .

1.3 Propagation d'erreur

1.3.1 Introduction

La notion d'incertitude est à considérer dès que l'on travaille avec des données mesurées ou estimées. En effet, la seule donnée de la valeur mesurée peut conduire à de mauvaises interprétations ou conclusions. Pour éviter cela, il est nécessaire de fournir l'incertitude associée à cette donnée. Ceci permet de comparer des valeurs entre elles ou de tirer des conclusions. Pour cela, la propagation d'incertitude est une des solutions utilisées. Ici, nous proposons d'utiliser ces techniques pour la localisation par vision. De même, fournir une position sans en donner une incertitude, ne permet d'utiliser ces données de manière optimale. En effet, l'information de localisation est intéressante intrinsèquement, mais il est difficile de comparer/fusionner deux résultats sans une incertitude associée. Dans la section suivante, nous allons voir comment propager l'incertitude dans un système.

1.3.2 Principe

Le but de cette technique appelée propagation d'incertitude ou d'erreur est de déterminer la statistique des sorties d'un processus en connaissant le modèle de bruit des données d'entrée.

Cela nécessite de modéliser la distribution de bruit sur toutes les données du modèle. La plupart du temps, cette modélisation est effectuée de manière paramétrique, la distribution est assimilée à une loi statistique et définie par ses moments.

Le problème se pose ainsi :

- Soit \mathbf{y} , les entrées du système, un vecteur aléatoire de moyenne $\bar{\mathbf{y}}$ et de matrice de variance/covariance $C_{\mathbf{y}}$.
- Soit \mathbf{x} , les variables/données de sortie, un vecteur aléatoire de moyenne $\bar{\mathbf{x}}$ et de matrice de variance/covariance $C_{\mathbf{x}}$.
- Soit f la fonction modélisant le système et permettant de transformer \mathbf{y} en \mathbf{x} .

$$\mathbf{x} = f(\mathbf{y}) \quad (1.47)$$

Connaissant la distribution des variables d'entrée \mathbf{y} , il nous faut déterminer la distribution des variables de sortie \mathbf{x} .

Dans toute cette thèse, le modèle gaussien est utilisé pour les distributions d'entrée et de sortie. Il existe deux grandes familles de méthodes pour résoudre ce problème :

- Les méthodes de simulation : l'idée générale est de considérer le système utilisé comme une boîte noire. On lui donne plusieurs entrées connues de manière à déterminer la distribution de sortie. Ces méthodes ont l'avantage d'être "exactes" (pas de développement limité), de ne faire aucune hypothèse sur le modèle et d'être mises en oeuvre simplement. Nous utiliserons une méthode de cette famille, la méthode de Monte-Carlo, pour vérifier les résultats des méthodes analytiques développées. L'inconvénient de ces méthodes est le temps de calcul élevé. En effet, il est nécessaire d'effectuer un nombre important de tirages pour obtenir un résultat fiable et le temps pour chaque tirage dépend essentiellement du temps de traitement de f .
- Les méthodes analytiques : elles supposent le modèle du système connu et s'en servent pour propager analytiquement la distribution d'entrée vers la distribution de sortie. Ces méthodes ne sont pas forcément exactes car des approximations sont souvent effectuées pour simplifier les calculs. Cependant, une fois la formule de propagation établie, elles permettent de très rapidement propager l'incertitude. Il y a deux inconvénients à cette méthode : le modèle mathématique du système peut être compliqué (non linéaire, défini implicitement) et l'établissement de la formule de propagation d'incertitude est souvent non trivial. De plus, le modèle et les simplifications utilisés pour établir la formulation de la propagation peut entraîner des erreurs importantes. Ici, nous utiliserons une méthode consistant à linéariser au premier ordre le modèle du système pour effectuer la propagation. Cette méthode est aussi appelée méthode delta en statistique.

Méthode de simulation

La méthode de Monte-Carlo présentée ici permet de propager la distribution d'erreur des variables d'entrée vers les variables de sortie d'une fonction par échantillonnage.

La méthode de Monte-Carlo peut se résumer en trois points :

1. échantillonner la distribution d'entrée avec N tirages ;
2. appliquer la fonction f sur chaque tirage ;
3. estimer la moyenne et la variance de la distribution de sortie par les méthodes standard.

Nous allons maintenant détailler ces trois étapes de l'estimation, illustrées par les figures Fig. 1.1, Fig. 1.2 et Fig. 1.3.

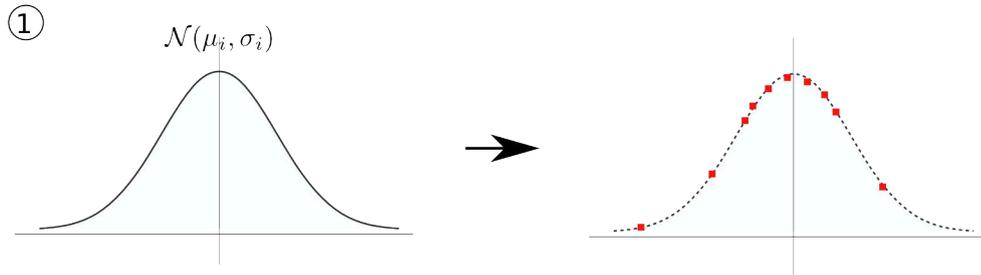


FIGURE 1.1: Échantillonnage , cas gaussien 1D.

Échantillonnage Le but de ce paragraphe est de montrer comment produire un ensemble de données suivant une loi donnée.

Nous voulons générer un ensemble de N réalisations \mathbf{y}^i du vecteur gaussien \mathbf{y} tel que $\mathbf{y} \sim \mathcal{N}(\bar{\mathbf{y}}, C_{\mathbf{y}})$. Pour cela, nous générons un ensemble de N réalisations \mathbf{u}^i d'un vecteur aléatoire \mathbf{u} suivant une loi gaussienne centrée réduite : $\mathbf{u} \sim \mathcal{N}(0_{|\mathbf{y}|}, I_{|\mathbf{y}|})$. Nous utilisons la méthode de Box-Muller [10] sur un générateur aléatoire uniforme pour générer les \mathbf{u}^i .

A partir de \mathbf{u} , nous construisons \mathbf{y} en effectuant les opérations suivantes :

$$\mathbf{y} = \bar{\mathbf{y}} + C_{\mathbf{y}}^{\frac{1}{2}} \mathbf{u} \quad (1.48)$$

avec $C_{\mathbf{y}}^{\frac{1}{2}}$ la racine carré de $C_{\mathbf{y}}$ obtenue par SVD ou par Cholesky.

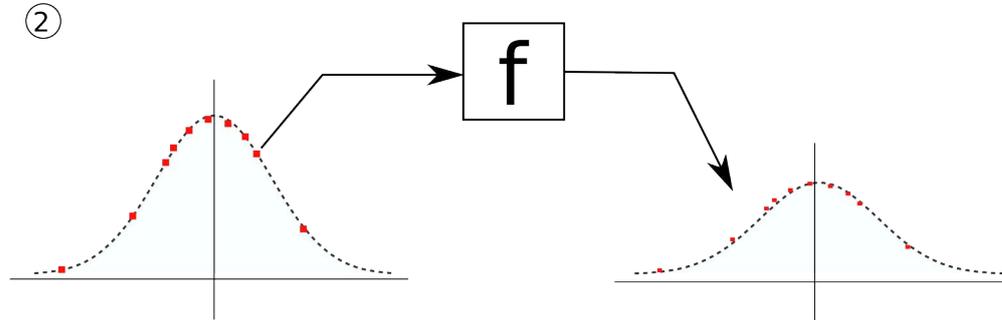


FIGURE 1.2: Propagation , cas gaussien 1D.

Propagation Maintenant que nous avons échantillonné la distribution d'entrée, nous pouvons calculer un échantillonnage de la distribution de sortie. Pour chaque y^i généré, nous calculons son image par la fonction f :

$$\forall i \in [1..N], \mathbf{x}^i = f(\mathbf{y}^i) \quad (1.49)$$

Les \mathbf{x}^i représentent l'échantillonnage de la distribution de sortie.

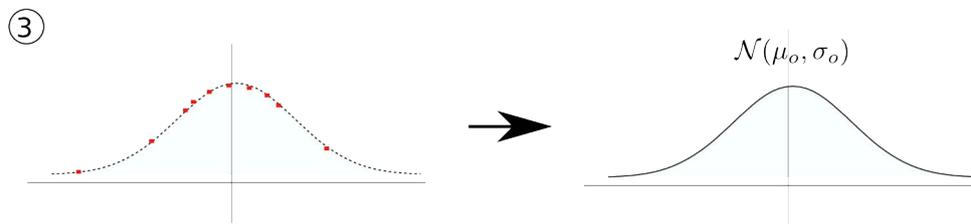


FIGURE 1.3: Estimation des paramètres de sortie, cas gaussien 1D.

Estimation des paramètres de la distribution de sortie Nous cherchons maintenant à estimer les paramètres de la distribution du vecteur gaussien \mathbf{x} .

Pour cela, nous utilisons des estimateurs de la moyenne et de la covariance. Ainsi :

$$\hat{x} = \frac{1}{N} \sum_i \mathbf{x}^i \quad (1.50)$$

et

$$\hat{C}_{\mathbf{x}} = \frac{1}{N-1} \sum_i (\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^i - \bar{\mathbf{x}})^T \quad (1.51)$$

Si \bar{x} n'est pas connue, nous utiliserons son estimation Eq. 1.50. Ainsi, nous avons déterminé la loi de la distribution de sortie \mathbf{x} en fonction la distribution d'entrée \mathbf{y} .

Méthode analytique

La propagation d'erreur au premier ordre est une méthode permettant de propager les erreurs des variables d'entrée vers les variables de sortie dans le cas où le modèle du système f est connu, même si celui-ci est non-linéaire ou implicite.

Pour cela, on utilise le développement de Taylor de la fonction f au premier ordre autour du point étudié ($\bar{\mathbf{y}}$) :

$$f(\mathbf{y}) = f(\bar{\mathbf{y}}) + J_{\mathbf{y}}(\mathbf{y} - \bar{\mathbf{y}}) + O(\|\mathbf{y} - \bar{\mathbf{y}}\|_2^2) \quad (1.52)$$

avec $J_{\mathbf{y}}^f$ la jacobienne de f par rapport à \mathbf{y} évaluée en $\bar{\mathbf{y}}$. La moyenne est donnée par :

$$\begin{aligned} E(\mathbf{x}) &= E(f(\mathbf{y})) \\ &\approx E(f(\bar{\mathbf{y}}) + J_{\mathbf{y}}(\mathbf{y} - \bar{\mathbf{y}})) \\ &\approx E(f(\bar{\mathbf{y}})) + J_{\mathbf{y}}(E(\mathbf{y}) - \bar{\mathbf{y}}) \\ &\approx f(\bar{\mathbf{y}}) \end{aligned} \quad (1.53)$$

La covariance est donnée par :

$$\begin{aligned} C_{x/x} &= C_{f(\mathbf{y})/f(\mathbf{y})} \\ &\approx C_{f(\bar{\mathbf{y}}) + J_{\mathbf{y}}^f(\mathbf{y} - \bar{\mathbf{y}})/f(\bar{\mathbf{y}}) + J_{\mathbf{y}}^f(\mathbf{y} - \bar{\mathbf{y}})} \\ &\approx C_{f(\bar{\mathbf{y}})/f(\bar{\mathbf{y}})} + C_{f(\bar{\mathbf{y}})/J_{\mathbf{y}}^f(\mathbf{y} - \bar{\mathbf{y}})} \\ &\quad + C_{J_{\mathbf{y}}^f(\mathbf{y} - \bar{\mathbf{y}})/f(\bar{\mathbf{y}})} + C_{J_{\mathbf{y}}^f(\mathbf{y} - \bar{\mathbf{y}})/J_{\mathbf{y}}^f(\mathbf{y} - \bar{\mathbf{y}})} \\ &\approx J_{\mathbf{y}}^f C_{\mathbf{y} - \bar{\mathbf{y}}/\mathbf{y} - \bar{\mathbf{y}}} J_{\mathbf{y}}^{fT} \\ &\approx J_{\mathbf{y}}^f C_{\mathbf{y}/\mathbf{y}} J_{\mathbf{y}}^{fT} \end{aligned} \quad (1.54)$$

Et ainsi :

$$\bar{\mathbf{x}} \approx f(\bar{\mathbf{y}}) \quad (1.55)$$

$$C_{\mathbf{x}} \approx J_{\mathbf{y}}^f C_{\mathbf{y}} J_{\mathbf{y}}^{fT} \quad (1.56)$$

De plus, si $[\mathbf{z}, \mathbf{y}]$ suit une loi jointe de moyenne $[\bar{\mathbf{z}}, \bar{\mathbf{y}}]$ et de covariance $C_{[\mathbf{z}, \mathbf{y}]} = \begin{pmatrix} C_{\mathbf{z}/\mathbf{z}} & C_{\mathbf{z}/\mathbf{y}} \\ C_{\mathbf{y}/\mathbf{z}} & C_{\mathbf{y}/\mathbf{y}} \end{pmatrix}$ et si f ne dépend pas de \mathbf{z} ($\mathbf{x} = f(\mathbf{y})$), alors on peut calculer la

corrélation entre \mathbf{x} et \mathbf{z} :

$$\begin{aligned}
 C_{\mathbf{x}/\mathbf{z}} &= C_{f(\mathbf{y})/\mathbf{z}} \\
 &\approx C_{f(\bar{\mathbf{y}})+J_{\bar{\mathbf{y}}}^f(\mathbf{y}-\bar{\mathbf{y}})/\mathbf{z}} \\
 &\approx C_{f(\bar{\mathbf{y}})/\mathbf{z}} + C_{J_{\bar{\mathbf{y}}}^f(\mathbf{y}-\bar{\mathbf{y}})/\mathbf{z}} \\
 &\approx J_{\bar{\mathbf{y}}}^f C_{(\mathbf{y}-\bar{\mathbf{y}})/\mathbf{z}} \\
 &\approx J_{\bar{\mathbf{y}}}^f C_{\mathbf{y}/\mathbf{z}}.
 \end{aligned} \tag{1.57}$$

Et ainsi :

$$C_{[\mathbf{z},\mathbf{x}]} = \begin{pmatrix} C_{\mathbf{z}/\mathbf{z}} & C_{\mathbf{z}/\mathbf{y}} J_{\bar{\mathbf{y}}}^{fT} \\ J_{\bar{\mathbf{y}}}^f C_{\mathbf{y}/\mathbf{z}} & J_{\bar{\mathbf{y}}}^f C_{\mathbf{y}/\mathbf{y}} J_{\bar{\mathbf{y}}}^{fT} \end{pmatrix} \tag{1.58}$$

1.3.3 Linéarisation d'une fonction définie implicitement

Soit f une fonction définie implicitement par :

$$\mathbf{x} = f(\mathbf{y}) = \arg \min_{\mathbf{z}} (e(\mathbf{y}, \mathbf{z})). \tag{1.59}$$

Le résultat suivant peut être obtenu soit en utilisant le théorème des fonctions implicites (chapitre 5 de [40]), soit par linéarisation (§5 de [45]). On présente ici le résultat utilisant le théorème des fonctions implicites :

- Soit :
 - une fonction $e : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ de classe C^∞ .
 - $\mathbf{y}_0 \in \mathbb{R}^m$ un vecteur de mesure
 - $\mathbf{x}_0 \in \mathbb{R}^p$ un minimum local de $\mathbf{z} \mapsto e(\mathbf{y}_0, \mathbf{z})$.
- Si la hessienne H de e par rapport à \mathbf{z} est inversible en $(\mathbf{y}, \mathbf{z}) = (\mathbf{y}_0, \mathbf{x}_0)$.
- Alors, il existe :
 - un ouvert U' de \mathbb{R}^m contenant \mathbf{y}_0
 - un ouvert U'' de \mathbb{R}^p contenant \mathbf{x}_0
 - une fonction $f : C^\infty$ de $U' \rightarrow U''$
 Tels que :
 - $\forall (\mathbf{y}, \mathbf{x}) \in U' \times U''$ les deux relations suivantes sont équivalentes :
 - “ \mathbf{x} est un minimum local de $\mathbf{z} \mapsto e(\mathbf{y}_0, \mathbf{z})$ par rapport à \mathbf{z} ”
 - “ $\mathbf{x} = f(\mathbf{y})$ ”
 De plus nous avons l’expression suivante de la différentielle de f :

$$\frac{\partial f}{\partial \mathbf{y}} = -H^{-1} \frac{\partial \Phi}{\partial \mathbf{y}} \quad (1.60)$$

$$\text{avec : } \Phi = \frac{\partial e^T}{\partial \mathbf{z}} \quad (1.61)$$

$$H = \frac{\partial \Phi}{\partial \mathbf{z}} \quad (1.62)$$

Application au cas des moindres carrés :

Dans la suite de cette thèse, nous aurons besoin d’appliquer ce résultat à la fonction implicite suivante :

$$\mathbf{x} = f(\mathbf{y}) = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{y} - F(\mathbf{z}, \mathbf{p})\|_2^2 \quad (1.63)$$

Dans cette fonction, \mathbf{y} est le vecteur des mesures et \mathbf{p} est un vecteur de paramètres de F . Nous allons donc appliquer le résultat précédent à cette fonction pour obtenir les dérivées de f par rapport à \mathbf{y} et \mathbf{p} .

Dans un premier temps, nous calculons les dérivées de f par rapport à \mathbf{y} avec un paramètre \mathbf{p} constant. La fonction e s’écrit ainsi :

$$(\mathbf{y}, \mathbf{z}) \mapsto e(\mathbf{y}, \mathbf{z}) = \frac{1}{2} (\mathbf{y} - F(\mathbf{z}, \mathbf{p}))^T (\mathbf{y} - F(\mathbf{z}, \mathbf{p})) \quad (1.64)$$

d'où :

$$\begin{aligned}\Phi &= \frac{\partial e^T}{\partial \mathbf{z}} \\ &= -\frac{\partial(F(\mathbf{z}, \mathbf{p}))^T}{\partial \mathbf{z}} (\mathbf{y} - F(\mathbf{z}, \mathbf{p}))\end{aligned}\quad (1.65)$$

$$\begin{aligned}\mathbf{H} &= \frac{\partial \Phi}{\partial \mathbf{z}} \\ &\approx \frac{\partial F(\mathbf{z}, \mathbf{p})^T}{\partial \mathbf{z}} \frac{\partial F(\mathbf{z}, \mathbf{p})}{\partial \mathbf{z}}\end{aligned}\quad (1.66)$$

$$\frac{\partial \Phi}{\partial \mathbf{y}} = \frac{\partial}{\partial \mathbf{y}} \left(-\frac{\partial(F(\mathbf{z}, \mathbf{p}))^T}{\partial \mathbf{z}} (\mathbf{y} - F(\mathbf{z}, \mathbf{p})) \right)\quad (1.67)$$

$$= -\frac{\partial(F(\mathbf{z}, \mathbf{p}))^T}{\partial \mathbf{z}}\quad (1.68)$$

On note ici, que pour le calcul du hessien \mathbf{H} , nous avons utilisé l'approximation de Gauss-Newton pour éviter le calcul des dérivées secondes.

Dans la propagation d'erreur, nous aurons besoin de connaître la dérivée de f par rapport à \mathbf{p} . Dans ce cas \mathbf{y} est considéré comme constant. Nous appliquons donc le théorème précédent à la fonction :

$$(\mathbf{p}, \mathbf{z}) \mapsto e(\mathbf{y}, \mathbf{z}) = \frac{1}{2}(\mathbf{y} - F(\mathbf{z}, \mathbf{p}))^T (\mathbf{y} - F(\mathbf{z}, \mathbf{p}))\quad (1.69)$$

On obtient :

$$\frac{\partial \Phi}{\partial \mathbf{p}} = \frac{\partial}{\partial \mathbf{p}} \left(-\frac{\partial(F(\mathbf{z}, \mathbf{p}))^T}{\partial \mathbf{z}} (\mathbf{y} - F(\mathbf{z}, \mathbf{p})) \right)\quad (1.70)$$

$$\approx \frac{\partial(F(\mathbf{z}, \mathbf{p}))^T}{\partial \mathbf{z}} \frac{\partial(F(\mathbf{z}, \mathbf{p}))}{\partial \mathbf{p}}\quad (1.71)$$

Pour résumer, nous avons établi les formules suivantes en notation abrégée :

$$J_{\mathbf{y}}^f = (J_{\mathbf{z}}^{F^T} J_{\mathbf{z}}^F)^{-1} J_{\mathbf{z}}^{F^T}\quad (1.72)$$

$$J_{\mathbf{p}}^f = -(J_{\mathbf{z}}^{F^T} J_{\mathbf{z}}^F)^{-1} J_{\mathbf{z}}^{F^T} J_{\mathbf{p}}^F\quad (1.73)$$

1.3.4 Définition d'une contrainte de jauge

Nous avons vu précédemment que, pour propager la covariance de l'entrée vers la sortie d'une fonction, nous pouvons linéariser cette fonction au premier

ordre et que le calcul de la matrice jacobienne est alors nécessaire. Nous avons ensuite présenté comment calculer ce jacobien dans le cas d'une fonction définie implicitement. L'existence de la fonction implicite nécessite que le système la définissant soit entièrement déterminé. Dans le cas contraire, nous ne pourrions pas définir de covariance.

L'idée ici, dans le cas d'un système indéterminé avec une infinité de solutions, est d'ajouter des contraintes supplémentaires de manière à rendre ce système déterminé. Dans le cadre de la propagation d'erreur, ces contraintes seront appelées contraintes de jauge.

Il est à noter que le choix de la jauge dans laquelle est exprimée l'incertitude est important, car les valeurs et les formes des ellipses d'incertitude varient fortement suivant ce choix. Pour plus d'information à ce sujet, le lecteur peut se référer aux travaux de Morris *et al.* et Kanatani *et al.* ([77],[78],[58]) qui se sont attachés à étudier l'influence du choix de la jauge et les techniques permettant de transformer les incertitudes exprimées d'une jauge dans une autre. Ici, nous nous contenterons de contraintes triviales qui consistent à figer le nombre nécessaire de paramètres.

Dans le cas de la reconstruction 3D, la reconstruction est déterminée à une origine, une orientation et une échelle près. Il est donc nécessaire de fixer 7 paramètres pour obtenir un système entièrement déterminé et l'existence de la fonction implicite.

Dans le but d'utiliser cette reconstruction pour la localisation par vision, nous choisirons de fixer les paramètres de la première pose et la coordonnée ayant le plus varié d'une autre pose (dans notre cas la coordonnée en z de la 10^{ème} pose). Les covariances seront donc définies par rapport au début de la trajectoire. Cette contrainte triviale est appliquée simplement en supprimant les lignes ou les colonnes qui correspondent à ces paramètres dans la matrice jacobienne et la hessienne. Ainsi, le hessien devient inversible, la jacobienne de rang plein, et la covariance peut-être calculée.

Plus de détails peuvent être trouvés sur la manière d'imposer des contraintes plus compliquées dans [102] et [70] pour la reconstruction 3D.

Chapitre 2

Etat de l'art : SLAM et propagation d'erreur en vision

Sommaire

2.1	Introduction	30
2.2	localisation et reconstruction simultanées	30
2.2.1	Ajustement de faisceau et Full-SLAM	31
2.2.2	Approches utilisant le filtrage	32
2.2.3	Slam relatif	33
2.3	Propagation d'erreur en vision 3D	34

Ce chapitre présente un état de l'art sur les techniques de SLAM et de propagation d'incertitude en vision par ordinateur. Une attention particulière est portée à la localisation par vision monoculaire.

2.1 Introduction

Dans la suite de ce chapitre, nous détaillerons les techniques utilisées en SLAM et les différentes approches existantes. Puis dans un second temps, nous verrons comment la propagation d'erreur peut s'appliquer pour des techniques de vision par ordinateur.

2.2 localisation et reconstruction simultanées

Les techniques de localisation par vision ont été étudiées depuis longtemps, de deux points de vues différents, et avec deux vocabulaires différents.

D'un côté, il y a l'approche des roboticiens et celle du SLAM ("Simultaneous Localisation And Mapping", localisation et cartographie simultanée en français) dont le but est de localiser un robot dans son environnement sans *a priori* sur celui-ci. La carte doit donc être reconstruite en même temps que la position doit être déterminée. Ce problème a été formalisé dans les années 80 ([95],[81]). De cette modélisation probabiliste du problème découlent notamment les techniques utilisant le filtre de Kalman [55]. Il a été montré théoriquement que le problème du SLAM converge [22] à condition de fermer des boucles dans l'exploration de l'environnement (cela consiste à observer à nouveau des points de l'environnement déjà cartographiés) et ainsi à incrémentalement augmenter la précision de la carte. Historiquement, cette approche fusionne l'information provenant des nombreux capteurs présents sur les robots ([29],[3]). Avec l'augmentation des puissances de calcul, les roboticiens se sont intéressés aux caméras et à l'information contenue dans l'image et ont adapté leurs techniques à la vision.

De l'autre côté, il y a l'approche de la communauté de la vision par ordinateur qui s'appuie sur des travaux existants en photogrammétrie. Ici, on parle de Structure-from-Motion (parfois traduit par stéréo-mouvement). Cette technique ([49],[40]) consiste à reconstruire une scène à partir d'un ensemble d'images prises depuis une position inconnue, éventuellement avec une caméra non calibrée. Le problème est donc similaire à celui du SLAM : il est aussi nécessaire de déterminer simultanément les positions des prises de vues et l'environnement sous forme d'une reconstruction dense ou éparse. Le structure-from-motion utilise fortement les relations géométriques et l'ajustement de faisceaux pour obtenir une localisation et une reconstruction correcte.

2.2.1 Ajustement de faisceau et Full-SLAM

Historiquement, en vision par ordinateur et en photogrammétrie, le but était de reconstruire l'environnement. L'acquisition des données était faite dans une phase préalable au traitement (ces méthodes sont dénommées hors-ligne). Les images provenaient de photographies, et chacune était soigneusement prise pour un résultat optimum. L'ajustement de faisceau est la technique utilisée pour estimer la pose des caméras (position et orientation) et la représentation de l'environnement, et ce, à partir de toutes les mesures 2D. C'est pour cette raison qu'il est aussi référencé sous le terme full-Slam dans la communauté robotique ([26],[92]).

Puis, pour automatiser le processus de reconstruction, des vidéos sont directement utilisées plutôt qu'un ensemble épars de photographies. Un prétraitement est alors utilisé pour sélectionner les images intéressantes [91] (appelées images clefs).

Des travaux récents ([96],[2],[87]) ont montré que cette technique permet d'obtenir des reconstructions de très bonne qualité à grande échelle (une ville).

L'ajustement de faisceaux est une technique fortement coûteuse en temps de calcul et d'espace mémoire, et est le plus souvent inapplicable pour des applications embarquées. Les méthodes hiérarchiques permettent de réduire le nombre et la taille des ajustements de faisceaux effectués pour traiter l'ajout d'information ([86],[43]).

Voici des résultats récents en faveur de l'ajustement de faisceaux :

- une comparaison entre les différentes méthodes montre que l'utilisation de l'ajustement de faisceau donne de meilleurs résultats que le filtre de Kalman lors de la résolution du problème du SLAM [97].
- l'utilisation de la structure creuse de second niveau permet d'augmenter fortement les performances des méthodes d'ajustement de faisceaux. Cette approche a été présentée dans deux articles [52], [63]. Il existe aussi d'autres méthodes plus anciennes [102] pour exploiter la structure creuse de second niveau dans l'ajustement de faisceaux (Cholesky avec profil appliqué sur le système des caméras réduits).
- l'utilisation d'une forme factorisée de la matrice d'information par Dellaert *et al.* ([26],[54]) qui permet de calculer et de mettre à jour efficacement la matrice d'information et d'extraire les covariances nécessaires.

Toutes les méthodes précédentes sont hors-ligne et ne permettent pas de traiter de longues séquences en temps réel. De nouvelles méthodes de Structure-from-motion incrémentale sont apparues pour traiter ce cas (Nister [82], Engles *et al.* [32] et Mouragnon *et al.* [80]). Elles se basent sur les outils de la vision, adaptés

de manière à être utilisés de façon incrémentale et temps réels. Ces techniques ressemblent aux techniques de lissage ("smoothing") et au filtre informatif [92]. Les méthodes incrémentales ont montré de très bonnes performances et permettent de reconstruire des trajectoires de plusieurs kilomètres ([19],[62]) et elles peuvent être couplées avec une méthode de gestion de la carte en tache de fond [60].

Dans le cas de l'ajustement de faisceau, l'incertitude sur les paramètres estimés est obtenue par propagation d'erreur. Il est de plus possible d'obtenir cette incertitude dans le repère que l'on souhaite par l'utilisation des contraintes de jauge : on peut se référer au papier de Triggs [102] ou à la synthèse bibliographique proposée par Lhuillier *et al.* [70]. Il est possible de tenir compte en plus des erreurs sur les données celles provenant du calibrage des caméras [1]. L'estimation de l'incertitude pour l'ajustement de faisceau nécessite l'inversion d'une matrice de la taille du vecteur de paramètres. Le processus d'estimation de l'incertitude est donc coûteux. Il est néanmoins possible de tenir compte de la structure particulière de la matrice pour accélérer les calculs. Par ailleurs, les méthodes incrémentales ne fournissent pas d'incertitude absolue.

2.2.2 Approches utilisant le filtrage

Contrairement à l'approche précédente, historiquement, les techniques de filtrage utilisées en SLAM sont des méthodes en ligne. Le filtre de Kalman est tout d'abord utilisé dans sa version étendue (EKF). Dans les approches par filtrage chaque mesure est intégrée au fur et à mesure et l'ensemble des données n'est pas conservé, mais seulement la valeur courante des paramètres estimés ainsi que leur covariance. La version duale du filtre de Kalman est le filtre informatif (EIF), il stocke la matrice d'information plutôt que la matrice de covariance. Il a notamment l'avantage de permettre d'intégrer simultanément plusieurs mesures de manière efficace.

Toutes ces techniques ont été utilisées pour la localisation par vision en monoculaire [25],[66] utilisant l'EKF et [38] utilisant l'EIF.

Mais ces approches font des erreurs d'estimation de l'incertitude dues à la linéarisation autour de l'état courant des fonctions d'observation et du modèle de mouvement non linéaires utilisés. Ce sont ces erreurs qui peuvent devenir importantes lorsque l'état devient incertain (covariance importante). Des études ([4],[15]) ont été menées pour comprendre dans quel cas ces phénomènes arrivent. Trois types d'approches ont été proposées pour réduire les erreurs de linéarisation.

La première technique utilisée est le lissage ("smoothing"). Elle consiste à ne plus tenir compte uniquement de la position à l'état courant, mais aussi des

positions précédentes. Cela permet de réduire les erreurs de linéarisation et d'estimation [92].

La seconde technique est l'utilisation de particules plutôt qu'une linéarisation au premier ordre comme pour l'EKF. Elle permet de réduire les erreurs de linéarisation et d'adéquation au modèle Gaussien. La méthode utilisant cette technique est appelée filtre de Kalman sans parfum (UKF). L'UKF a été utilisé en vision monoculaire dans [17]. Une autre technique dérivée de la méthode de Monte-Carlo utilise les particules, il s'agit du filtre à particule ([27],[28]). Il est notamment utilisé dans FastSLAM [76], où l'estimation de la trajectoire et celle de la mise à jour de la carte sont séparées grâce à l'utilisation du théorème de Rao-Blackwell[14]. Cette idée a été utilisée pour le SLAM visuel monoculaire dans [30].

La dernière solution est l'approche par carte locale. L'utilisation de sous-cartes permet en effet de rester dans le domaine de fonctionnement du filtre de Kalman (où l'incertitude reste faible) et ainsi de limiter les problèmes de consistance [67], [84] (en vision monoculaire [85]).

Notons que les techniques utilisant le filtrage ont l'avantage de fournir non seulement les paramètres estimés, mais également leurs incertitudes.

2.2.3 Slam relatif

Contrairement aux deux approches précédentes, ici le paramétrage des poses des caméras et des points 3D n'est plus effectué dans un repère commun absolu. Dans les travaux de Sibley *et al.* [93], l'optimisation se fait via un ajustement de faisceaux, mais une caméra est paramétrée par la transformation entre cette caméra et la caméra précédente. De même, les coordonnées des points 3D sont exprimées dans le repère de la première caméra observant ce point. L'approche est démontrée sur l'une des plus longues séquences connues en SLAM (142km). Cette approche apporte également une solution élégante à la résolution de l'optimisation après une fermeture de boucle.

Ici on trouve aussi les méthodes qui construisent un graphe de poses puis les optimisent ([61],[98]).

Les positions réelles (non relative) des éléments peuvent être retrouvées, notamment en utilisant les techniques de relaxation de graphe de pose, par exemple la méthode de [44]. Ici, l'incertitude relative à un repère absolu (par exemple celui de la première caméra) est représentée par une chaîne d'incertitudes locales. L'incertitude globale peut être recalculée à partir de la chaîne, par exemple, avec la

méthode de Smith *et al.* [94].

2.3 Propagation d'erreur en vision 3D

La propagation d'incertitude a été très utilisée dans tous les domaines. Un guide ISO [7] a été publié pour faciliter l'utilisation de ces techniques. On retrouve l'utilisation de ces techniques dans toutes les autres communautés proches de la vision par exemple en géométrie [103] ou en imagerie médicale [41]. Elles sont très importantes en photogrammétrie et en métrologie : par exemple [21] où l'on cherche à connaître la précision des mesures effectuées dans l'image ou la qualité du modèle 3D reconstruit [18].

Dans le domaine de la vision, les travaux de Faugeras [40], Haralick [45], Kanatani [56] ont popularisé ces techniques. Il est ici à noter que l'estimation de l'incertitude est souvent effectuée en utilisant la propagation d'incertitude dans l'estimateur, qui est dans la plupart des cas un estimateur au maximum de vraisemblance ou aux moindres carrés. De plus, les processus d'estimation sont souvent chaînés, la sortie du premier devenant l'entrée d'un processus de plus haut niveau. Pour estimer l'incertitude, la propagation d'erreur sera aussi chaînée et il faut faire attention ici aux corrélations existantes entre les éléments.

Voici deux exemples.

Matrice fondamentale La propagation d'incertitude est d'abord utilisée pour les éléments de base comme la détection et l'extraction des primitives 2D : par exemple, pour l'extraction de points d'intérêt [13] ou des lignes, cercles et ellipses [104].

A partir des correspondances entre points d'intérêt, il est ensuite possible d'estimer la matrice fondamentale ou la matrice essentielle.

Les travaux de Csurka *et al.* [23] utilisent la propagation d'erreur pour estimer l'erreur effectuée sur la matrice fondamentale à partir de correspondance 2D/2D dans le cas d'une estimation par minimisation non linéaire. Cette incertitude a aussi été estimée pour d'autres algorithmes et paramétrisations ([100],[99],[106]). Plusieurs applications découlent de la connaissance de cette incertitude :

- tout d'abord, cette incertitude fournit un moyen de comparer les différentes méthodes d'estimation existantes (Zhang *et al.* [105], Engles *et al.* [31]) ;
- ensuite elle permet d'améliorer la mise en correspondance (brandt *et al.* [12]) en tenant compte de cette incertitude dans le cas où la transformation entre

les caméras est connue, mais imprécise (par exemple dans le cas d'une paire stéréo)

- enfin et surtout, elle permet d'estimer la qualité de l'estimation du mouvement de la caméra [101].

Calcul de pose Les travaux de Royer *et al.* [89], permettent d'estimer l'incertitude de la pose de la caméra. Contrairement au cas précédent, la caméra est relocalisée à partir d'un nuage de points 3D préalablement reconstruit. Cet article présente une manière de calculer l'incertitude de la pose de la caméra à partir du nuage de point 3D et leur covariance. Ces covariances doivent être préalablement évaluées. L'article propose et compare 4 méthodes plus ou moins approchées pour effectuer ce calcul de manière efficace.

Chapitre 3

Ajustement de faisceaux global

Sommaire

3.1	Introduction	38
3.2	Formulation	40
3.3	Résolution du système creux	41
3.4	Modèle de bruit	45
3.5	Propagation d'erreur	46
3.6	Résumé de l'ajustement de faisceau global	49

Ce chapitre présente une description de l'ajustement de faisceaux. Il introduit une formalisation mathématique de celui-ci et de la propagation d'erreur associée. Bien qu'il n'y ait aucune contribution sur cette méthode, les notions introduites ici sont essentielles à la bonne présentation et compréhension des méthodes d'ajustement de faisceaux locaux que nous allons présenter dans les chapitres suivants.

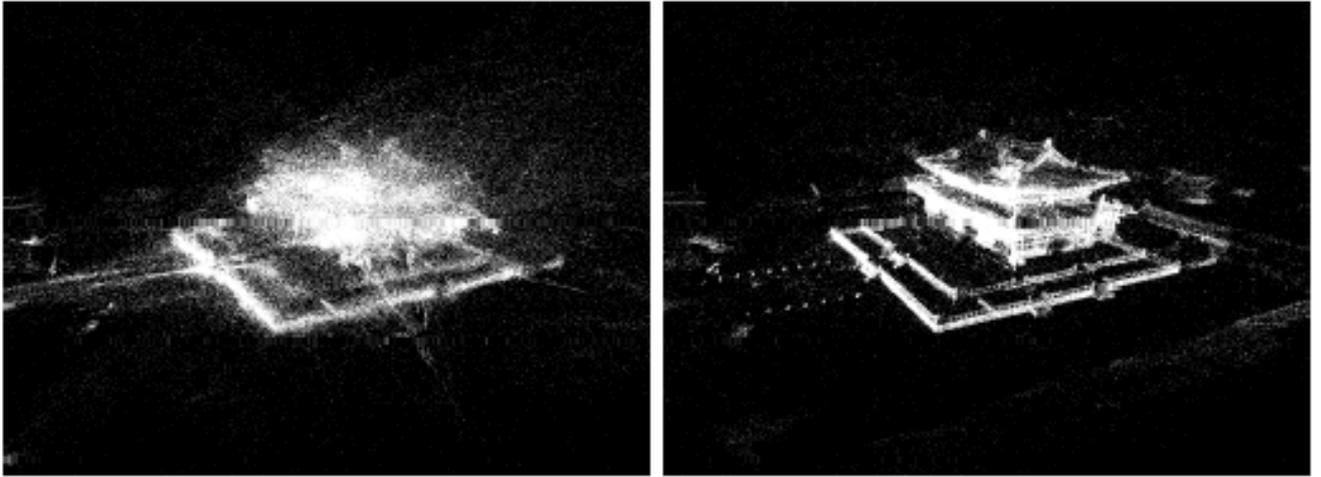


FIGURE 3.1: Exemple d'utilisation d'un ajustement de faisceaux pour la reconstruction de scène 3D. A gauche la reconstruction avant ajustement de faisceaux. A droite après l'ajustement. Cette image a été présentée par Yekeun Jeong *et al.* [52].

3.1 Introduction

L'ajustement de faisceaux est une technique provenant de la photogrammétrie qui est utilisée pour réduire le bruit et les incohérences géométriques dans une reconstruction 3D (voir figure 3.1). Il fut introduit dans le domaine de la vision dans les années 90. Depuis, il a été fortement utilisé et reste encore aujourd'hui étudié pour en améliorer les performances. Pour les différentes techniques d'implémentation et d'utilisation, le lecteur pourra se référer à Triggs *et al.* [102] ou pour un état des travaux récents dans ce domaine à Yekeun Jeong *et al.* [52].

L'ajustement de faisceaux permet de raffiner conjointement un ensemble de paramètres géométriques. Deux types de paramètres sont pris en considération :

- tout d'abord, l'ensemble des paramètres permettant de définir les caractéristiques du/des capteurs observant la scène au cours du temps. Ces paramètres sont composés de la position et de l'orientation, habituellement appelé pose, du ou des capteurs au cours du temps (paramètres extrinsèques) et des paramètres de calibrage inhérent à ces capteurs qui définissent la manière dont ils observent le monde (paramètres intrinsèques).
- ensuite, l'ensemble des paramètres de la scène pouvant être constitué d'un ensemble de primitives plus ou moins complexes (points 3D, lignes ...) ou d'une surface paramétrée (maillage) dans le cas d'une reconstruction dense.

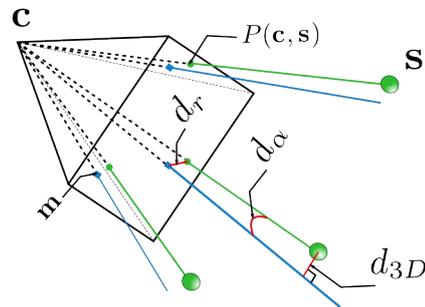


FIGURE 3.2: L’ajustement de faisceaux cherche à minimiser l’erreur entre les faisceaux 3D (en vert) et 2D (en bleu). L’erreur utilisée est soit directement l’erreur angulaire (d_α), soit l’erreur 3D (d_{3D}) ou plus classiquement l’erreur de reprojection dans l’image (d_r). Dans cette thèse, s , m , c sont respectivement un point 3D, un point 2D, une caméra et la fonction de projection est notée P .

Le nom ajustement de faisceaux vient du critère utilisé. En effet, il cherche à ajuster au mieux deux ensembles de rayons. Le premier est l’ensemble des rayons partant des primitives 3D et passant par le centre optique de la caméra. Le second est l’ensemble des rayons partant du centre optique de la caméra et passant par les primitives 2D (voir Fig.3.2).

Il existe plusieurs manières de mesurer l’adéquation entre ces deux faisceaux de rayons. Il est possible de mesurer l’angle entre les deux rayons. Cette erreur est utilisée dans le cas des modèles génériques de caméra [69]. Plus classiquement, dans le cas d’une caméra suivant un modèle sténopé, l’erreur de reprojection est utilisée. Elle a l’avantage d’être directement mesurable dans l’image [49]. Dans le cas de point 3D, l’erreur sera la distance euclidienne dans l’image entre la projection de la primitive 3D dans l’image et une observation de cette primitive par le capteur. Il est aussi possible de tenir compte de l’information photométrique [20]. Dans notre cas, nous nous intéresserons à la solution minimisant une erreur 2D image. Ainsi, conjointement, dans un même processus, le nuage de points 3D et un ensemble de poses de caméras vont être optimisés.

Cette optimisation est réalisée en minimisant l’erreur de reprojection entre le point 2D, observation du point 3D dans l’image, et la reprojection de ce point par la fonction de projection du capteur utilisé.

Cette erreur de reprojection s’écrit (voir figure 3.2) :

$$d_r = \|\mathbf{m} - P(\mathbf{c}, \mathbf{s})\|_2 \quad (3.1)$$

Cette erreur sera minimisée pour chaque couple caméra/point 3D dont une ob-

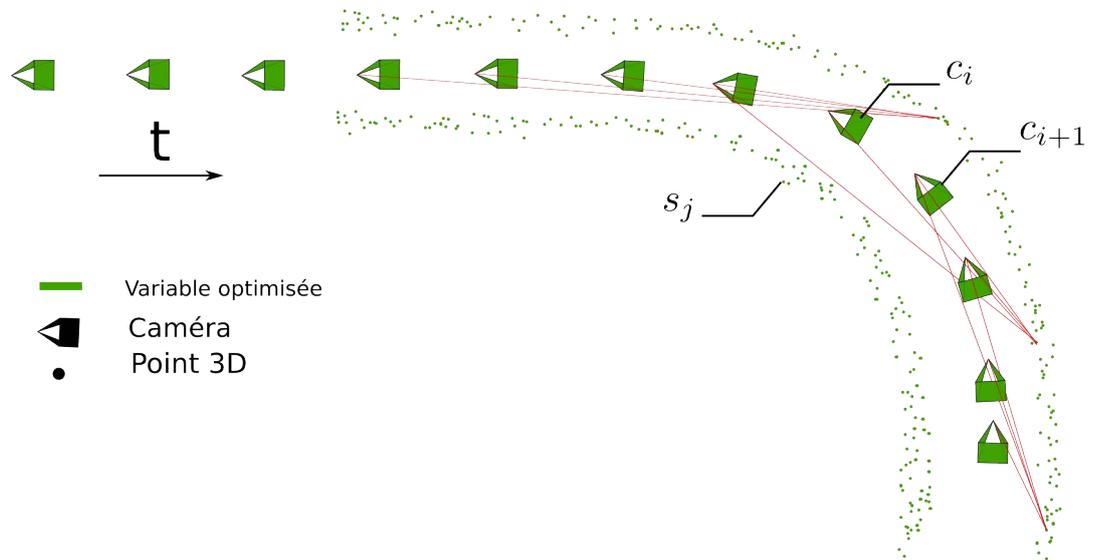


FIGURE 3.3: Exemple d'optimisation par AFG, présentant graphiquement les variables utilisées.

servation 2D a été trouvée dans l'image. Pour que le système soit suffisamment contraint, il est nécessaire d'avoir au minimum 3 observations de points 3D par caméra et d'avoir au minimum deux observations par point 3D. En pratique, ces chiffres sont largement dépassés, et le problème est alors surcontraint : on dispose de moins d'inconnues que d'équations. Une solution consiste à résoudre ce problème aux moindres carrés. Notons que la fonction de projection est non linéaire. Il nous faut donc maintenant résoudre un problème d'optimisation de type moindre carré non linéaire.

Dans la suite de ce chapitre, nous allons tout d'abord formaliser l'ajustement de faisceaux (§3.2), puis montrer comment utiliser la structure creuse pour résoudre le problème d'optimisation de façon efficace (§3.3). Dans un second temps, nous allons modéliser les erreurs sur les données d'entrée que sont les points 2D (§3.4). Et enfin, nous montrerons comment est obtenue l'incertitude sur les poses et les points 3D à partir de la modélisation effectuée précédemment (§3.5).

3.2 Formulation

Nous allons maintenant formaliser l'ajustement de faisceaux global (les fonctions et variables comportent un g pour les différentier des ajustements de fais-

ceux définis par la suite). La figure 3.3 présente graphiquement les variables utilisées.

Pour un ensemble de points 3D \mathbf{s} et de caméras \mathbf{c} , $\mathbf{m}_{i,j}$ est le point 2D observation du point 3D \mathbf{s}_j dans la caméra \mathbf{c}_i . La projection de \mathbf{s}_j dans \mathbf{c}_i est $P(\mathbf{c}_i, \mathbf{s}_j)$. On regroupera l'ensemble des observations $\mathbf{m}_{i,j}$ dans \mathbf{m} . \mathfrak{M}^g est l'ensemble des couples (i, j) tel que $\mathbf{m}_{i,j}$ existe.

On va donc minimiser la fonction de coût :

$$(\mathbf{c}, \mathbf{s}) \mapsto \sum_{(i,j) \in \mathfrak{M}^g} \|\mathbf{m}_{i,j} - P(\mathbf{c}_i, \mathbf{s}_j)\|_2^2 = \|\mathbf{m} - F^g(\mathbf{c}, \mathbf{s})\|_2^2 \quad (3.2)$$

De manière à obtenir une notation plus compacte, on définit $\mathbf{y} = \mathbf{m}$, $\mathbf{x} = [\mathbf{c} \ \mathbf{s}]$. Avec ces notations, la fonction ajustement faisceaux global f^g peut se réécrire de la manière suivante :

$$f^g(\mathbf{y}) = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{y} - F^g(\tilde{\mathbf{x}})\|_2^2 \quad (3.3)$$

En pratique, dans les applications réelles, le nombre de caméras et de points 3D peut devenir conséquent. Effectuer cette minimisation avec ces tailles importantes reste la première difficulté de l'ajustement de faisceaux global. Dans le cadre de l'application visée, les ordres de grandeur pour les dimensions des vecteurs sont les suivantes. Pour une trajectoire de 400 m soit une séquence vidéo de 2700 images, le système reconstruit 384 poses de la caméra, 16365 points 3D, à partir de 74236 points 2D. Nous obtenons donc $|\mathbf{y}| = 74236 \times 2 = 148472$, $|\mathbf{x}| = 6 \times 384 + 3 \times 16365 = 51399$.

3.3 Résolution du système creux

La minimisation du problème non linéaire formulé précédemment sera effectuée grâce à l'algorithme de Levenberg-Marquard (§1.2.2). Dans cette section, nous détaillons le calcul de la jacobienne et la méthode de résolution du système. Pour obtenir plus de détail sur un point particulier, le lecteur pourra se référer à [102].

La minimisation du problème par l'application de l'algorithme de Levenberg-Marquard conduit à résoudre le système suivant :

$$\left(\mathbf{J}_{\mathbf{x}}^{F^g T} \mathbf{J}_{\mathbf{x}}^{F^g} + \lambda \mathbf{I}_{|\mathbf{x}|} \right) \delta_{\mathbf{x}} = \mathbf{J}_{\mathbf{x}}^{F^g T} \mathbf{e} \quad (3.4)$$

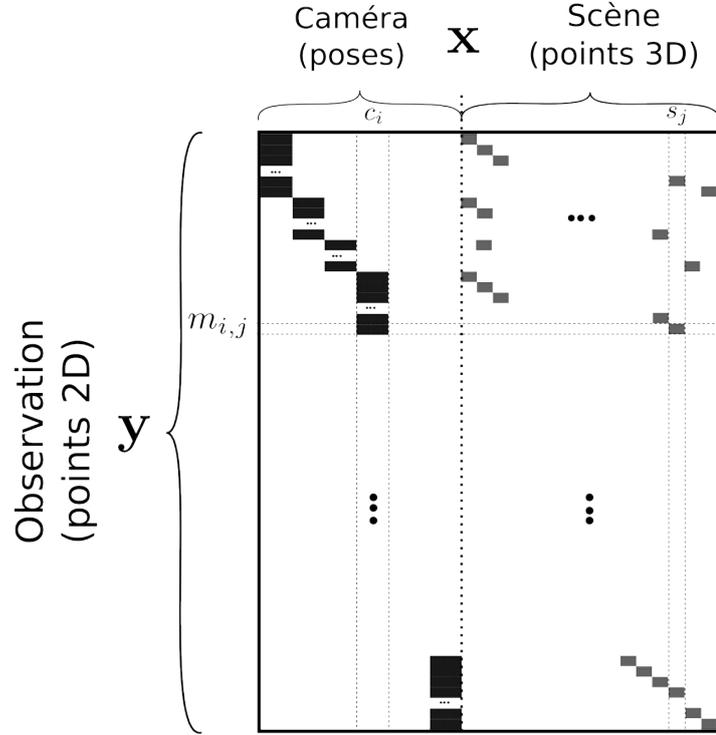


FIGURE 3.4: Structure de la matrice Jacobienne

avec $J_x^{F^g}$ la matrice jacobienne de F^g par rapport à \mathbf{x} , $J_x^{F^g T} J_x^{F^g}$ une matrice de taille $|\mathbf{x}| \times |\mathbf{x}|$, δ_x le vecteur d'incrément des estimées contenant $6|c|+3|s|$ valeurs, $\mathbf{e} = \mathbf{y} - F^g(\mathbf{x})$ le vecteur des erreurs de reprojection.

Dans le cadre de l'ajustement de faisceaux, le terme $\lambda I_{|\mathbf{x}|}$ est souvent remplacé par $\lambda \text{diag}(J_x^{F^g T} J_x^{F^g})$. Avec $\text{diag}(\mathbf{X})$ tel que $\begin{cases} \forall i \neq j, & \text{diag}(\mathbf{X})_{i,j} = 0 \\ \forall i, & \text{diag}(\mathbf{X})_{i,i} = X_{i,i} \end{cases}$

Nous allons utiliser le fait que la Jacobienne de la fonction de coût est creuse pour nous permettre de réduire le temps de calcul des matrices et de résolution du système, ainsi que l'occupation mémoire. En effet, une observation correspond à deux lignes de la matrice jacobienne, soit à la reprojection d'un point 3D dans une caméra. La jacobienne a donc une structure par blocs (2×6 et 2×3) où pour chaque observation (2 lignes), un bloc 2×6 pour les paramètres de caméra et un autre bloc 2×3 pour les 3 paramètres de point 3D sont non nuls (voir Fig. 3.4).

De par cette propriété de $J_x^{F^g}$, l'approximation de la Hessienne $\mathbf{H} = \left(J_x^{F^g T} J_x^{F^g} \right)$ a aussi une structure creuse (voir Fig. 3.5). La partie caméra \mathbf{U} est une matrice di-

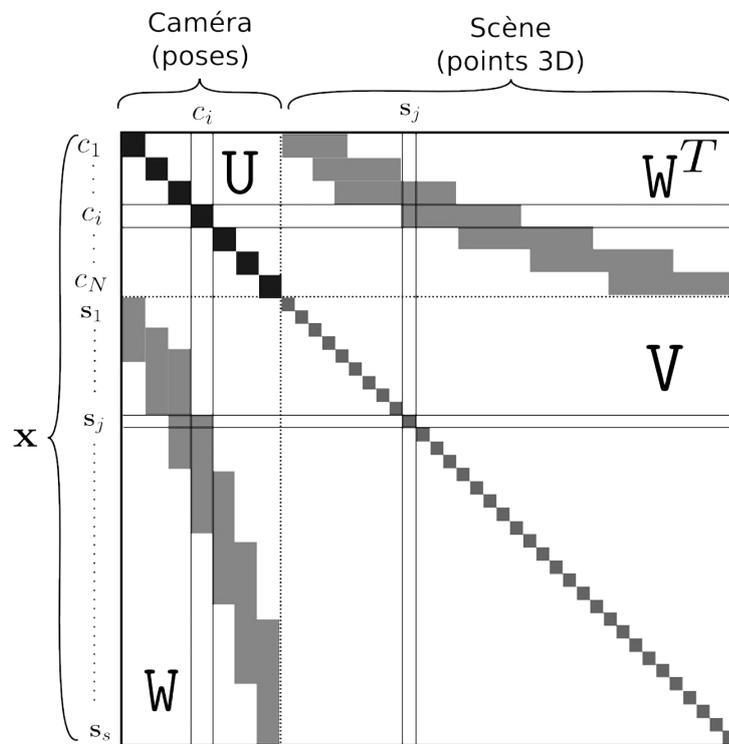


FIGURE 3.5: Structure de l'approximation de la Hessienne

agonale par blocs 6×6 et la partie point 3D V est une matrice diagonale par blocs 3×3 . La partie restante W est constituée de blocs 3×6 .

En utilisant ce fait, pour limiter l'espace mémoire utilisé, il n'est plus nécessaire de stocker J_x^{Fg} qui est une matrice de grande taille. Bien qu'il soit toujours nécessaire de calculer tous les éléments non nuls de J_x^{Fg} , il est possible de stocker uniquement $J_x^{FgT} J_x^{Fg}$ et $J_x^{FgT} e$ (soit $|x| \times |x| + |m|$ plutôt que $|x| \times |m|$ valeurs).

De plus, sans perdre en généralité, nous supposons dans la suite que le nombre de points 3D est grand devant le nombre de caméras (dans le cas contraire, on inverse le rôle des points 3D et des caméras dans la résolution du système).

En tenant compte de la structure des matrices, on réécrit le système sous cette forme en séparant la partie caméra de la partie structure :

$$\begin{pmatrix} U & W^T \\ W & V \end{pmatrix} \begin{pmatrix} \delta_c \\ \delta_s \end{pmatrix} = \begin{pmatrix} b_c \\ b_s \end{pmatrix} \quad (3.5)$$

avec U le bloc caméra, V le bloc point 3D et W le bloc contenant les interactions entre points 3D et caméras de H augmentée (la pondération ajoutée par le terme $\lambda \text{diag}(H)$ ne change pas la structure du système) ; δ_c la partie caméra et δ_s la partie point 3D de δ_x , b_c la partie caméra et b_s la partie point 3D de $J_x^{FgT} e$.

Pour pouvoir utiliser efficacement la structure des matrices, le système va être résolu en deux temps comme dans [102]. Pour cela, la partie caméra est découplée de la partie structure en prémultipliant le système précédent par $\begin{pmatrix} I & -W^T V^{-1} \\ 0 & I \end{pmatrix}$.

$$\begin{pmatrix} U - W^T V^{-1} W & 0 \\ W & V \end{pmatrix} \begin{pmatrix} \delta_c \\ \delta_s \end{pmatrix} = \begin{pmatrix} b_c - W^T V^{-1} b_s \\ b_s \end{pmatrix} \quad (3.6)$$

Comme V est diagonale par blocs, V^{-1} est la matrice diagonale des inverses des blocs, qui se calculent rapidement.

Dans un premier temps, nous nous intéressons à la résolution de la première partie du système qui correspond aux caméras de la reconstruction :

$$(U - W^T V^{-1} W) \delta_c = b_c - W^T V^{-1} b_s \quad (3.7)$$

Notons que ce système à une taille réduite ($|c|$) seulement proportionnelle au nombre de caméras. La matrice étant symétrique définie positive, la décomposition de Cholesky est utilisée pour transformer le système en systèmes triangulaires. Enfin, les deux systèmes triangulaires obtenus sont résolus. Dans un second temps, une fois la partie caméra (δ_c) déterminée, il est possible de résoudre la seconde partie du système :

$$\delta_s = V^{-1} (b_s - W \delta_c) \quad (3.8)$$

Le coût de la résolution de ce deuxième système est très réduit puisque V^{-1} a déjà été calculée.

3.4 Modèle de bruit

Dans cette section, nous décrivons la modélisation du bruit sur les primitives 2D. Ce point est important, car il nous permettra dans la section suivante d'estimer l'erreur effectuée sur la reconstruction à partir de la modélisation posée dans cette section.

Les primitives 2D extraites à partir des images sont les données inhérentes à l'estimation de la structure et des poses de caméra. L'erreur effectuée sur le calcul de la position des caméras et des points 3D provient donc intégralement de l'erreur effectuée à la détection des primitives. Cette erreur est la superposition de divers phénomènes : d'abord des erreurs dues à la formation de l'image elle-même (bruit du capteur, d'échantillonnage, de quantification), à l'extraction des primitives (manque de répétabilité, erreur due au changement de point de vue, manque de résolution de l'image) enfin à la modélisation du capteur (modèle sténopé, paramètres intrinsèques et correction de la distorsion). Dans cette thèse, nous ne considérons pas les bruits venant de la modélisation du capteur.

Le modèle choisi pour caractériser l'erreur sur les points 2D est un bruit blanc Gaussien. Le bruit peut être estimé de plusieurs façons :

1. fixé comme un *a priori* ;
2. évalué à partir du détecteur de points d'intérêts ;
3. évalué à partir de la reconstruction 3D.

Le cas 1. est l'hypothèse la plus simple : la valeur du bruit est fixée grâce à un *a priori* connu. Cette constante peut être le résultat d'un processus d'apprentissage ou inférée à partir d'une étude expérimentale préalable.

Dans le cas 2., le processus de détection des points dans l'image est modélisé. Cette approche est développée plus en détail dans [13] et [57].

Enfin, dans le cas 3., il est possible aussi d'estimer la précision de la localisation des points 2D grâce au résidu de l'optimisation par ajustement de faisceaux.

Ici, nous avons choisi d'estimer le bruit à partir de l'erreur de reprojection résiduelle (cas 3). Nous allons maintenant décrire l'estimation des paramètres de ce bruit. L'hypothèse de bruit blanc gaussien choisi précédemment permet d'écrire que :

$$\mathbf{m} \sim \mathcal{N}(\bar{\mathbf{m}}, \bar{\sigma}^2 \mathbf{I}_{|\mathbf{m}|}) \quad (3.9)$$

La fonction de densité de probabilité associée D est :

$$D : \mathbf{m} \mapsto \frac{1}{(2\pi\bar{\sigma}^2)^{\frac{|\mathbf{m}|}{2}}} e^{-\frac{1}{2\bar{\sigma}^2} \|\mathbf{m} - F^g(\bar{\mathbf{c}}, \bar{\mathbf{s}})\|_2^2} \quad (3.10)$$

Nous cherchons à estimer le paramètre σ^2 de cette distribution conjointement avec les paramètres géométriques \mathbf{c}, \mathbf{s} . Nous allons les estimer au maximum de vraisemblance (cf §1.2.1).

L'estimateur $(\hat{\sigma}^2, \hat{\mathbf{c}}, \hat{\mathbf{s}})$ au maximum de vraisemblance de $(\bar{\sigma}^2, \bar{\mathbf{c}}, \bar{\mathbf{s}})$ minimise $-\ln(D)$ et doit donc annuler les dérivées de $-\ln(D)$.

Notons que $-\ln(D)$ est minimum implique que $(\hat{\mathbf{c}}, \hat{\mathbf{s}})$ minimise $(\mathbf{c}, \mathbf{s}) \mapsto \|\mathbf{m} - F^g(\mathbf{c}, \mathbf{s})\|_2^2$, donc $(\hat{\mathbf{c}}, \hat{\mathbf{s}})$ est le résultat de l'ajustement de faisceaux global.

Maintenant, nous recherchons $\hat{\sigma}^2$ annulant la dérivée de la fonction $\sigma^2 \mapsto -\ln(D)$. Nous obtenons :

$$\hat{\sigma}^2 = \frac{\sum_{(i,j) \in \mathfrak{M}} \|\mathbf{m}_{i,j} - F^g(\hat{\mathbf{c}}, \hat{\mathbf{s}})\|_2^2}{|\mathbf{m}|} \quad (3.11)$$

Ici on utilisera plutôt l'estimateur non biaisé de σ comme proposé dans [70] :

$$\hat{\sigma}^2 = \frac{\sum_{(i,j) \in \mathfrak{M}} \|\mathbf{m}_{i,j} - F^g(\hat{\mathbf{c}}, \hat{\mathbf{s}})\|_2^2}{|\mathbf{m}| - (|\mathbf{c}| + |\mathbf{s}| - 7)} \quad (3.12)$$

Le dénominateur est corrigé de manière à prendre en compte le nombre réel de données utilisées. On enlève donc au nombre de données utilisées $|\mathbf{m}|$ le nombre de paramètres indépendants pour paramétrer la géométrie, soit 6 par pose de caméra plus 3 par points 3D, auxquels on enlève les 7 paramètres (3 pour la translation, 3 pour la rotation, et 1 pour l'échelle) permettant de fixer la jauge (cf.§1.3.4).

3.5 Propagation d'erreur

Nous allons maintenant propager le bruit tel que défini en section 3.4 dans l'ajustement de faisceaux global (défini en section 3.2).

Le but de cette étape est d'estimer l'impact de l'erreur des variables d'entrée sur les variables de sortie du système. Pour l'AFG, les variables d'entrée sont les points 2D détectés dans les images et les variables de sortie sont les poses de caméras et la position des points 3D (voir fig.3.6).

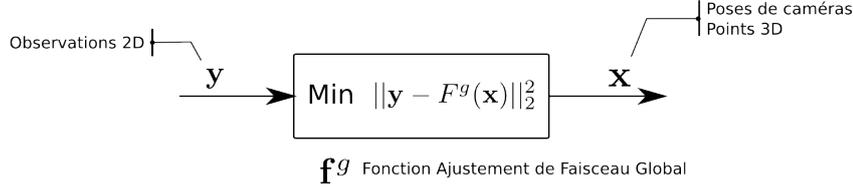


FIGURE 3.6: Entrée/sortie du système ajustement de faisceaux global

Nous avons vu précédemment que l'ajustement de faisceaux s'écrit comme l'application de la fonction f^g sur l'ensemble des points 2D. Un développement de Taylor à l'ordre 1 est effectué pour linéariser la fonction autour du point $\bar{\mathbf{y}}$. Cette approximation de la fonction f^g est :

$$\mathbf{y} \mapsto f^g(\bar{\mathbf{y}}) + \mathbf{J}_y^{f^g} (\mathbf{y} - \bar{\mathbf{y}}) + O(\|\mathbf{y} - \bar{\mathbf{y}}\|_2^2). \quad (3.13)$$

La jacobienne de la fonction implicite f^g est calculée comme défini dans §1.3.3 et est égale à :

$$\mathbf{J}_y^{f^g} = \left(\mathbf{J}_x^{F^g T} \mathbf{J}_x^{F^g} \right)^{-1} \mathbf{J}_x^{F^g T}. \quad (3.14)$$

On note ici que la jacobienne devrait être calculée en $\bar{\mathbf{x}}$. Cette valeur étant inconnue, la jacobienne $\mathbf{J}_y^{f^g}$ sera calculée en \mathbf{x} .

Pour pouvoir inverser la matrice $\mathbf{J}_x^{F^g T} \mathbf{J}_x^{F^g}$, la jauge a été fixée comme définie §1.3.4.

Maintenant, l'incertitude des points 2D est propagée pour obtenir l'incertitude sur la reconstruction :

$$\mathbf{C}_x = \mathbf{C}_{f^g(\mathbf{y})} \approx \mathbf{C}_{f^g(\bar{\mathbf{y}}) + \mathbf{J}_y^{f^g} (\mathbf{y} - \bar{\mathbf{y}})} \quad (3.15)$$

$$\approx \mathbf{C}_{\mathbf{J}_y^{f^g} (\mathbf{y} - \bar{\mathbf{y}})} \quad (3.16)$$

$$\approx \mathbf{J}_y^{f^g} \mathbf{C}_y \mathbf{J}_y^{f^g T}. \quad (3.17)$$

Si le bruit sur les points 2D est modélisé par une Gaussienne de moyenne $\bar{\mathbf{m}}$ et de covariance $\sigma^2 \mathbf{I}_{|\mathbf{m}|}$ identique pour tous les points ($\mathbf{m} \sim \mathcal{N}(\bar{\mathbf{m}}, \sigma^2 \mathbf{I}_{|\mathbf{m}|})$), alors $\mathbf{C}_y = \sigma^2 \mathbf{I}_{|\mathbf{m}|}$ et le calcul de la covariance se simplifie en :

$$\mathbf{C}_x = \sigma^2 \left(\mathbf{J}_x^{F^g T} \mathbf{J}_x^{F^g} \right)^{-1}. \quad (3.18)$$

Ces hypothèses permettent un calcul plus rapide de la covariance associée à la reconstruction. Pour accélérer les calculs, comme pour la minimisation, la jacobienne n'est pas conservée, et seuls les éléments non nuls de $\mathbf{J}_x^{F^g T} \mathbf{J}_x^{F^g}$ sont évalués. Contrairement à $\mathbf{J}_x^{F^g T} \mathbf{J}_x^{F^g}$, son inverse $\left(\mathbf{J}_x^{F^g T} \mathbf{J}_x^{F^g} \right)^{-1}$ est une matrice pleine.

Cependant, il est possible de tenir compte de la structure de $J_{\mathbf{x}}^{FgT} J_{\mathbf{x}}^{Fg}$ pour diminuer la complexité de l'inversion. Nous avons vu précédemment (section 3.3) que grâce à sa structure particulière, la matrice approximation de la matrice Hessienne peut se découper de la manière suivante :

$$\begin{pmatrix} J_{\mathbf{x}}^{FgT} & J_{\mathbf{x}}^{Fg} \end{pmatrix} = \begin{pmatrix} \mathbf{U} & \mathbf{W}^T \\ \mathbf{W} & \mathbf{V} \end{pmatrix} \quad (3.19)$$

avec \mathbf{U} et \mathbf{V} des matrices diagonales par blocs respectivement 6×6 et 3×3 . Cette matrice peut maintenant s'inverser par blocs (voir §1.2.5) :

$$\begin{pmatrix} J_{\mathbf{x}}^{FgT} & J_{\mathbf{x}}^{Fg} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{Z} & \mathbf{Y}^T \\ \mathbf{Y} & \mathbf{X} \end{pmatrix} \quad (3.20)$$

avec

$$\mathbf{Z} = (\mathbf{U} - \mathbf{W}^T \mathbf{V}^{-1} \mathbf{W})^{-1}, \quad (3.21)$$

$$\mathbf{Y} = -\mathbf{V}^{-1} \mathbf{W} \mathbf{Z} \quad (3.22)$$

$$\text{et } \mathbf{X} = \mathbf{V}^{-1} + \mathbf{V}^{-1} \mathbf{W} \mathbf{Z} \mathbf{W}^T \mathbf{V}^{-1}. \quad (3.23)$$

Ainsi, on réduit la complexité de l'inversion de $\begin{pmatrix} J_{\mathbf{x}}^{FgT} & J_{\mathbf{x}}^{Fg} \end{pmatrix}$ qui était au départ de $O((|\mathbf{c}| + |\mathbf{s}|)^3)$ si on ne tient pas compte de la structure creuse de la matrice en $O(|\mathbf{c}||\mathbf{s}|^2 + |\mathbf{c}|^3)$.

Notons que grâce à cette technique, il est possible de limiter l'estimation de la covariance à la partie caméra uniquement ($\sigma^2 \mathbf{Z}$). Ceci permet de réduire fortement le temps d'estimation si la covariance de la structure 3D ne nous intéresse pas.

La présentation de l'ajustement de faisceaux global et de la propagation d'erreur associée qui vient d'être détaillée va nous permettre maintenant de décrire les ajustements de faisceaux locaux ainsi que la propagation d'erreur associée.

3.6 Résumé de l'ajustement de faisceau global

Les entrées/sorties :

Ensemble des observations utilisées : \mathfrak{M}

Les données $\mathbf{y} = \mathbf{m}$: l'ensemble des points 2D.

Les estimés $\mathbf{x} = [\mathbf{cs}]$: l'ensemble de toutes les caméras et tous les points 3D.

Ajustement de Faisceau Global :

Fonction de projection pour l'ajustement de faisceaux global :

$$F^g(\mathbf{c}, \mathbf{s}) = [\dots P(\mathbf{c}_i, \mathbf{s}_j) \dots], \quad \forall (i, j) \in \mathfrak{M},$$

Fonction ajustement de faisceaux global :

$$f^g(\mathbf{y}) = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{y} - F^g(\tilde{\mathbf{x}})\|_2^2$$

Covariance de l'ajustement de faisceaux global :

Si les points 2D suivent un bruit blanc gaussien de variance σ^2 , $\mathbf{y} = \mathbf{m} \sim \mathcal{N}(\bar{\mathbf{m}}, \sigma^2 \mathbf{I}_{|\mathbf{m}|})$:

$$\begin{aligned} C_{\mathbf{x}} &= \sigma^2 \left(\mathbf{J}_{\mathbf{x}}^{F^g T} \mathbf{J}_{\mathbf{x}}^{F^g} \right)^{-1} \\ &= \sigma^2 \mathbf{H}^{-1} \end{aligned}$$

Chapitre 4

Ajustement de faisceaux local

Sommaire

4.1	Introduction	52
4.2	Formulation	53
4.3	Propagation d'erreur	55
4.3.1	Différence par rapport à l'ajustement de faisceaux global	55
4.3.2	Hypothèses et modèles de bruit	55
4.3.3	Covariance résultante	56
4.3.4	Cas avec corrélation	57
4.3.5	Cas avec indépendance	59
4.4	Résumé de l'ajustement de faisceaux local	64

Ce chapitre présente, dans un premier temps, l'expression de l'Ajustement de Faisceaux Local tel qu'il fût introduit pendant la thèse d'Etienne Mouragnon [79]. Dans un second temps, nous développons nos travaux concernant la propagation d'erreur associée à cet ajustement de faisceaux. Ces travaux ont fait l'objet des publications suivantes ([33],[35],[34]).

4.1 Introduction

Les méthodes d'ajustement de faisceaux global (AFG) sont particulièrement adaptées dans le cadre de la reconstruction 3D hors ligne¹, par exemple pour la reconstruction d'objets, de bâtiments, de villes. Dans ce cas, le temps de traitement n'est pas très contraignant. De plus, l'absence de contrainte embarquée permet toujours, si nécessaire, d'augmenter les ressources matérielles disponibles pour diminuer les temps de traitement.

Les travaux présentés ici sont appliqués à la localisation et la reconstruction simultanée. Il s'agit de localiser en temps réel une caméra mobile tout en reconstruisant son environnement. La contrainte temps est forte. En effet, on doit être capable de fournir la position de la caméra avant que celle-ci ait changée de manière significative. C'est cette contrainte qui définit notre notion de temps réel. En environnement urbain, le véhicule est limité à une vitesse de 50 km/h, soit 13 m/s. Afin de garantir une précision suffisante, il est nécessaire que le temps de traitement reste inférieur à 100 ms entre le moment où l'image est acquise et le moment où le processus de localisation fournit sa donnée.

Dans ce cadre, l'AFG ne permet pas, à tout instant, de répondre à cette problématique. En effet, la complexité de l'AFG s'exprime en fonction du nombre de points 3D et du nombre de caméras. Ces deux nombres augmentant avec le temps de façon quasi linéaire, le temps de calcul va croître de façon cubique et rapidement atteindre les limites acceptables en termes de temps de calcul.

En 2006, Mouragnon *et al.* [80] et Engels *et al.* [32] présentent une nouvelle approche du problème d'estimation de la position en utilisant l'ajustement de faisceaux. Celle-ci permet de réduire les temps de calcul garantissant le temps réel tout au long de la séquence et en conservant les performances de l'ajustement de faisceaux global. Cette méthode part du principe que toutes les caméras n'ont pas besoin d'être raffinées à chaque itération : les anciennes caméras du début de la trajectoire profitent peu des nouvelles données et restent proches de leur position précédente. Ainsi, dans les ajustements de faisceaux locaux, seulement une partie des caméras et du nuage de points 3D, concernés par les observations courantes, vont être optimisés.

Mouragnon *et al.* proposent d'utiliser une fenêtre glissante de N caméras et de se limiter à ce nombre de caméras pour effectuer l'optimisation. Comme présenté dans la figure 4.1, les N caméras sont séparées en deux groupes. Le premier est le groupe des caméras optimisées et contient les n dernières caméras.

1. en temps différé par rapport à l'acquisition

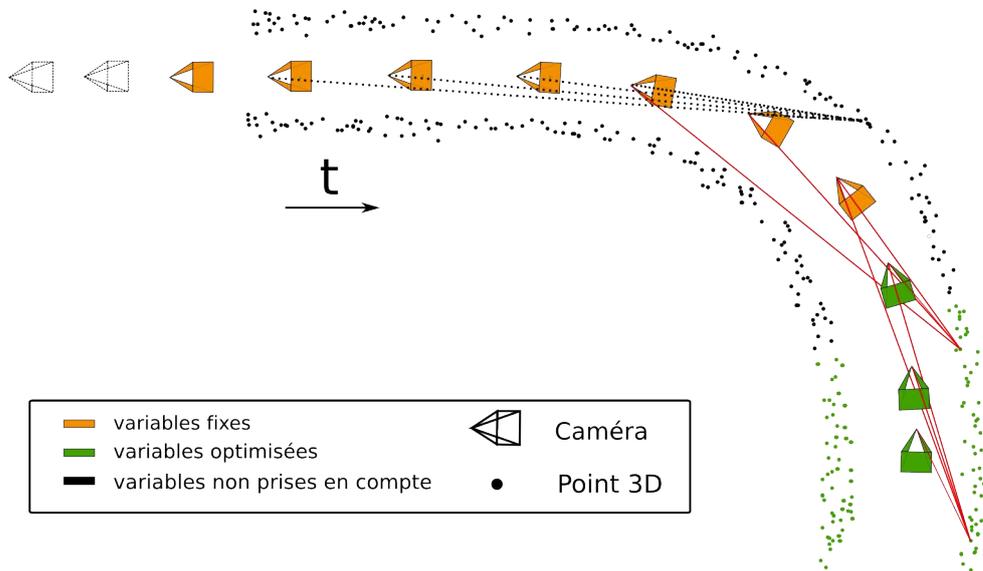


FIGURE 4.1: Exemple d'optimisation par AFL, présentant graphiquement les variables utilisées

Le second est le groupe des caméras qui servent de contraintes et contient $N - n$ caméras. Ce dernier n'est pas optimisé, mais les observations détectées dans ces images sont considérées pour l'optimisation des points 3D associés. Seuls les points 3D qui ont au moins une observation détectée dans les caméras optimisées sont conservés dans l'AFL. Cette méthode permet une réduction de la complexité, et donne la possibilité d'effectuer un ajustement incrémental efficace sur de longues séquences. Malheureusement, aucune incertitude associée à la localisation n'est fournie. Cette lacune empêche la fusion optimale des données de localisation provenant de l'algorithme de SLAM visuel avec d'autres données extérieures. Dans ce chapitre, nous présenterons notre méthode pour pallier ce problème.

Nous allons maintenant formaliser cet ajustement de faisceaux local.

4.2 Formulation

Dans ce paragraphe, nous écrivons la fonction de coût de l'ajustement de faisceaux local à l'instant t en utilisant des notations similaires à celles de l'ajustement de faisceaux global (nous utiliserons l'exposant l pour les différencier de celle de l'ajustement de faisceaux global).

L'ensemble des caméras de la fenêtre glissante est partitionné en deux vecteurs : $\mathbf{c}_o = [\mathbf{c}_{t-n+1}, \dots, \mathbf{c}_t]$ le vecteur des caméras optimisées correspondant au temps $[t - n + 1 \dots t]$ et $\mathbf{p} = [\mathbf{c}_{t-N+1}, \dots, \mathbf{c}_{t-n}]$ est le vecteur des paramètres de l'ajustement de faisceaux local (les caméras fixes qui ne seront pas optimisées) correspondant au temps $[t - N + 1, \dots, t - n]$.

L'ensemble des observations conservées par la fenêtre glissante sont les observations dans les caméras de la fenêtre glissante des points 3D optimisés :

$$\mathfrak{M}^l = \{\forall(i, j) \in \mathfrak{M} / t - N < i \leq t, \exists i_0, (i_0, j) \in \mathfrak{M} \& t - n < i_0 \leq t\} \quad (4.1)$$

On notera \mathbf{y} le vecteur des observations qui contiendra ces points 2D.

$$\mathbf{y} = [\dots \mathbf{m}_{i,j} \dots], \quad \forall(i, j) \in \mathfrak{M}^l \quad (4.2)$$

Nous allons donc chercher à optimiser l'ensemble des caméras \mathbf{c}_o et les points 3D observés par elles. Ces paramètres seront regroupés dans le vecteur des paramètres estimés \mathbf{x} :

$$\mathbf{x} = [\mathbf{c}_o | \dots \mathbf{s}_k \dots], \quad k \in \{j / \exists i, (i, j) \in \mathfrak{M}^l\} \quad (4.3)$$

La fonction de projection F^l est :

$$F^l : [\mathbf{p}, \mathbf{x}] \mapsto [\dots P(\mathbf{c}_i, \mathbf{s}_j, \dots)], \quad \forall(i, j) \in \mathfrak{M}^l \quad (4.4)$$

La fonction f^l va pouvoir s'écrire sous une forme succincte :

$$f^l(\mathbf{p}, \mathbf{y}) = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{y} - F^l(\mathbf{p}, \tilde{\mathbf{x}})\|_2^2 \quad (4.5)$$

Sous forme développée, $f^l(\mathbf{p}, \mathbf{y})$ minimise :

$$\begin{aligned} \tilde{\mathbf{x}} \mapsto & \sum_{\substack{(i,j) \in \mathfrak{M}^l \\ t-N < i \leq t-n}} \|\mathbf{m}_{i,j} - P(\mathbf{c}_i, \tilde{\mathbf{s}}_j)\|_2^2 \\ & + \sum_{\substack{(i,j) \in \mathfrak{M}^l \\ t-n < i \leq t}} \|\mathbf{m}_{i,j} - P(\tilde{\mathbf{c}}_i, \tilde{\mathbf{s}}_j)\|_2^2 \end{aligned} \quad (4.6)$$

Dans la pratique, $n = 3$ et $N = 10$ sont souvent choisis. A titre d'exemple, sur les séquences réelles d'un véhicule se déplaçant en ville, il y a en moyenne 213 points détectés dans chaque image dont environ 200 ont une correspondance 3D, ce qui se traduit par des vecteurs de dimensions suivantes : $|\mathbf{y}| = 10 \times 200 \times 2 = 4000$, $|\mathbf{p}| = 7 \times 6 = 42$, $|\mathbf{x}| = 3 \times 6 + 200 \times 3 = 618$.

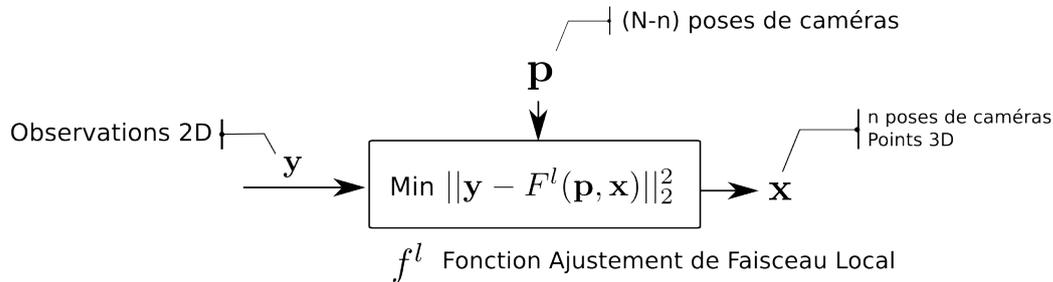


FIGURE 4.2: Entrées/sorties de l'Ajustement de faisceaux local

4.3 Propagation d'erreur

4.3.1 Différence par rapport à l'ajustement de faisceaux global

Contrairement à l'ajustement faisceaux global, l'ajustement de faisceaux local est un processus itératif. On va réestimer à chaque itération la valeur des dernières caméras (c_o) à partir d'un ensemble de caméras considérées comme fixes (\mathbf{p}). Maintenant, le vecteur d'entrée ne contient plus non seulement l'ensemble des points 2D vus dans les images, mais aussi cet ensemble de caméras fixes (cf Fig.4.2). Nous avons deux possibilités :

1. nous déclarons les caméras fixes comme parfaitement connues et dans ce cas leur incertitude sera nulle. Nous utiliserons alors la méthode standard d'inversion de la matrice hessienne et cette mesure représentera l'incertitude locale des nouvelles caméras par rapport aux anciennes.
2. Nous tenons compte de l'incertitude des caméras fixes qui par ailleurs devra être estimée au préalable. Et ainsi, nous propageons l'erreur itérativement depuis le début de la trajectoire. L'objectif est d'obtenir une covariance similaire à celle que nous fournit la propagation d'incertitude dans un AFG appliqué à toute la trajectoire déjà reconstruite.

Nous allons maintenant décrire le modèle de bruit associé aux entrées et montrer comment nous tenons compte de l'incertitude des caméras fixes dans la propagation.

4.3.2 Hypothèses et modèles de bruit

Le système possède donc deux types de variables d'entrée dont il est nécessaire de modéliser le bruit : les points 2D vus dans les différentes caméras et les

poses des caméras fixes.

Pour les points 2D, les hypothèses d'indépendance et de même variance sont conservées :

$$\mathbf{m} \sim \mathcal{N}(\bar{\mathbf{m}}, \sigma^2 \mathbf{I}_{|\mathbf{m}|}). \quad (4.7)$$

L'incertitude des caméras fixes est modélisée par une gaussienne. L'indépendance entre les différentes poses ne peut être supposée, car l'estimation de la pose d'une caméra est intrinsèquement liée à la position des caméras précédentes en raison du processus itératif. Ainsi :

$$\mathbf{p} \sim \mathcal{N}(\bar{\mathbf{p}}, C_{\mathbf{p}}). \quad (4.8)$$

De plus, il existe une relation entre les points 2D déjà utilisés pour la localisation et les caméras déjà localisées. Pour cela, deux modélisations sont utilisées pour prendre en compte les interactions entre les points 2D \mathbf{m} et les caméras \mathbf{p} .

- La première prend en compte l'ensemble des interactions entre les caméras et les points 2D (§4.3.4). Elle sera dénommée "avec corrélation" dans la suite de ce mémoire.
- La seconde suppose l'indépendance entre les points 2D et les caméras (§4.3.5) et sera dénommée "avec indépendance". Cette seconde modélisation est une assertion forte. En effet, les poses des caméras étant estimées à partir des points 2D, nous pouvons difficilement supposer que ces deux termes sont indépendants.

Nous allons maintenant présenter la propagation d'erreur pour l'ajustement de faisceaux local pour les deux modélisations que nous venons de décrire.

4.3.3 Covariance résultante

La fonction d'ajustement de faisceaux local est linéarisée pour pouvoir propager la covariance dans le système. La fonction f^l est pour cela approximée par son développement de Taylor au premier ordre au point $[\bar{\mathbf{y}}\bar{\mathbf{p}}]$:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{p} \\ f^l(\mathbf{p}, \mathbf{y}) \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{p}} \\ f^l(\bar{\mathbf{p}}, \bar{\mathbf{y}}) \end{pmatrix} + \begin{pmatrix} \mathbf{I}_{|\mathbf{y}|} & 0_{|\mathbf{y}| \times |\mathbf{p}|} \\ 0_{|\mathbf{p}| \times |\mathbf{y}|} & \mathbf{I}_{|\mathbf{p}|} \\ \mathbf{J}_{\mathbf{y}}^{f^l} & \mathbf{J}_{\mathbf{p}}^{f^l} \end{pmatrix} \begin{pmatrix} \mathbf{y} - \bar{\mathbf{y}} \\ \mathbf{p} - \bar{\mathbf{p}} \end{pmatrix}. \quad (4.9)$$

La jacobienne de la fonction implicite f^l (Eq. 4.6) est calculée comme définie dans la section §1.3.3 et est égale à :

$$\mathbf{J}_{\mathbf{y}}^{f^l} = \left(\mathbf{J}_{\mathbf{x}}^{F^l T} \mathbf{J}_{\mathbf{x}}^{F^l} \right)^{-1} \mathbf{J}_{\mathbf{x}}^{F^l T}$$

$$\mathbf{J}_{\mathbf{p}}^{f^l} = - \left(\mathbf{J}_{\mathbf{x}}^{F^{lT}} \mathbf{J}_{\mathbf{x}}^{F^l} \right)^{-1} \mathbf{J}_{\mathbf{x}}^{F^{lT}} \mathbf{J}_{\mathbf{p}}^{F^l}. \quad (4.10)$$

De la section précédente (§4.3.2), nous savons que :

$$[\mathbf{mp}] = [\mathbf{yp}] \sim \mathcal{N}([\bar{\mathbf{y}}\bar{\mathbf{p}}], C_{[\mathbf{yp}]}). \quad (4.11)$$

En propageant ce bruit en utilisant les équations Eq.4.9 et Eq.4.10, nous obtenons :

$$[\mathbf{ypx}] \sim \mathcal{N}([\bar{\mathbf{y}}\bar{\mathbf{p}}\bar{\mathbf{x}}], C_{[\mathbf{ypx}]}) \quad (4.12)$$

avec sa covariance

$$C_{[\mathbf{ypx}]} = \begin{pmatrix} \mathbf{I}_{|y|} & 0_{|y| \times |p|} \\ 0_{|p| \times |y|} & \mathbf{I}_{|p|} \\ \mathbf{J}_{\mathbf{y}}^{f^l} & \mathbf{J}_{\mathbf{p}}^{f^l} \end{pmatrix} C_{[\mathbf{yp}]} \begin{pmatrix} \mathbf{I}_{|y|} & 0 & \mathbf{J}_{\mathbf{y}}^{f^{lT}} \\ 0 & \mathbf{I}_{|p|} & \mathbf{J}_{\mathbf{p}}^{f^{lT}} \end{pmatrix}. \quad (4.13)$$

4.3.4 Cas avec corrélation

Dans cette section, la propagation d'incertitude est détaillée dans le cas où l'on conserve la corrélation entre les points 2D et les caméras précédemment estimées. La corrélation existe seulement entre les caméras déjà estimées en utilisant les points 2D déjà observés. Les nouveaux points 2D observés par les caméras optimisées c_o sont indépendants des caméras fixes \mathbf{p} . Nous notons \mathbf{y}^- l'ensemble des points 2D à l'itération précédente. Donc $\mathbf{y} \cap \mathbf{y}^-$ est l'ensemble des points déjà observés et $\mathbf{y} \setminus \mathbf{y}^-$ est l'ensemble des nouveaux points 2D.

Ainsi $\mathbf{y} \setminus \mathbf{y}^-$ et $[\mathbf{y} \cap \mathbf{y}^-, \mathbf{p}]$ sont indépendants et :

$$C_{[\mathbf{yp}]} = C_{[\mathbf{y} \setminus \mathbf{y}^-, \mathbf{y} \cap \mathbf{y}^-, \mathbf{p}]} = \begin{pmatrix} \sigma^2 \mathbf{I}_{|\mathbf{y} \setminus \mathbf{y}^-|} & 0_{|\mathbf{y} \setminus \mathbf{y}^-| \times |\mathbf{y} \cap \mathbf{y}^-, \mathbf{p}|} \\ 0_{|\mathbf{y} \cap \mathbf{y}^-, \mathbf{p}| \times |\mathbf{y} \setminus \mathbf{y}^-|} & C_{[\mathbf{y} \cap \mathbf{y}^-, \mathbf{p}]} \end{pmatrix}. \quad (4.14)$$

En développant la formulation précédente (Eq. 4.13), nous obtenons la formulation simplifiée suivante :

$$C_{[\mathbf{ypx}]} = \begin{pmatrix} C_{[\mathbf{yp}]} & C_{[\mathbf{yp}]} \mathbf{J}_{[\mathbf{yp}]}^{f^l T} \\ \mathbf{J}_{[\mathbf{yp}]}^{f^l} C_{[\mathbf{yp}]} & \mathbf{J}_{[\mathbf{yp}]}^{f^l} C_{[\mathbf{yp}]} \mathbf{J}_{[\mathbf{yp}]}^{f^l T} \end{pmatrix}. \quad (4.15)$$

Avec ces hypothèses, il est nécessaire de calculer la mise à jour de la corrélation entre les caméras dans \mathbf{p} et les points 2D. Cela nécessite d'explicitement la matrice jacobienne $\mathbf{J}_{\mathbf{y}}^{f^l}$ qui est de grande taille ($|y| \times |x|$) et dont le calcul n'est pas trivial.

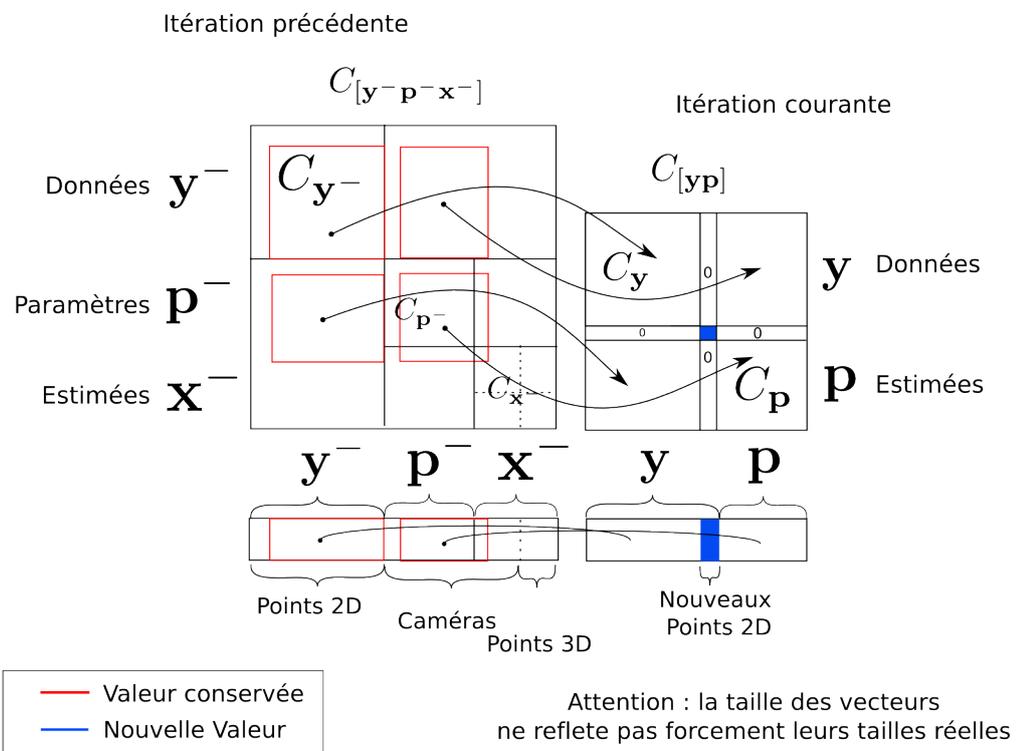


FIGURE 4.3: Récurrence pour le cas avec corrélation

A l'itération t , on estime $[\mathbf{y}\mathbf{p}\mathbf{x}]$ et $C_{[\mathbf{y}\mathbf{p}\mathbf{x}]}$ grâce à $[\mathbf{y}^- \mathbf{p}^- \mathbf{x}^-]$ et $C_{[\mathbf{y}^- \mathbf{p}^- \mathbf{x}^-]}$ comme suit :

1. \mathbf{p} est un sous-vecteur de $[\mathbf{p}^- \mathbf{x}^-]$, \mathbf{y} est composé d'un sous-vecteur de \mathbf{y}^- et de nouveaux points 2D (voir figure 4.3).
2. $C_{[\mathbf{y} \cap \mathbf{y}^- | \mathbf{p}]}$ est une sous-matrice de $C_{[\mathbf{y}^- \mathbf{p}^- \mathbf{x}^-]}$ (voir figure 4.3).
3. On estime $\mathbf{x} = f^l(\mathbf{p}, \mathbf{y})$ en utilisant l'AFL (Eq. 4.5)
4. On estime $C_{[\mathbf{y}\mathbf{p}\mathbf{x}]}$ en utilisant l'Eq. 4.15

Dans cet algorithme, on estime $C_{[\mathbf{p}\mathbf{x}]}$. Nous rappelons que l'estimation de la covariance des points 3D (C_s) n'est pas nécessaire : il est possible de se limiter à l'estimation de la partie caméra de \mathbf{x} (C_{c_o}), notamment pour diminuer les temps de calcul.

FIGURE 4.4: Algorithme de calcul de la covariance de l'ajustement de faisceaux local (cas avec corrélation).

Récurrance

Le processus d'ajustement de faisceaux local étant itératif, l'estimation de la covariance à un instant dépend de l'estimation au temps précédent. A chaque itération, la fenêtre glissante se déplace d'une caméra. La figure 4.3 montre les relations entre les différentes variables. Les variables du temps précédent sont notées avec un $-$ pour les différencier.

Ainsi, \mathbf{p} est un sous-vecteur de $[\mathbf{p}^- \mathbf{x}^-]$. \mathbf{y} est composé d'un sous-vecteur de \mathbf{y}^- et des nouveaux points 2D (en bleu dans la Fig. 4.3). La covariance $C_{[\mathbf{y}\mathbf{p}]}$ est composée d'une sous-matrice de $C_{[\mathbf{y}^- \mathbf{p}^- \mathbf{x}^-]}$ provenant de l'estimation précédente et d'un bloc diagonal pour les nouveaux points 2D.

La figure 4.4 résume l'algorithme d'AFL et la propagation d'erreur associée.

4.3.5 Cas avec indépendance

Dans cette section, on traite le cas de l'hypothèse simplificatrice d'indépendance : c'est-à-dire que les vecteurs gaussiens \mathbf{p} et \mathbf{y} sont supposés indépendants.

$$[\mathbf{y}\mathbf{p}] \sim \mathcal{N}([\bar{\mathbf{y}}\bar{\mathbf{p}}], C_{[\mathbf{y}\mathbf{p}]}) \quad C_{[\mathbf{y}\mathbf{p}]} = \begin{pmatrix} \sigma^2 \mathbf{I}_{|\mathbf{m}|} & \mathbf{0}_{|\mathbf{m}| \times |\mathbf{p}|} \\ \mathbf{0}_{|\mathbf{p}| \times |\mathbf{m}|} & C_{\mathbf{p}} \end{pmatrix}. \quad (4.16)$$

En combinant l'expression de la propagation de la covariance (Eq. 4.13) et

l'hypothèse d'indépendance (Eq. 4.16), la covariance se simplifie en :

$$C_{[px]} = \begin{pmatrix} C_p & C_p J_p^{f^l T} \\ J_p^{f^l} C_p & J_p^{f^l} C_p J_p^{f^l T} + \sigma^2 J_y^{f^l} J_y^{f^l T} \end{pmatrix} \quad (4.17)$$

Cette formulation permet de réduire fortement les calculs. En effet, il n'est plus nécessaire d'estimer les corrélations entre les points 2D précédemment vus et les caméras paramètres et nous allons voir dans le paragraphe suivant comment tirer parti de la structure creuse pour augmenter la vitesse de calcul.

Estimation rapide

Ici, nous montrons le détail du calcul creux utilisé pour accélérer l'estimation de la covariance $C_{[px]}$.

Pour calculer cette covariance, il nous faut calculer deux termes : $J_y^{f^l} J_y^{f^l T}$ et $J_p^{f^l}$

En développant ces deux termes, on se rend compte qu'il y a des éléments communs que nous allons utiliser pour augmenter l'efficacité de l'estimation. Tout d'abord de la même manière que pour l'ajustement de faisceaux global, le premier terme se simplifie :

$$J_y^{f^l} = \left(J_x^{F^l T} J_x^{F^l} \right)^{-1} J_x^{F^l T} \quad (4.18)$$

$$J_y^{f^l} J_y^{f^l T} = \left(J_x^{F^l T} J_x^{F^l} \right)^{-1} \cancel{J_x^{F^l T} J_x^{F^l}} \left(J_x^{F^l T} J_x^{F^l} \right)^{-1} \quad (4.19)$$

$$= \left(J_x^{F^l T} J_x^{F^l} \right)^{-1} \quad (4.20)$$

$$= H^{-1} \quad (4.21)$$

Cette matrice $J_x^{F^l T} J_x^{F^l}$ a la même structure que celle de l'ajustement de faisceaux global (mais l'ensemble des paramètres n'est pas le même). Nous pouvons donc effectuer l'inversion par blocs en tenant compte de la structure particulière de H (cf §3.5). De plus, nous avons :

$$J_p^{f^l} = - \left(J_x^{F^l T} J_x^{F^l} \right)^{-1} J_x^{F^l T} J_p^{F^l} \quad (4.22)$$

$$= -H^{-1} J_x^{F^l T} J_p^{F^l} \quad (4.23)$$

Pour calculer le deuxième terme, nous utilisons le calcul de H^{-1} déjà effectué. Une fois, ces précalculs terminés, nous pouvons les utiliser pour calculer $C_{[px]}$.

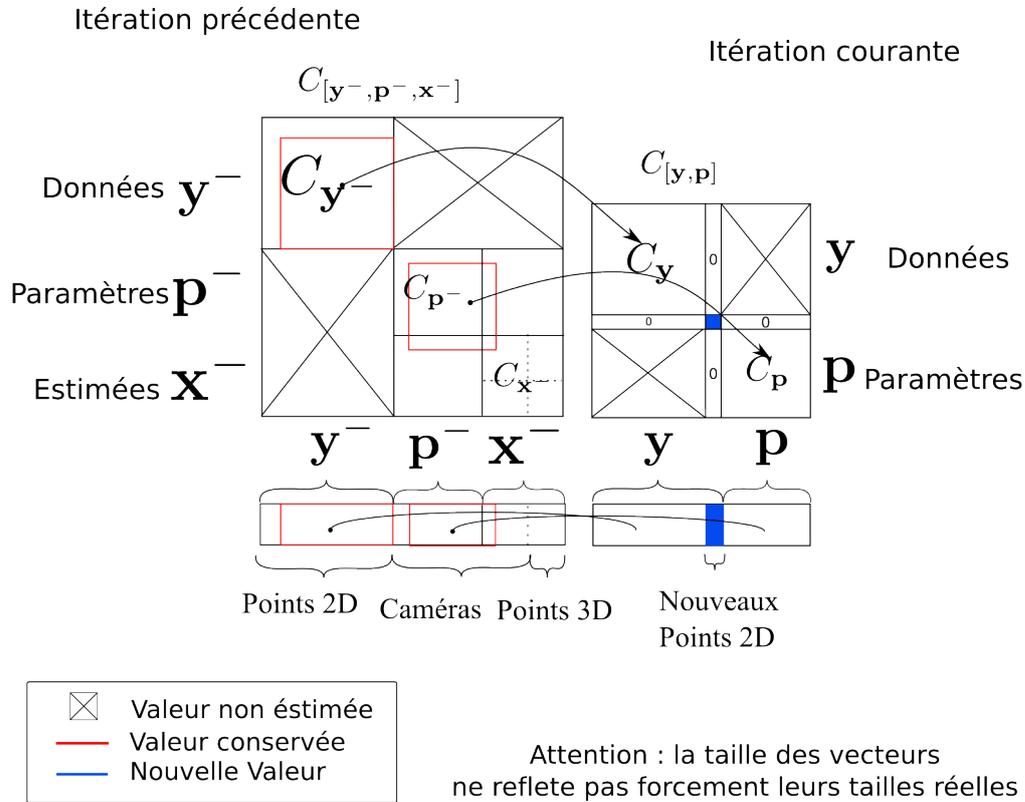


FIGURE 4.5: Récurrence pour le cas avec indépendance

$$C_{[px]} = \begin{pmatrix} C_p & -C_p \begin{pmatrix} J_p^{F^l T} & J_x^{F^l} \end{pmatrix} H^{-1} \\ -H^{-1} \begin{pmatrix} J_x^{F^l T} & J_p^{F^l} \end{pmatrix} C_p & H^{-1} \begin{pmatrix} J_x^{F^l T} & J_p^{F^l} \end{pmatrix} C_p \begin{pmatrix} J_p^{F^l T} & J_x^{F^l} \end{pmatrix} H^{-1} + \sigma^2 H^{-1} \end{pmatrix} \quad (4.24)$$

Récurrence

Le processus étant incrémental, l'estimation courante de la covariance dépend du résultat au temps précédent. Ici, on détaille les relations entre ces différentes variables illustrées dans la figure 4.5. Les variables du temps précédent seront marquées avec un $^-$ pour les différencier des états courants.

Les variables d'entrées de l'étape courante sont \mathbf{y} et \mathbf{p} . \mathbf{p} est un sous-vecteur de $[\mathbf{p}^- \mathbf{x}^-]$. \mathbf{y} est composé d'un sous-vecteur de \mathbf{y}^- et des nouveaux points ob-

On estime $[\mathbf{px}]$ et $C_{[\mathbf{px}]}$ à partir de $[\mathbf{p}^- \mathbf{x}^-]$ et $C_{[\mathbf{p}^- \mathbf{x}^-]}$ comme suit :

1. \mathbf{p} est un sous-vecteur de $[\mathbf{p}^- \mathbf{x}^-]$ (voir figure 4.5)
2. $C_{\mathbf{p}}$ est une sous-matrice de $C_{[\mathbf{p}^- \mathbf{x}^-]}$ (voir figure 4.5)
3. On estime $\mathbf{x} = f_t^l(\mathbf{p}, \mathbf{y})$ en utilisant l'AFL (Eq. 4.5)
4. On estime $C_{[\mathbf{px}]}$ en utilisant l' Eq. 4.24

Dans cet algorithme, on estime $C_{[\mathbf{px}]}$. Nous rappelons que l'estimation de la covariance des points 3D (C_s) n'est pas nécessaire : il est possible de se limiter à l'estimation de la partie caméra de \mathbf{x} (C_{c_o}), notamment pour diminuer les temps de calcul. Pour plus de détails voir §4.3.5.

FIGURE 4.6: Algorithme de calcul de la covariance de l'ajustement de faisceaux local (cas avec hypothèse d'indépendance).

servés. Grâce à l'hypothèse d'indépendance, les covariances de \mathbf{y} et \mathbf{p} ne sont plus corrélées et sont maintenant estimées de manière indépendante contrairement au cas précédent. Ainsi :

1. la matrice de covariance C_y est une matrice diagonale par blocs composée de deux blocs : une sous matrice de C_{y^-} contenant la covariance des points 2D déjà observés et un bloc contenant la covariance des nouveaux points 2D (en bleu sur la figure).
2. Et $C_{\mathbf{p}}$ est une sous matrice de $C_{[\mathbf{p}^- \mathbf{x}^-]}$ contenant la covariance des caméras contraintes.

Récurrance "minimale" Pour propager la covariance d'itération en itération, il est seulement nécessaire pour cet algorithme de calculer la covariance des poses des caméras \mathbf{p} . Il est donc inutile d'évaluer la covariance des points.

Ici, on se propose de détailler le calcul par blocs de la covariance des caméras. On rappelle le résultat obtenu précédemment (Eq. 4.24) :

$$C_{[\mathbf{px}]} = \begin{pmatrix} C_{\mathbf{p}} & -C_{\mathbf{p}} \begin{pmatrix} J_{\mathbf{p}}^{F^l T} & J_{\mathbf{x}}^{F^l} \end{pmatrix} H^{-1} \\ -H^{-1} \begin{pmatrix} J_{\mathbf{x}}^{F^l T} & J_{\mathbf{p}}^{F^l} \end{pmatrix} C_{\mathbf{p}} & H^{-1} \begin{pmatrix} J_{\mathbf{x}}^{F^l T} & J_{\mathbf{p}}^{F^l} \end{pmatrix} C_{\mathbf{p}} \begin{pmatrix} J_{\mathbf{p}}^{F^l T} & J_{\mathbf{x}}^{F^l} \end{pmatrix} H^{-1} + \sigma^2 H^{-1} \end{pmatrix}$$

Dans cette matrice, pour effectuer la récurrence, il est uniquement nécessaire d'évaluer la covariance de la partie caméra $[\mathbf{pc}_o]$.

Pour cela, nous posons :

$$\begin{pmatrix} J_x^{F^l T} & J_x^{F^l} \end{pmatrix} = H = \begin{pmatrix} U & W^T \\ W & V \end{pmatrix} \quad (4.25)$$

$$H^{-1} = \begin{pmatrix} Z & Y^T \\ Y & X \end{pmatrix} \quad (4.26)$$

$$\begin{pmatrix} J_x^{F^l T} & J_p^{F^l} \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix} \quad (4.27)$$

$\begin{pmatrix} J_x^{F^l T} & J_p^{F^l} \end{pmatrix}$ est une matrice de taille $|x| \times |p|$, on note A la partie caméra et B la partie correspondant aux points 3D.

Les deux blocs inférieurs de l'Eq. 4.24 peuvent se réécrire comme suit en utilisant les matrices par blocs :

$$\begin{aligned} H^{-1} \begin{pmatrix} J_x^{F^l T} & J_p^{F^l} \end{pmatrix} C_p \begin{pmatrix} J_p^{F^l T} & J_x^{F^l} \end{pmatrix} H^{-1} & \begin{pmatrix} Z & Y^T \\ Y & X \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} C_p (A^T \ B^T) \begin{pmatrix} Z & Y^T \\ Y & X \end{pmatrix} \\ + \sigma^2 H^{-1} & = + \sigma^2 \begin{pmatrix} Z & Y^T \\ Y & X \end{pmatrix} \end{aligned} \quad (4.28)$$

$$H^{-1} \begin{pmatrix} J_x^{F^l T} & J_p^{F^l} \end{pmatrix} C_p = \begin{pmatrix} Z & Y^T \\ Y & X \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} C_p \quad (4.29)$$

Notons que la partie caméra A est nulle du fait de la non dépendance de l'équation de projection à deux caméras différentes (chaque équation ne fait intervenir qu'une unique caméra). Donc, en développant, la partie caméra $[p_{c_o}]$ sera donnée par :

$$C_{[p_{c_o}]} = \begin{pmatrix} C_p & C_p B^T Y \\ Y^T B C_p & Y^T B C_p B^T Y + \sigma^2 Z \end{pmatrix} \quad (4.30)$$

Sous cette forme, en calculant H^{-1} par bloc puis en calculant seulement $C_{[p_{c_o}]}$, nous obtenons un algorithme de propagation de covariance qui est utilisable en temps réel.

4.4 Résumé de l'ajustement de faisceaux local

Les entrées/sortie :

Ensemble des observations utilisées :

$$\mathfrak{M}^l = \{\forall(i, j) \in \mathfrak{M}/t - N < i \leq t, \exists i_0, (i_0, j) \in \mathfrak{M} \ \& \ t - n < i_0 \leq t\}$$

Les données	$\mathbf{y} = [\dots \mathbf{m}_{i,j} \dots]$: l'ensemble des points 2D.
Les paramètres	$\mathbf{p} = [\mathbf{c}_{t-N+1}, \dots, \mathbf{c}_{t-n}]$: les caméras fixes.
Les variables estimées	$\mathbf{x} = [\mathbf{c}_o \dots \mathbf{s}_k \dots]$: les caméras optimisées et les points 3D vus dans ces caméras.

avec $\mathbf{c}_o = [\mathbf{c}_{t-n+1}, \dots, \mathbf{c}_t]$, $k \in \{j/\exists i, (i, j) \in \mathfrak{M}^l\}$

Ajustement de faisceaux local

Fonction de projection de l'ajustement de faisceaux local :

$$F^l(\mathbf{p}, \mathbf{x}) = [\dots P(\mathbf{c}_i, \mathbf{s}_j) \dots] \forall (i, j) \in \mathfrak{M}^l$$

fonction ajustement de faisceaux local :

$$\mathbf{x} = f^l(\mathbf{y}) = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{m} - F^l(\mathbf{p}, \tilde{\mathbf{x}})\|_2^2$$

Covariance avec corrélation

Le modèle de bruit :

$$C_{[\mathbf{y}\mathbf{p}]} = C_{[\mathbf{y}\setminus\mathbf{y}^-, \mathbf{y}\cap\mathbf{y}^-, \mathbf{p}]} = \begin{pmatrix} \sigma^2 \mathbf{I}_{|\mathbf{y}\setminus\mathbf{y}^-|} & 0_{|\mathbf{y}\setminus\mathbf{y}^-| \times |\mathbf{y}\cap\mathbf{y}^-, \mathbf{p}|} \\ 0_{|\mathbf{y}\cap\mathbf{y}^-, \mathbf{p}| \times |\mathbf{y}\setminus\mathbf{y}^-|} & C_{[\mathbf{y}\cap\mathbf{y}^-, \mathbf{p}]} \end{pmatrix}.$$

La propagation d'erreur s'écrit :

$$C_{[\mathbf{y}\mathbf{p}\mathbf{x}]} = \begin{pmatrix} C_{[\mathbf{y}\mathbf{p}]} & C_{[\mathbf{y}\mathbf{p}]} \mathbf{J}_{[\mathbf{y}\mathbf{p}]}^{f^l T} \\ \mathbf{J}_{[\mathbf{y}\mathbf{p}]}^{f^l} C_{[\mathbf{y}\mathbf{p}]} & \mathbf{J}_{[\mathbf{y}\mathbf{p}]}^{f^l} C_{[\mathbf{y}\mathbf{p}]} \mathbf{J}_{[\mathbf{y}\mathbf{p}]}^{f^l T} \end{pmatrix}$$

Covariance avec hypothèse d'indépendance

Le modèle de bruit :

$$C_{[\mathbf{y}\mathbf{p}]} = \begin{pmatrix} \sigma^2 \mathbf{I}_{|\mathbf{m}|} & 0_{|\mathbf{m}| \times |\mathbf{p}|} \\ 0_{|\mathbf{p}| \times |\mathbf{m}|} & C_{\mathbf{p}} \end{pmatrix}.$$

La propagation d'erreur s'écrit :

$$C_{[\mathbf{p}\mathbf{x}]} = \begin{pmatrix} C_{\mathbf{p}} & -C_{\mathbf{p}} \left(\mathbf{J}_{\mathbf{p}}^{F^l T} \mathbf{J}_{\mathbf{x}}^{F^l} \right) \mathbf{H}^{-1} \\ -\mathbf{H}^{-1} \left(\mathbf{J}_{\mathbf{x}}^{F^l T} \mathbf{J}_{\mathbf{p}}^{F^l} \right) C_{\mathbf{p}} & \mathbf{H}^{-1} \left(\mathbf{J}_{\mathbf{x}}^{F^l T} \mathbf{J}_{\mathbf{p}}^{F^l} \right) C_{\mathbf{p}} \left(\mathbf{J}_{\mathbf{p}}^{F^l T} \mathbf{J}_{\mathbf{x}}^{F^l} \right) \mathbf{H}^{-1} + \sigma^2 \mathbf{H}^{-1} \end{pmatrix}$$

Chapitre 5

Ajustement de faisceaux local pondéré

Sommaire

5.1	Introduction	66
5.2	Formulation	67
5.3	Différence par rapport à l'AFG et l'AFL	68
5.4	Propagation d'erreur	69
5.4.1	Cas avec corrélation	70
5.4.2	Cas avec hypothèse d'indépendance	73
5.5	Résumé de l'ajustement de faisceaux local pondéré	78

Ce chapitre présente un nouvel ajustement de faisceaux local, l'ajustement de faisceaux local pondéré. Cet ajustement, contrairement à celui du chapitre précédent, tient compte de l'incertitude sur les caméras précédentes. Dans un premier temps, nous développerons la formulation de cet ajustement de faisceaux, puis nous montrerons comment dériver la propagation d'incertitude dans ce cas.

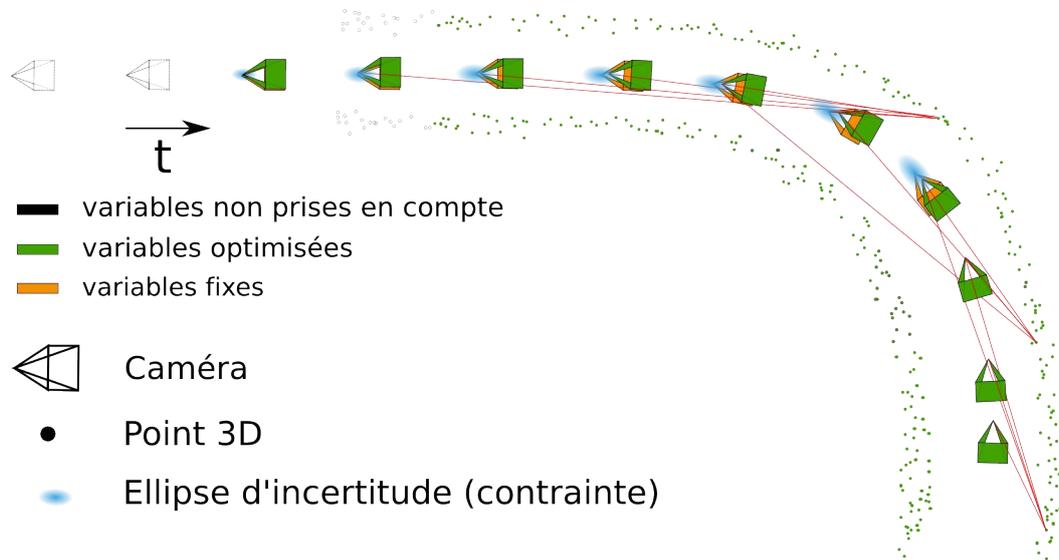


FIGURE 5.1: Exemple d'optimisation par AFLP, présentant graphiquement les variables utilisées.

5.1 Introduction

Comme l'ajustement de faisceaux local, le but est ici de raffiner itérativement la trajectoire d'une caméra mobile. L'AFL proposé par Mouragnon *et al.* [80] considère les caméras fixes comme un *a priori* connu et ne tient pas compte de l'incertitude sur la reconstruction effectuée préalablement. Nous allons proposer un nouvel ajustement de faisceaux local qui tient compte de la confiance sur les caméras précédemment estimées.

Le même schéma que précédemment est conservé, mais les caméras qui étaient jusqu'alors considérées comme parfaitement connues donc fixes seront maintenant raffinées. Les entrées de l'AFL sont maintenant les points 2D (\mathbf{m}) et l'estimation précédente des positions des caméras contraintes (\mathbf{c}_c^-) ainsi que la confiance sur cette estimation ($C_{\mathbf{c}_c^-}$). Nous cherchons à estimer les positions optimisées de toutes les caméras de la fenêtre glissante et la structure liée (\mathbf{x}). Dans ce nouvel ajustement de faisceaux local, il n'y a plus de paramètres fixes.

5.2 Formulation

Comme pour l'ajustement de faisceaux local, nous nous plaçons dans la fenêtre glissante contenant les N dernières caméras à l'instant t . Les poses des caméras et la structure doivent satisfaire deux contraintes. La première, la contrainte de l'ajustement de faisceaux, impose que les reprojections des points 3D dans les images soient le plus proche possible des points 2D observés (\mathbf{m}). La seconde, la contrainte due à l'*a priori* sur les caméras, impose que les caméras contraintes (\mathbf{c}_c) restent proches de leurs positions précédemment estimées (\mathbf{c}_c^-).

Les caméras de la fenêtre glissante se partitionnent en deux : $\mathbf{c}_c = [\mathbf{c}_{t-N+1} \dots \mathbf{c}_{t-n}]$ est le vecteur des caméras contraintes et $\mathbf{c}_o = [\mathbf{c}_{t-n+1} \dots \mathbf{c}_t]$ le vecteur des caméras optimisées (sans contrainte). $\mathbf{c}_c^- = [\mathbf{c}_{t-N+1}^{t-1} \dots \mathbf{c}_{t-n}^{t-1}]$ est le vecteur des poses des caméras contraintes à l'instant précédent.

Les observations conservées dans la fenêtre glissante sont :

$$\mathfrak{M}^{lp} = \{(i, j) \in \mathfrak{M} / t - N < i \leq t\} \quad (5.1)$$

Les données servant à l'estimation sont dans le vecteur \mathbf{y} , qui contient les points 2D observés et l'*a priori* sur la position précédente des caméras (\mathbf{c}_c^-) :

$$\mathbf{y} = [\dots \mathbf{m}_{i,j} \dots | \mathbf{c}_c^-], \quad \forall (i, j) \in \mathfrak{M}^{lp} \quad (5.2)$$

Les variables estimées \mathbf{x} sont toutes les caméras de la fenêtre glissante et les points 3D observés par elles :

$$\mathbf{x} = [\mathbf{c}_c \mathbf{c}_o | \dots \mathbf{s}_k \dots], \quad \forall k \in \{j / \exists (i, j) \in \mathfrak{M}^{lp}\} \quad (5.3)$$

Il est à noter que, en pratique, les points 3D ayant moins de 3 observations ne sont pas utilisés dans l'optimisation.

La fonction de projection de ce nouvel ajustement de faisceaux local est la suivante :

$$F^{lp} : \mathbf{x} \mapsto \begin{pmatrix} P(\mathbf{c}_i, \mathbf{s}_j), & \forall (i, j) \in \mathfrak{M}^{lp} \\ \mathbf{c}_c \end{pmatrix} \quad (5.4)$$

$$= \begin{pmatrix} F^g(\mathbf{x}) \\ \mathbf{c}_c \end{pmatrix} \quad (5.5)$$

Nous allons, comme précédemment, spécifier un modèle gaussien pour les observations \mathbf{y} (les points 2D \mathbf{m} et les caméras \mathbf{c}_c^-).

Nous supposons que $\mathbf{y} = [\mathbf{m} \mathbf{c}_c^-] \sim \mathcal{N}([\bar{\mathbf{m}} \bar{\mathbf{c}}_-], C_{[\mathbf{m} \mathbf{c}_c^-]})$.

De plus, par définition de F^{lp} , $\bar{\mathbf{y}} = [\bar{\mathbf{m}}\bar{\mathbf{c}}_c^-] = F^{lp}(\bar{\mathbf{x}})$. Notons que cela implique que $\bar{\mathbf{c}}_c = \bar{\mathbf{c}}_c^-$.

Les paramètres inconnus du modèle statistique sont \mathbf{x} (nous supposons que $C_{[\mathbf{m}\bar{\mathbf{c}}_c^-]}$ est connue et non bruitée). La fonction densité de probabilité du vecteur gaussien \mathbf{y} est :

$$d(\mathbf{y}|\bar{\mathbf{x}}) = K e^{-\frac{1}{2}\mathbf{z}^T C_y^{-1}\mathbf{z}}, \quad (5.6)$$

avec $\mathbf{z} = \mathbf{y} - F^{lp}(\bar{\mathbf{x}})$ et $K \in \mathbb{R}$.

L'estimateur au maximum de vraisemblance $f^{lp}([\mathbf{m}\bar{\mathbf{c}}_c^-])$ de $\bar{\mathbf{x}}$ déduit de l'Eq. 5.6 est :

$$\arg \min_{\bar{\mathbf{x}}} \tilde{\mathbf{z}}^T C_y^{-1} \tilde{\mathbf{z}} \text{ avec } \tilde{\mathbf{z}} = \mathbf{y} - F^{lp}(\tilde{\mathbf{x}}) = \begin{pmatrix} \mathbf{m} - F^g(\tilde{\mathbf{x}}) \\ \mathbf{c}_c^- - \tilde{\mathbf{c}}_c \end{pmatrix}. \quad (5.7)$$

Pour revenir à une formulation plus classique aux moindres carrés, les données d'entrée et la fonction de projection F^{lp} sont normalisées par la racine carrée de la covariance des données d'entrée. Nous définissons ainsi :

$$\mathbf{y}^n = C_y^{-\frac{1}{2}} \mathbf{y} \quad (5.8)$$

$$F^{lp^n} = C_y^{-\frac{1}{2}} F^{lp} \quad (5.9)$$

avec la racine carrée $C_y^{\frac{1}{2}}$ de C_y tel que :

$$C_y = C_y^{\frac{1}{2}} C_y^{\frac{1}{2}T} \quad (5.10)$$

Ceci permet de revenir à la forme classique d'une minimisation aux moindres carrés :

$$f^{lp}(\mathbf{y}) = \arg \min_{\bar{\mathbf{x}}} \|\mathbf{y}^n - F^{lp^n}(\bar{\mathbf{x}})\|_2^2. \quad (5.11)$$

Cette fonction minimise la fonction de coût suivante sous forme développée :

$$(\mathbf{x} = [\tilde{\mathbf{c}}_c \tilde{\mathbf{c}}_o \tilde{\mathbf{s}}]) \mapsto \left\| C_{[\mathbf{m}\bar{\mathbf{c}}_c^-]}^{-\frac{1}{2}} \begin{pmatrix} \mathbf{m}_{i,j} - P(\tilde{\mathbf{c}}_i, \tilde{\mathbf{s}}_j), \quad \forall (i,j) \in \mathfrak{M}^{lp} \\ \mathbf{c}_i^{t-1} - \tilde{\mathbf{c}}_i^t, \quad \forall i \in \{t-N+1 \dots t-n\} \end{pmatrix} \right\|_2^2 \quad (5.12)$$

5.3 Différence par rapport à l'AFG et l'AFL

l'AFLP précédemment formulé présente de grandes similitudes avec les ajustements de faisceaux présentés jusqu'à alors. L'ajustement de faisceaux local pondéré

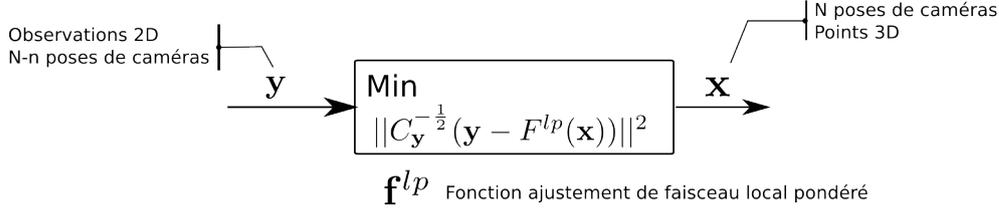


FIGURE 5.2: Entrée/sorties de l'Ajustement de faisceaux local pondéré.

peut-être vu comme la "formulation faible" de l'AFL : les contraintes fortes sur les caméras dans l'AFL sont relâchées et prises en compte de façon faible. La covariance sur les caméras précédemment estimées donne la confiance que doit donner l'ajustement de faisceaux à la pose de caméra précédemment évaluée. En effet, si $C_{c_c^-} \rightarrow 0$, l'AFL pondéré devient un AFL, les caméras sont contraintes à rester fixes durant l'optimisation. Au contraire, si $C_{c_c^-} \rightarrow \infty$, l'AFL pondéré devient un AFG sur les N dernières caméras et il ne tient pas compte de la position des caméras précédemment estimées.

5.4 Propagation d'erreur

Dans cette section, nous établissons les relations permettant de propager l'erreur pour chaque itération. Puis, nous détaillerons les modèles de bruit choisis et pour chacun, nous donnerons les formulations détaillées de propagation d'erreur et les implications en terme de récurrence entre les itérations.

Dans un premier temps, nous allons déterminer la relation entre la covariance $C_{[\mathbf{y}\mathbf{x}]}$ des données de sortie en fonction de $C_{\mathbf{y}}$ (figure 5.2).

La fonction f^{lp} est approximée par son développement linéaire de Taylor au point $\bar{\mathbf{y}}$.

$$\begin{pmatrix} \mathbf{y} \\ f^{lp}(\mathbf{y}) \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{y}} \\ f^{lp}(\bar{\mathbf{y}}) \end{pmatrix} + \begin{pmatrix} C_{\bar{\mathbf{y}}}^{\frac{1}{2}} \\ \mathbf{J}_{\bar{\mathbf{y}}^n}^{f^{lp}} \end{pmatrix} (\mathbf{y}^n - \bar{\mathbf{y}}^n). \quad (5.13)$$

Le vecteur $[\mathbf{y} f^{lp}(\mathbf{y})]$ est gaussien et sa covariance est :

$$C_{[\mathbf{y}\mathbf{x}]} = C_{[\mathbf{y} f^{lp}(\mathbf{y})]} = \begin{pmatrix} C_{\bar{\mathbf{y}}}^{\frac{1}{2}} \\ \mathbf{J}_{\bar{\mathbf{y}}^n}^{f^{lp}} \end{pmatrix} C_{\mathbf{y}^n} \begin{pmatrix} C_{\bar{\mathbf{y}}}^{\frac{1}{2}} \\ \mathbf{J}_{\bar{\mathbf{y}}^n}^{f^{lp}} \end{pmatrix}^T. \quad (5.14)$$

Par hypothèse, \mathbf{y} est un vecteur gaussien ($\mathbf{y} \sim \mathcal{N}(\bar{\mathbf{y}}, C_{\mathbf{y}})$) donc la définition de \mathbf{y}^n (Eq. 5.8) implique que $\mathbf{y}^n \sim \mathcal{N}(\bar{\mathbf{y}}^n, \mathbf{I}_{|\mathbf{y}|})$ et donc :

$$C_{[\mathbf{y}\mathbf{x}]} = \begin{pmatrix} C_{\mathbf{y}}^{\frac{1}{2}} \\ \mathbf{J}_{\mathbf{y}^n}^{f^{lp}} \end{pmatrix} \begin{pmatrix} C_{\mathbf{y}}^{\frac{1}{2}} \\ \mathbf{J}_{\mathbf{y}^n}^{f^{lp}} \end{pmatrix}^T. \quad (5.15)$$

Nous allons maintenant détailler les modèles de bruit choisis et leurs impacts sur la propagation d'erreur.

5.4.1 Cas avec corrélation

Modèle de bruit

Ici, nous détaillons le modèle de bruit dans le cas où l'on tient compte de la corrélation entre les points \mathbf{m} et l'estimation précédente des caméras contraintes \mathbf{c}_c^- . Ainsi, le modèle de bruit des données dans ce cas est le suivant :

$$\mathbf{y} \sim \mathcal{N}(\bar{\mathbf{y}}, C_{\mathbf{y}})$$

avec $C_{\mathbf{y}} = C_{[\mathbf{m}\mathbf{c}_c^-]} = \begin{pmatrix} C_{\mathbf{m}} & C_{\mathbf{m}/\mathbf{c}_c^-} \\ C_{\mathbf{c}_c^-/\mathbf{m}} & C_{\mathbf{c}_c^-} \end{pmatrix} \quad (5.16)$

De plus, nous avons fait l'hypothèse que les points 2D ont tous la même distribution d'erreur gaussienne et sont indépendants entre eux, donc : $C_{\mathbf{m}} = \sigma^2 \mathbf{I}_{|\mathbf{m}|}$

Covariance résultante

A partir de Eq. 5.15, nous déduisons :

$$C_{[\mathbf{y}\mathbf{x}]} = \begin{pmatrix} C_{\mathbf{y}} & C_{\mathbf{y}}^{\frac{1}{2}} \mathbf{J}_{\mathbf{y}^n}^{f^{lp}T} \\ \mathbf{J}_{\mathbf{y}^n}^{f^{lp}} C_{\mathbf{y}}^{\frac{T}{2}} & \mathbf{J}_{\mathbf{y}^n}^{f^{lp}} \mathbf{J}_{\mathbf{y}^n}^{f^{lp}T} \end{pmatrix} \quad (5.17)$$

Nous rappelons l'expression de la jacobienne d'une fonction implicite, et nous l'appliquons à l'ajustement de faisceaux local pondéré (défini par l'Eq. 5.11) :

$$\mathbf{J}_{\mathbf{y}^n}^{f^{lp}} = \left(\mathbf{J}_{\mathbf{x}}^{F^{lp}nT} \mathbf{J}_{\mathbf{x}}^{F^{lp}n} \right)^{-1} \mathbf{J}_{\mathbf{x}}^{F^{lp}nT} \quad (5.18)$$

Pour plus de détails, cette formule est démontrée dans la section 1.3.3.

Nous détaillons maintenant le calcul de deux parties de l'équation Eq. 5.17 :

$$C_y^{\frac{1}{2}} J_{y^n}^{f^{lp}T} = C_y^{\frac{1}{2}} J_x^{F^{lp^n}} \left(J_x^{F^{lp^n}T} J_x^{F^{lp^n}} \right)^{-1} \quad (5.19)$$

$$J_{y^n}^{f^{lp}} J_{y^n}^{f^{lp}T} = \left(\cancel{J_x^{F^{lp^n}T}} \cancel{J_x^{F^{lp^n}}} \right)^{-1} \cancel{J_x^{F^{lp^n}T}} \cancel{J_x^{F^{lp^n}}} \left(J_x^{F^{lp^n}T} J_x^{F^{lp^n}} \right)^{-1} \quad (5.20)$$

$$= \left(J_x^{F^{lp^n}T} J_x^{F^{lp^n}} \right)^{-1}. \quad (5.21)$$

Nous continuons en développant $J_x^{F^{lp^n}}$ à partir de la définition de F^{lp^n} (Eq.5.9). Nous déduisons que :

$$J_x^{F^{lp^n}} = C_y^{-\frac{1}{2}} J_x^{F^{lp}}. \quad (5.22)$$

Ainsi :

$$\begin{aligned} C_y^{\frac{1}{2}} J_{y^n}^{f^{lp}T} &= \cancel{C_y^{\frac{1}{2}}} \cancel{C_y^{-\frac{1}{2}}} J_x^{F^{lp}} \left(\left(C_y^{-\frac{1}{2}} J_x^{F^{lp}} \right)^T C_y^{-\frac{1}{2}} J_x^{F^{lp}} \right)^{-1} \\ &= J_x^{F^{lp}} \left(J_x^{F^{lp}T} C_y^{-\frac{1}{2}} C_y^{-\frac{1}{2}} J_x^{F^{lp}} \right)^{-1} \\ &= J_x^{F^{lp}} \left(J_x^{F^{lp}T} C_y^{-1} J_x^{F^{lp}} \right)^{-1} \end{aligned} \quad (5.23)$$

$$J_{y^n}^{f^{lp}} J_{y^n}^{f^{lp}T} = \left(J_x^{F^{lp}T} C_y^{-1} J_x^{F^{lp}} \right)^{-1}. \quad (5.24)$$

Nous posons :

$$H' = \left(J_x^{F^{lp^n}T} J_x^{F^{lp^n}} \right) = \left(J_x^{F^{lp}T} C_y^{-1} J_x^{F^{lp}} \right). \quad (5.25)$$

Et finalement, en rassemblant les Eq.5.23 et Eq.5.24 dans Eq.5.17 et appliquant la définition de H' (Eq.5.25) nous obtenons :

$$C_{[y^x]} = \begin{pmatrix} C_y & J_x^{F^{lp}} H'^{-1} \\ H'^{-1} J_x^{F^{lp}T} & H'^{-1} \end{pmatrix}. \quad (5.26)$$

Récurrance

La figure Fig. 5.3 présente la relation entre les variables d'entrée y du temps courant à partir des variables estimées à l'instant précédent ($[y^- x^-]$). Attention dans cette version, contrairement à l'AFL, les caméras contraintes apparaissent

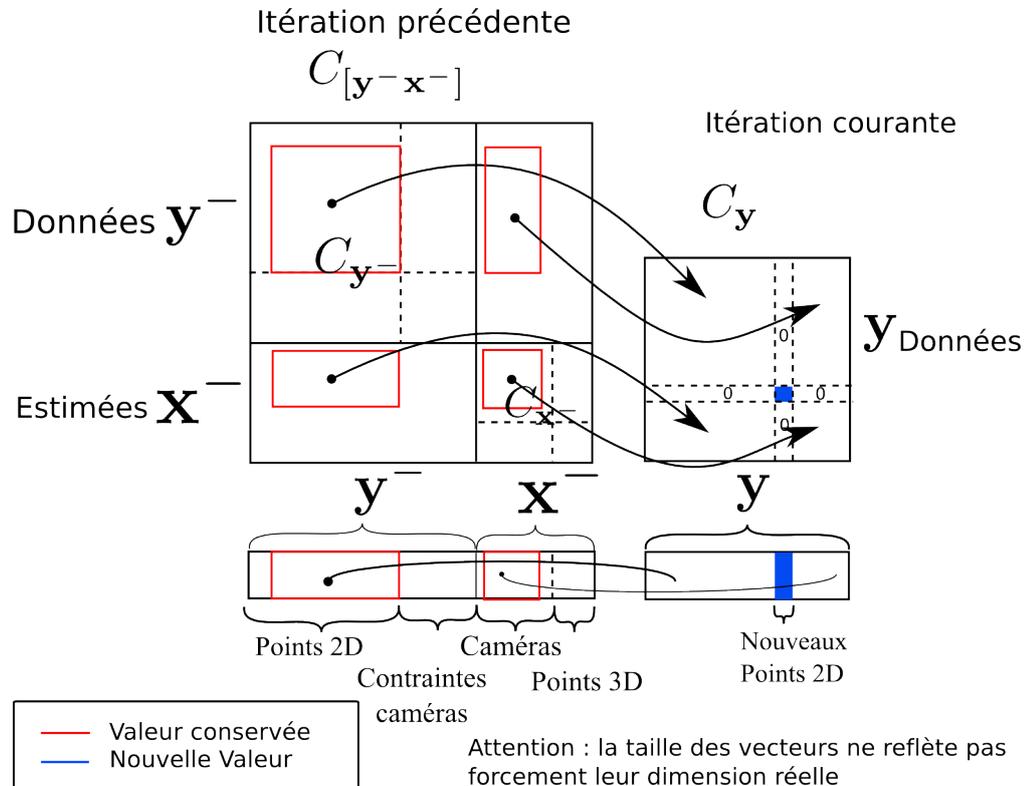


FIGURE 5.3: Récurrence pour le cas avec corrélation.

deux fois : une première fois dans le vecteur d'entrée avec leurs valeurs de l'itération précédente, une seconde fois dans le vecteur estimé avec leurs valeurs raffinés.

$y = [m c_c^-]$ est déduit de l'itération précédente, car, c_c^- est une partie de x^- et m est constitué d'une partie de m^- et de nouveaux points 2D.

La covariance C_y est une matrice contenant une sous matrice de $C_{[m^- x^-]}$ (correspondant aux paramètres des caméras contraintes c_c^- et des points 2D encore observés) et une matrice $\sigma^2 I_{|m|}$ (correspondant aux nouveaux points 2D, ceux-ci étant indépendants des éléments de $[m^- x^-]$).

La figure 5.4 résume l'algorithme proposé ici.

On estime $[\mathbf{yx}]$ et $C_{[\mathbf{yx}]}$ depuis $[\mathbf{y}^- \mathbf{x}^-]$ et $C_{[\mathbf{y}^- \mathbf{x}^-]}$ comme suit :

1. \mathbf{y} est la concaténation d'un sous-vecteur de $[\mathbf{y}^- \mathbf{x}^-]$ et des nouveaux points 2D (voir figure 5.3)
2. $C_{\mathbf{y}}$ est composée d'une sous-matrice de $C_{[\mathbf{y}^- \mathbf{x}^-]}$ et de $C_{\mathbf{m}}$ (voir figure 5.3)
3. On estime $\mathbf{x} = f^{lp}(\mathbf{y})$ en utilisant l'AFLP (Eq. 5.7)
4. On estime $C_{[\mathbf{yx}]}$ en utilisant l'Eq. 5.26

Dans cet algorithme, on estime $C_{[\mathbf{yx}]}$. Nous rappelons que l'estimation de la covariance des points 3D ($C_{\mathbf{s}}$) n'est pas nécessaire : il est possible de se limiter à l'estimation de la covariance de la partie caméra de \mathbf{x} ($C_{[\mathbf{yc}]}$), notamment pour diminuer les temps de calcul.

FIGURE 5.4: Résumé de l'estimation par l'AFLP et du calcul de covariance, en tenant compte de la corrélation entre observation et caméras déjà estimées.

5.4.2 Cas avec hypothèse d'indépendance

Cette section a pour but d'explicitier la matrice $C_{[\mathbf{yx}]}$ définie dans Eq. 5.15 dans le cas où \mathbf{m} et \mathbf{c}_c^- sont supposés indépendants.

Modèle de bruit

Sous l'hypothèse d'indépendance, le modèle de bruit Eq. 5.16 se simplifie en :

$$C_{\mathbf{y}} = C_{[\mathbf{mc}^-]} = \begin{pmatrix} C_{\mathbf{m}} & 0_{|\mathbf{m}| \times |\mathbf{c}_c^-|} \\ 0_{|\mathbf{m}| \times |\mathbf{c}_c^-|} & C_{\mathbf{c}_c^-} \end{pmatrix}. \quad (5.27)$$

De plus, $C_{\mathbf{m}}$ est toujours égale à $\sigma^2 \mathbf{I}_{|\mathbf{m}|}$ (§5.4.1).

Covariance résultante

L'expression du calcul de covariance établie pour le cas précédent est toujours valable. Nous allons maintenant intégrer les hypothèses d'indépendance pour simplifier l'expression :

$$C_{[\mathbf{yx}]} = \begin{pmatrix} C_{\mathbf{y}} & \mathbf{J}_{\mathbf{x}}^{F^{lp}} \mathbf{H}'^{-1} \\ \mathbf{H}'^{-1} \mathbf{J}_{\mathbf{x}}^{F^{lp}T} & \mathbf{H}'^{-1} \end{pmatrix}. \quad (5.28)$$

Contrairement à la section précédente, du fait de l'hypothèse d'indépendance,

il n'est plus nécessaire d'expliciter la valeur des corrélations entre les caméras et les points 2D. Ainsi, dans ce paragraphe $H'^{-1} J_x^{F^{lp}T}$ ne sera pas calculé.

Nous allons maintenant détailler le calcul de $J_x^{F^{lp^n}}$ sous les hypothèses définies au §5.4.2 (indépendance entre caméras et points 2D, points 2D indépendants identiquement distribués). De plus, nous rappelons que $\mathbf{x} = [c_c c_{os}]$. L'équation 5.22 devient :

$$\begin{aligned} J_x^{F^{lp^n}} &= C_y^{-\frac{1}{2}} J_x^{F^{lp}} \\ &= C_y^{-\frac{1}{2}} \begin{pmatrix} J_{c_c}^{F^{lp}} & J_{[c_{os}]}^{F^{lp}} \end{pmatrix} \\ &= C_y^{-\frac{1}{2}} \begin{pmatrix} J_{c_c}^{F^g} & J_{[c_{os}]}^{F^g} \\ \mathbf{I}_{|c_c|} & 0_{|c_c| \times |[c_{os}]|} \end{pmatrix} \end{aligned} \quad (5.29)$$

$$= \begin{pmatrix} \frac{1}{\sigma} J_{c_c}^{F^g} & \frac{1}{\sigma} J_{[c_{os}]}^{F^g} \\ C_{c_c}^{-\frac{1}{2}} & 0_{|c_c^-| \times |[c_{os}]|} \end{pmatrix} \quad (5.30)$$

Ainsi H' devient :

$$H' = \begin{pmatrix} J_x^{F^{lp^n}T} & J_x^{F^{lp^n}} \end{pmatrix} \quad (5.31)$$

$$= \begin{pmatrix} C_{c_c}^{-1} & 0_{|c_c^-| \times |[c_{os}]|} \\ 0_{|[c_{os}]| \times |c_c^-|} & 0_{|[c_{os}]| \times |[c_{os}]|} \end{pmatrix} + \frac{1}{\sigma^2} \underbrace{\begin{pmatrix} J_x^{F^gT} & J_x^{F^g} \end{pmatrix}}_H \quad (5.32)$$

Ainsi, nous obtenons l'expression de la covariance C_x :

$$C_x = \left(\begin{pmatrix} C_{c_c}^{-1} & 0_{|c_c^-| \times |[c_{os}]|} \\ 0_{|[c_{os}]| \times |c_c^-|} & 0_{|[c_{os}]| \times |[c_{os}]|} \end{pmatrix} + \frac{1}{\sigma^2} H \right)^{-1} \quad (5.33)$$

$$= \begin{pmatrix} C_{c_c}^{-1} + \frac{1}{\sigma^2} J_{c_c}^{F^gT} J_{c_c}^{F^g} & \frac{1}{\sigma^2} J_{c_c}^{F^gT} J_{[c_{os}]}^{F^g} \\ \frac{1}{\sigma^2} J_{[c_{os}]}^{F^gT} J_{c_c}^{F^g} & \frac{1}{\sigma^2} J_{[c_{os}]}^{F^gT} J_{[c_{os}]}^{F^g} \end{pmatrix}^{-1}. \quad (5.34)$$

Estimation rapide

Cette formulation est très proche de celle d'un ajustement de faisceaux global. La principale différence vient de la contrainte ajoutée sur les anciennes caméras. Les techniques d'estimation rapide (§3.5) sont les mêmes que pour l'ajustement de faisceaux global appliqué aux N dernières caméras.

$$C_x = \left(\begin{array}{cc|cc} C_{c_c}^{-1} + \frac{1}{\sigma^2} J_{c_c}^{FgT} J_{c_c}^{Fg} & \frac{1}{\sigma^2} J_{c_c}^{FgT} J_{c_o}^{Fg} & \frac{1}{\sigma^2} J_{c_c}^{FgT} J_s^{Fg} & \\ \frac{1}{\sigma^2} J_{c_o}^{FgT} J_{c_c}^{Fg} & \frac{1}{\sigma^2} J_{c_o}^{FgT} J_{c_o}^{Fg} & \frac{1}{\sigma^2} J_{c_o}^{FgT} J_s^{Fg} & \\ \frac{1}{\sigma^2} J_s^{FgT} J_{c_c}^{Fg} & \frac{1}{\sigma^2} J_s^{FgT} J_{c_o}^{Fg} & \frac{1}{\sigma^2} J_s^{FgT} J_s^{Fg} & \end{array} \right)^{-1} \quad (5.35)$$

$$= \begin{pmatrix} U & W^T \\ W & V \end{pmatrix}^{-1} \quad (5.36)$$

On remarque que la partie caméra U n'est plus diagonale par blocs en raison de l'ajout des contraintes. Cependant, cela n'influe en rien sur l'inversion par blocs de la matrice.

On peut également (de la même manière que pour l'AFG) estimer seulement la covariance de la partie caméra U (cf.§5.4.2).

Récurrance

Itération précédente $C([\mathbf{y}^- \mathbf{x}^-])$

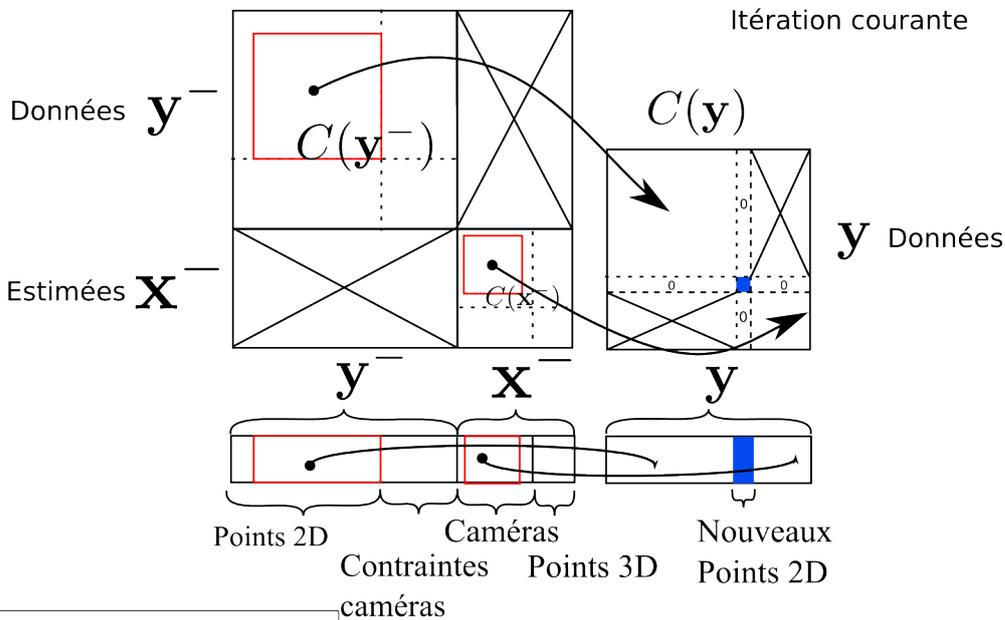


FIGURE 5.5: Récurrance pour le cas avec indépendance.

La figure Fig. 5.5 présente les relations entre les variables d'entrée \mathbf{y} du temps courant à partir des variables estimées à l'instant précédent ($[\mathbf{y}^- \mathbf{x}^-]$). Comme pour le cas avec corrélation, les caméras contraintes apparaissent deux fois : une première fois dans le vecteur d'entrée avec leurs valeurs de l'itération précédente, une seconde fois dans le vecteur estimé avec leurs valeurs raffinés. $\mathbf{y} = [\mathbf{m} \mathbf{c}_c^-]$ est déduit depuis l'itération précédente : \mathbf{c}_c^- est une partie de \mathbf{x}^- , \mathbf{m} est constitué d'une partie de \mathbf{m}^- et de nouveaux points 2D.

Pour la covariance, $C_{\mathbf{y}}$ est une matrice composée d'une sous matrice de $C_{\mathbf{x}^-}$ correspondant aux vecteurs des caméras contraintes \mathbf{c}_c^- et d'une matrice $\sigma^2 \mathbf{I}_{|\mathbf{m}|}$ pour les points 2D. Une fois ces précalculs terminés, on peut les utiliser pour calculer $C_{\mathbf{x}}$ à partir de $C_{\mathbf{y}}$.

La figure 5.6 résume l'algorithme proposé ici dans le cadre de l'hypothèse d'indépendance.

On peut estimer \mathbf{x} et $C_{\mathbf{x}}$ depuis \mathbf{x}^- et $C_{\mathbf{x}^-}$ comme suit :

1. \mathbf{y} est déterminée à partir de $[\mathbf{m}, \mathbf{x}^-]$ (voir figure 5.5)
2. $C_{\mathbf{y}}$ est déterminée à partir de $C_{\mathbf{m}}, C_{\mathbf{x}^-}$ (voir figure 5.5)
3. On estime $\mathbf{x} = f_t^{lp}(\mathbf{y})$ en utilisant l'AFLP (Eq. 5.11)
4. On estime $C_{\mathbf{x}}$ en utilisant l' Eq. 5.34

Dans cet algorithme, on estime $C_{\mathbf{x}}$. Nous rappelons que l'estimation de la covariance des points 3D ($C_{\mathbf{s}}$) n'est pas nécessaire : il est possible de se limiter à l'estimation de la covariance partie caméra de \mathbf{x} ($C_{[\mathbf{c}_c \mathbf{c}_o]}$), notamment pour diminuer les temps de calcul (cf §5.4.2).

FIGURE 5.6: Résumé de l'estimation par l'AFLP et du calcul de covariance, cas avec hypothèse d'indépendance entre observation et caméras déjà estimées.

Récurrence "minimale"

Dans le cas où l'on souhaite uniquement propager la covariance d'itération en itération, il n'est pas nécessaire de calculer entièrement $C_{\mathbf{x}}$ mais seulement la partie caméra. Nous avons vu au cours de cette section que ce calcul peut se mettre sous la forme suivante :

$$C_{\mathbf{x}} = \left(\begin{pmatrix} C_{\mathbf{c}_c}^{-1} & 0_{|\mathbf{c}_c^-| \times |\mathbf{c}_o \mathbf{s}|} \\ 0_{|\mathbf{c}_o \mathbf{s}| \times |\mathbf{c}_c^-|} & 0_{|\mathbf{c}_o \mathbf{s}| \times |\mathbf{c}_o \mathbf{s}|} \end{pmatrix} + \frac{1}{\sigma^2} \mathbf{H} \right)^{-1}. \quad (5.37)$$

Nous posons :

$$H = \begin{pmatrix} U & W^T \\ W & V \end{pmatrix}.$$

En développant la formulation précédente avec l'expression des matrices de par blocs, nous obtenons :

$$C_x = \left(\begin{pmatrix} C_{\mathbf{c}_c}^{-1} & 0_{|\mathbf{c}_c^-| \times |\mathbf{c}_o|} \\ 0_{|\mathbf{c}_o| \times |\mathbf{c}_c^-|} & 0_{|\mathbf{c}_o| \times |\mathbf{c}_o|} \end{pmatrix} + \frac{1}{\sigma^2} \begin{pmatrix} U & W^T \\ W & V \end{pmatrix} \right)^{-1}. \quad (5.38)$$

En appliquant l'inversion par bloc sur l'équation précédente, nous établissons l'expression de la covariance de la partie caméra :

$$C_{[\mathbf{c}_c \mathbf{c}_o]} = \left(\begin{pmatrix} C_{\mathbf{c}_c}^{-1} & 0_{|\mathbf{c}_c| \times |\mathbf{c}_o|} \\ 0_{|\mathbf{c}_o| \times |\mathbf{c}_c|} & 0_{|\mathbf{c}_o| \times |\mathbf{c}_o|} \end{pmatrix} + \frac{1}{\sigma^2} (U - W^T V^{-1} W) \right)^{-1}. \quad (5.39)$$

L'équation 5.39 permet de propager la covariance en temps réel.

5.5 Résumé de l'ajustement de faisceaux local pondéré

Les entrées/sorties :

Ensemble des observations utilisées : $\mathfrak{M}^{lp} = \{(i, j) \in \mathfrak{M} / t - N < i \leq t\}$

Les données $\mathbf{y} = [\dots \mathbf{m}_{i,j} \dots | \mathbf{c}_c^-]$: les points 2D et la position des caméras contraintes à l'itération précédente.

Les estimées $\mathbf{x} = [\mathbf{c}_c \mathbf{c}_o | \dots \mathbf{s}_k \dots]$: L'ensemble des caméras de la fenêtre glissante et les points 3D qu'elles observent.

Avec $(i, j) \in \mathfrak{M}^{lp}, k \in \{j / \exists (i, j) \in \mathfrak{M}^{lp}\}$

ajustement de faisceau local pondéré

Fonction de projection de l'ajustement de faisceaux local pondéré :

$$F^{lp} : \mathbf{x} \mapsto [F^g(\mathbf{x}) | \mathbf{c}_c]$$

Fonction ajustement de faisceaux local pondéré :

$$f^{lp}(\mathbf{y}) = \arg \min_{\tilde{\mathbf{x}}} \|\mathbf{y}^n - F^{lp^n}(\tilde{\mathbf{x}})\|_2^2$$

Covariance avec corrélation

Le modèle de bruit :

$$C_{\mathbf{y}} = C_{[\mathbf{m} \mathbf{c}_c^-]} = \begin{pmatrix} C_{\mathbf{m}} & C_{\mathbf{m}/\mathbf{c}_c^-} \\ C_{\mathbf{c}_c^-/\mathbf{m}} & C_{\mathbf{c}_c^-} \end{pmatrix} \text{ avec } C_{\mathbf{m}} = \sigma^2 \mathbf{I}_{|\mathbf{m}|}$$

La propagation d'erreur s'écrit :

$$C_{[\mathbf{y} \mathbf{x}]} = \begin{pmatrix} C_{\mathbf{y}} & \mathbf{J}_{\mathbf{x}}^{F^{lp}} \mathbf{H}^{-1} \\ \mathbf{H}^{-1} \mathbf{J}_{\mathbf{x}}^{F^{lp}T} & \mathbf{H}^{-1} \end{pmatrix} \text{ avec } \mathbf{H} = \begin{pmatrix} \mathbf{J}_{\mathbf{x}}^{F^{lp^n}T} & \mathbf{J}_{\mathbf{x}}^{F^{lp^n}} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{\mathbf{x}}^{F^{lp}T} & C_{\mathbf{y}}^{-1} \mathbf{J}_{\mathbf{x}}^{F^{lp}} \end{pmatrix}$$

Covariance avec indépendance

Dans ce cas la fonction de coût se simplifie et peut s'écrire sous cette forme :

$$f^{lp}(\mathbf{y}) = \arg \min_{\tilde{\mathbf{x}}} \frac{1}{\sigma^2} \|\mathbf{m} - F^g(\tilde{\mathbf{x}})\|_2^2 + (\mathbf{c}_c^- - \tilde{\mathbf{c}}_c)^T C_{\mathbf{c}_c^-}^{-1} (\mathbf{c}_c^- - \tilde{\mathbf{c}}_c)$$

Le modèle de bruit utilisé est :

$$C_{\mathbf{y}} = C_{[\mathbf{m} \mathbf{c}_c^-]} = \begin{pmatrix} C_{\mathbf{m}} & 0_{|\mathbf{m}| \times |\mathbf{c}_c^-|} \\ 0_{|\mathbf{m}| \times |\mathbf{c}_c^-|} & C_{\mathbf{c}_c^-} \end{pmatrix} \text{ avec } C_{\mathbf{m}} = \sigma^2 \mathbf{I}_{|\mathbf{m}|}$$

La propagation d'erreur s'écrit :

$$C_{\mathbf{x}} = \left(\begin{pmatrix} C_{\mathbf{c}_c^-}^{-1} & 0_{|\mathbf{c}_c^-| \times |[\mathbf{c}_o \mathbf{s}]|} \\ 0_{|[\mathbf{c}_o \mathbf{s}]| \times |\mathbf{c}_c^-|} & 0_{|[\mathbf{c}_o \mathbf{s}]| \times |[\mathbf{c}_o \mathbf{s}]|} \end{pmatrix} + \frac{1}{\sigma^2} \mathbf{H} \right)^{-1}$$

Chapitre 6

Validation expérimentale du calcul d'incertitude

Sommaire

6.1	Introduction	80
6.2	Présentation des séquences	81
6.3	Comparaison avec l'incertitude de l'AFG	81
6.3.1	Critères de comparaison	82
6.3.2	Vérifications préliminaires sur le comportement de l'incertitude	82
6.3.3	Résultat sur des séquences de simulation	84
6.3.4	Résultat sur la séquence de synthèse	89
6.3.5	Résultat sur une séquence réelle	92
6.3.6	Temps de calcul	96
6.3.7	Conclusion	96
6.4	Monte-Carlo	97
6.4.1	Méthodologie pour le test d'une application de l'AFL	97
6.4.2	Présentation des résultats	97
6.4.3	Analyses des résultats	102
6.4.4	Complément d'analyse	102
6.5	Test de Consistance	103
6.5.1	Définition	103
6.5.2	Résultat	104
6.6	Correction de la méthode AFLP	107
6.7	Conclusion	107

Dans les deux précédents chapitres, nous nous sommes attachés à décrire deux ajustements de faisceaux locaux et les méthodes permettant de propager l'incertitude en leur sein. Dans ce chapitre, nous avons la volonté de valider par des méthodes expérimentales les hypothèses faites dans les chapitres précédents, et ainsi démontrer l'a propos des méthodes développées pour propager l'incertitude dans un AFL.

6.1 Introduction

Le but de cette étude est de trouver une méthode permettant de répondre au problème d'estimation de la covariance et de sa propagation dans l'ajustement de faisceaux local. Nous cherchons une méthode qui fournit une estimation de l'incertitude des caméras de la fenêtre glissante dans le repère du début de la trajectoire (définie par les premières caméras). Ce but doit être atteint, tout en conservant les critères temps réel, puisque l'on se place dans un contexte embarqué.

Dans les deux précédents chapitres, quatre méthodes de propagation d'incertitude ont été présentées. Elles sont issues de deux ajustements de faisceaux différents et de deux modélisations du problème.

Les deux méthodes tenant compte de la corrélation entre les caméras et les points 2D (AFL avec corrélation et AFLP avec corrélation) sont des méthodes très coûteuses en temps de calcul et ne permettent pas d'obtenir l'estimation des covariances en temps réel. Les deux autres méthodes (AFL avec indépendance et AFLP avec indépendance) permettent d'obtenir de telles performances, mais l'utilisation de l'hypothèse d'indépendance entraîne des simplifications pouvant conduire à des résultats erronés. Il nous faut donc vérifier la qualité des résultats obtenus sur les covariances estimées.

Des essais ont montré que la méthode AFL avec indépendance diverge : les hypothèses simplificatrices étant trop fortes, celle-ci ne donne pas d'estimation utilisable de la covariance. Par ailleurs, la méthode AFLP avec corrélation ne peut répondre à nos attentes en terme de temps de calcul et n'a pas été implémentée.

Dans ce chapitre, nous nous intéressons donc seulement à deux méthodes :

- la méthode AFL avec corrélation ;
- la méthode AFLP avec indépendance.

6.2 Présentation des séquences

Nous allons tester nos algorithmes sur trois types de séquences.

1. Des séquences de simulation

Le nuage de points 3D et les positions de caméras sont synthétisés et bruités de manière à simuler un véhicule qui avance dans son environnement. Comme pour les séquences réelles, les points 3D sont observés en moyenne sur 5 images. L'avantage de ce type de séquence est que la vérité terrain, les paramètres de caméras et le niveau de bruit sont parfaitement connus et contrôlés. Elles seront particulièrement utiles pour les vérifications effectuées par simulation de Monte-Carlo (en section 6.4) et de consistance (en section 6.5). Par contre, les séquences ne reflètent pas rigoureusement la réalité.

2. Des séquences de synthèse

Ces séquences sont générées à partir d'un environnement 3D synthétique modélisé dans un logiciel 3D (ici 3D studio max). Des images de ces modèles sont générées suivant une trajectoire connue. L'avantage de ces séquences est la connaissance parfaite de la vérité terrain des caméras et de leurs paramètres intrinsèques. Le bruit dans les images n'est pas connu. L'inconvénient est ici le non-réalisme des textures.

3. Des séquences réelles.

Ces séquences ont été acquises en environnement urbain et peri-urbain. L'avantage de ces séquences est de décrire correctement la réalité de l'application. Dans ce cas, les paramètres de calibrage sont approximatifs puisque estimés préalablement à l'aide d'une mire [65]. De même, le bruit dans les images n'est pas connu. La vérité terrain est soit inconnue (dans ce cas, il reste la possibilité de se référer à une image satellite ou à une carte pour comparaison), soit donnée par un trajectomètre.

Le tableau 6.1 résume les avantages et les inconvénients de chaque type de séquence.

6.3 Comparaison avec l'incertitude de l'AFG

Dans ce paragraphe, nous allons comparer les résultats du calcul d'incertitude des deux méthodes retenues (l'AFL avec corrélation, l'AFLP avec indépendance) par rapport à celle de l'AFG. En effet, l'optimisation par un ajustement de faisceau local donne des résultats similaires à ceux obtenus par un ajustement de faisceau global. L'incertitude résultante devrait donc être similaire ou au moins supérieure.

Séquences / Paramètre	de Simulation	de Synthèse	Réel
Type de données	position de point 2D & 3D, caméras	images	images
Paramètres intrinsèques	connus	connus	mesurés
Bruit dans les "images"	connu	inconnu	inconnu
Réalisme	synthétique	bruit de détection réaliste	réel
Vérité terrain	connue	connue pour les caméras	mesurée ou inconnue

TABLE 6.1: Caractérisation des séquences utilisées.

6.3.1 Critères de comparaison

Plutôt que de comparer directement les différentes covariances, nous allons nous intéresser aux ellipses d'incertitude associées. Celles-ci permettent de visualiser graphiquement les covariances. Nous rappelons la définition de l'ellipsoïde d'incertitude à 90% (voir section 1.2.4) :

$$(\mathbf{x} - \bar{\mathbf{x}})^T C_{\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \leq 6.25 \quad (6.1)$$

avec \mathbf{x} un vecteur aléatoire et $C_{\mathbf{x}}$ sa covariance.

Ici, nous nous intéressons à l'incertitude projetée sur le plan du sol et nous utiliserons les critères suivants :

1. la longueur du grand axe de l'ellipse projetée,
2. l'orientation de l'ellipse définie par l'angle entre l'axe des abscisses et le grand axe,
3. le rapport petit axe sur grand axe (rapport d'aspect).

Pour être sûr d'avoir les mêmes données d'entrées pour les différentes méthodes, l'AFL est conservé pour l'optimisation géométrique. Par contre, la propagation d'erreur est faite avec chacune des méthodes développées.

6.3.2 Vérifications préliminaires sur le comportement de l'incertitude

Un premier essai consiste à vérifier que les méthodes convergent. Pour cela, ici nous présentons dans la Fig.6.1 l'incertitude calculée par les différentes méthodes sur une séquence réelle (la vue de dessus est présentée Fig.6.19 page 108).

Une deuxième expérience consiste à faire varier le niveau de bruit d'entrée appliqué sur les points 2D d'une séquence de simulation. Nous vérifions ensuite

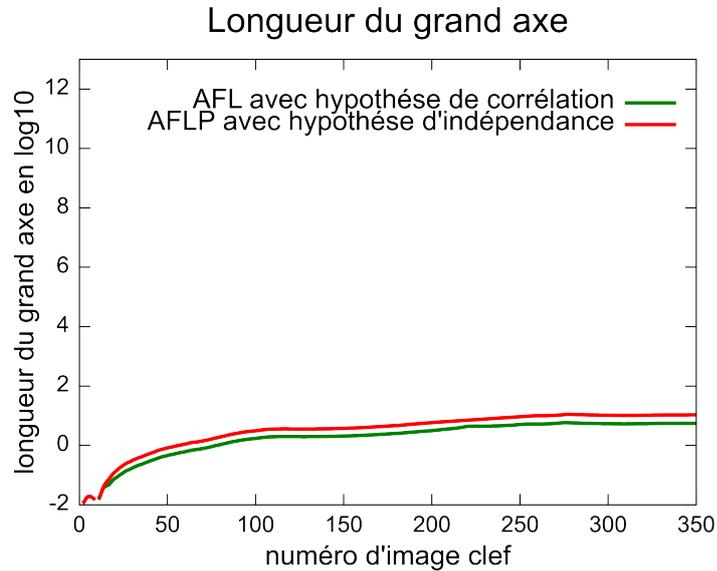


FIGURE 6.1: Longueur du grand axes des ellipses d'incertitude en \log_{10} pour chacune des méthodes : L'AFL avec hypothèse de corrélation, l'AFLP avec indépendance.

que la covariance estimée réagit correctement à différents niveaux de bruit et augmente bien avec le carré du bruit. La figure 6.2 présente les résultats pour les deux méthodes retenues pour des niveaux de bruit 2D variant de 0.0625 à 2. La figure représente la longueur du grand axe de l'ellipse d'incertitude de la position 3D de la caméra sur une échelle logarithmique pour chaque niveau de bruit. Nous vérifions ainsi que la covariance estimée se comporte de la manière attendue, c'est-à-dire que la covariance augmente avec le temps et que l'estimation ne diverge pas. Nous vérifions également que la longueur du grand axe varie linéairement avec le bruit.

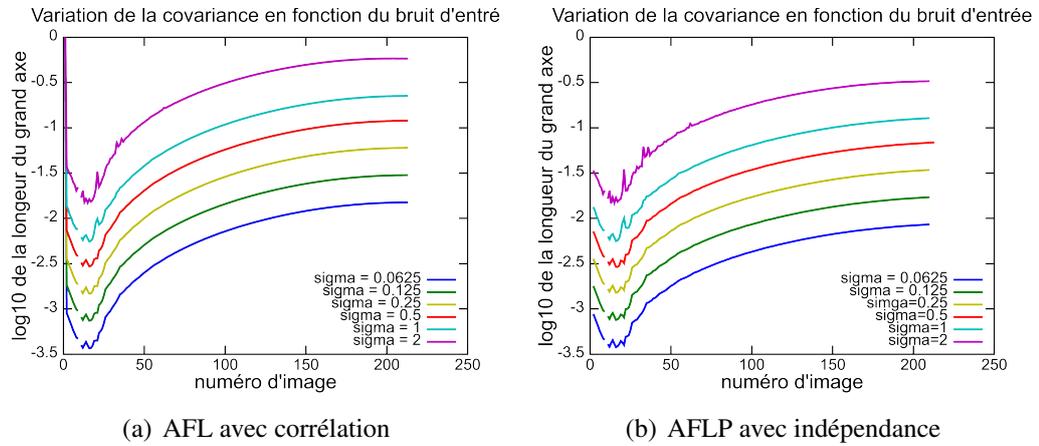


FIGURE 6.2: Variation de la covariance en fonction du bruit sur les points 2D.

6.3.3 Résultat sur des séquences de simulation

Dans cette section, nous utilisons les deux séquences de simulation avec un bruit 2D fixé de variance $\sigma^2 = 0.5$.

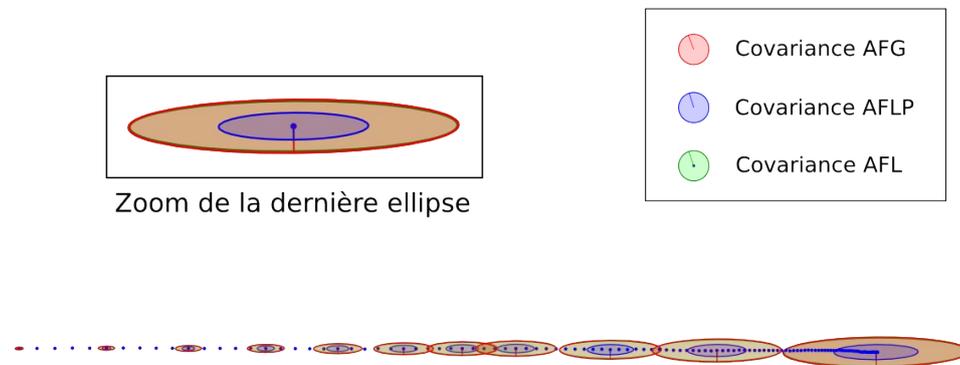


FIGURE 6.3: Vue de dessus des différentes incertitudes (AFG, AFL avec corrélation, AFLP avec indépendance). On remarquera que l'ellipse d'incertitude de l'AFL est presque confondue avec celui de l'AFG.

Séquence ligne droite Dans un premier temps, nous vérifions le comportement de la méthode sur une ligne droite avec $\sigma^2 = 0.5$. La figure 6.3 présente une projection en vue de dessus des différentes covariances. Sur cette figure, nous pouvons tout d'abord vérifier que l'incertitude croît en fonction du temps et que

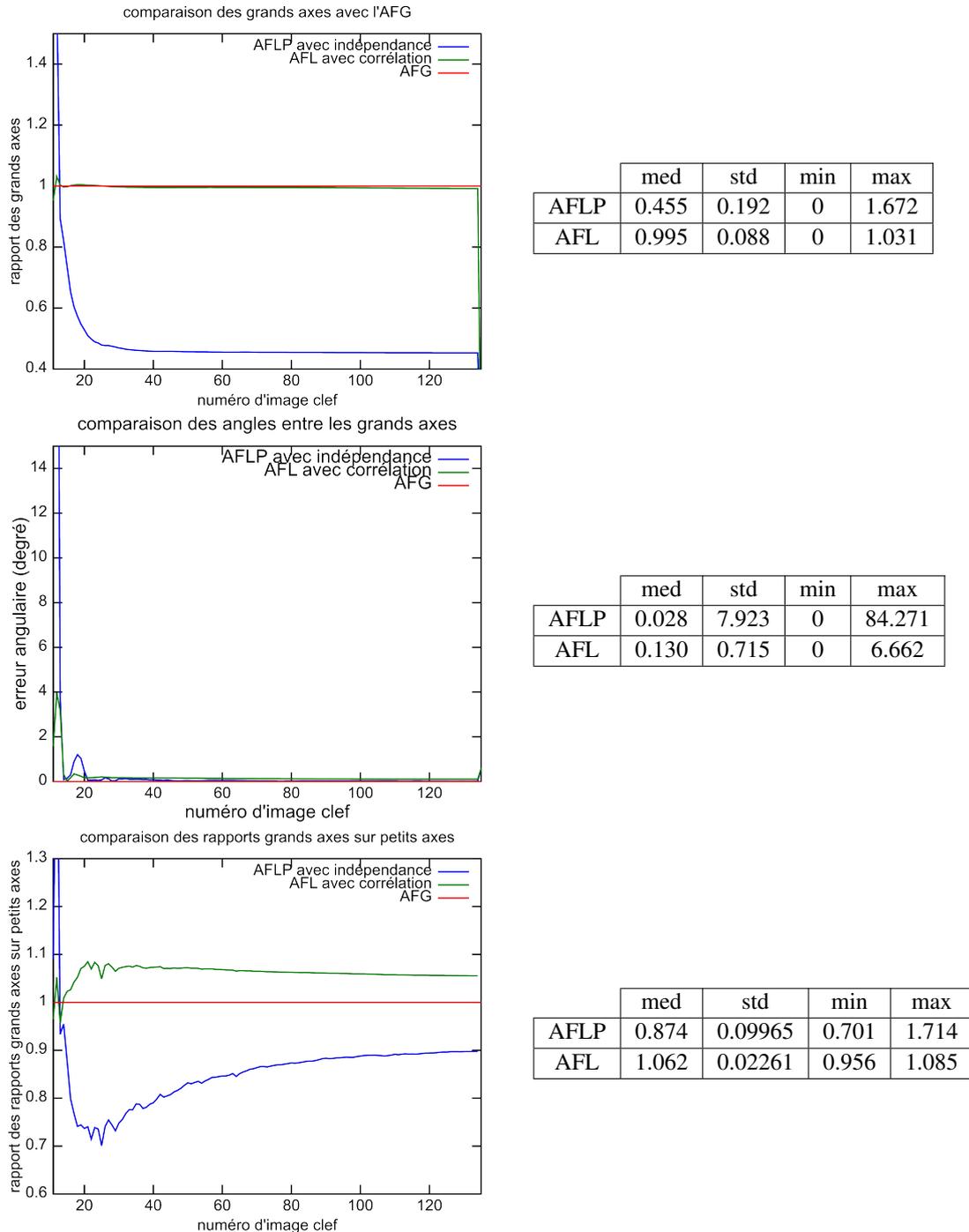


FIGURE 6.4: Comparaison sur la séquence en ligne droite avec la covariance globale, c'est-à-dire la comparaison des ellipses d'incertitude obtenues par rapport aux ellipses d'incertitudes de l'AFG. Cette figure présente les statistiques sur les différents critères (rapport des grands axes, erreur d'orientation et rapport d'aspect) ainsi que leurs évolutions au cours du temps.

l'on est donc bien en train de propager l'incertitude au cours des itérations. La figure 6.4 présente les statistiques associées. Cette dernière compare pour chacune des deux méthodes, les ellipses d'incertitudes estimées avec celles obtenues pour un AFG : le rapport de la longueur du grand axe, l'erreur d'orientation et du rapport d'aspect. Tout d'abord, l'orientation des ellipses est correcte pour les deux méthodes étudiées. Le rapport d'aspect est surestimé par l'AFL et sous-estimé par l'AFLP. L'estimation de la dimension du grand axe (fig.6.4 haut) est correcte pour l'AFL avec corrélation, mais incorrecte pour L'AFLP avec indépendance. Cependant, nous observons que le rapport de la longueur des grands axes de la méthode AFLP par rapport à ceux de l'AFG converge vers une constante.

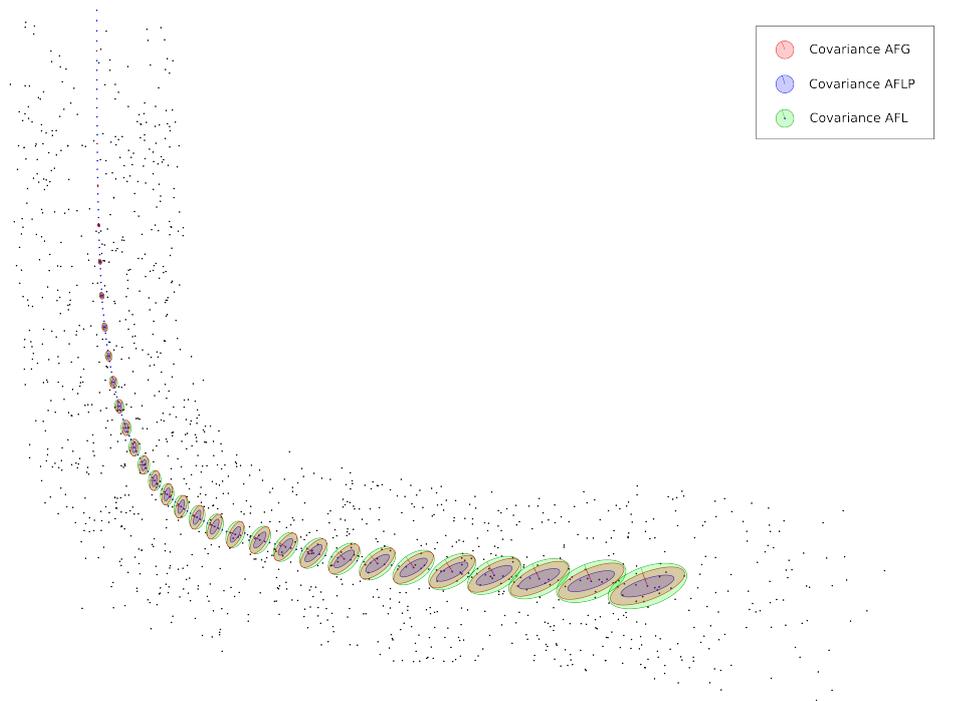


FIGURE 6.5: Vue de dessus de la séquence en virage.

Séquence virage La seconde séquence utilisée est aussi une séquence de simulation. Contrairement à la précédente, la trajectoire suit une courbe. Elle permet de tester l'influence d'un changement de direction et d'évaluer l'estimation de la forme de l'ellipse.

Sur cette séquence, l'AFL donne toujours une estimation correcte. En effet, la longueur du grand axe est correcte et l'erreur angulaire, bien que plus importante

que sur la séquence précédente reste acceptable. On note également que la forme de l'ellipsoïde n'est pas totalement respectée. Pour l'AFLP, encore une fois, la longueur du grand axe n'est pas respectée. Par contre, le rapport du grand axe par rapport à celui de l'AFG converge toujours vers une constante. L'ellipse est correctement orientée, et comme pour l'AFL la forme de l'ellipse n'est pas totalement respectée.

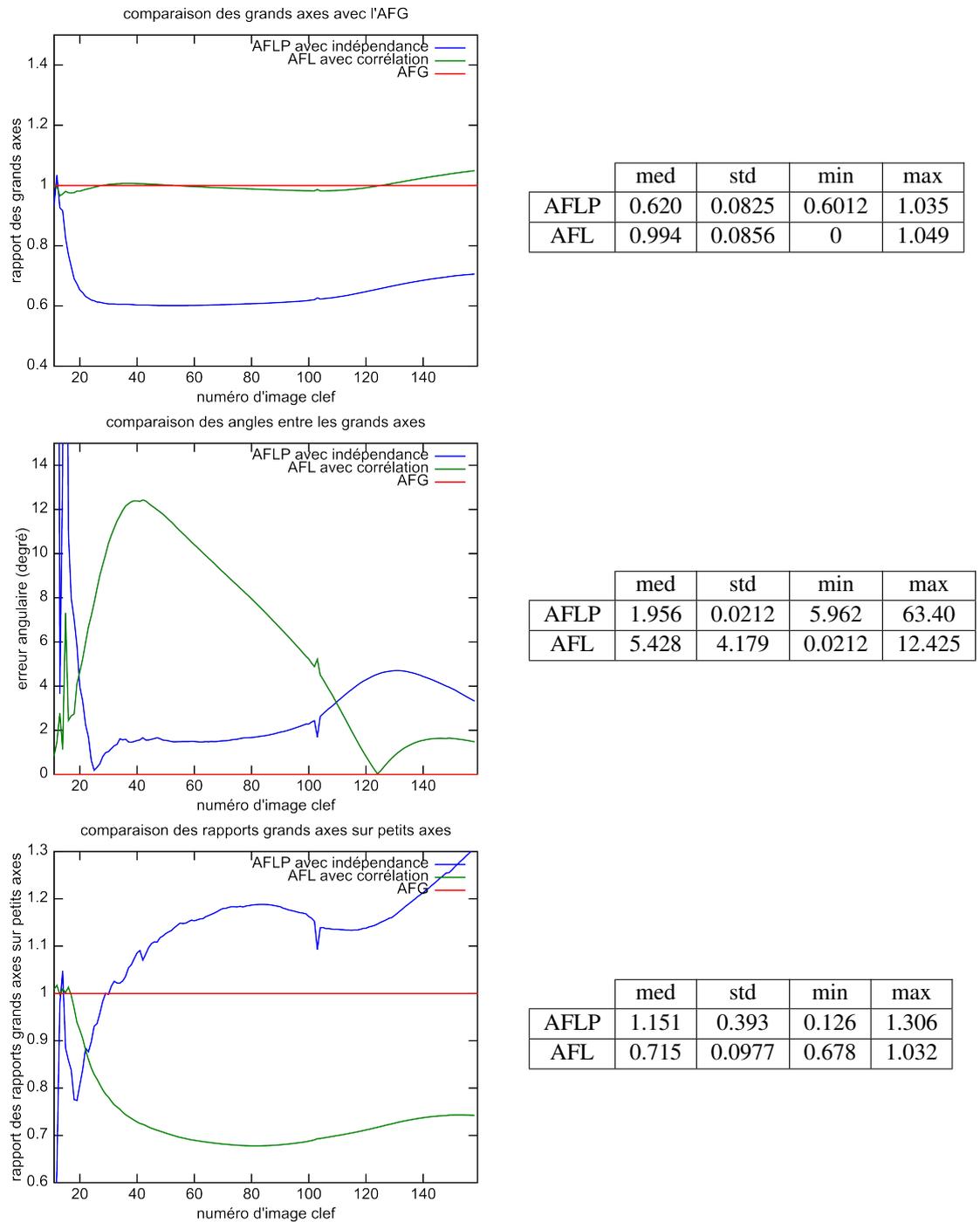


FIGURE 6.6: Comparaison avec la covariance globale sur la séquence en virage. Cette figure présente l'évolution du rapport des grands axes, de l'erreur angulaire et du rapport des rapports d'aspect pour les deux méthodes étudiées ici.

6.3.4 Résultat sur la séquence de synthèse



FIGURE 6.7: Trois images de la séquence de synthèse.

Cette séquence est la première séquence qui utilise toute la chaîne de localisation, c'est-à-dire qui extrait l'information 3D depuis les images. La séquence utilisée ici représente un couloir de 300m, les murs étant couverts avec des textures de synthèse. La figure Fig.6.7 présente un exemple des 2900 images générées. La reconstruction sélectionne 233 images clef et reconstruit 6938 points 3D.

Les figures 6.8 et 6.9 montrent les résultats.

Ici, nous voyons clairement que la trajectoire reconstruite ne boucle pas et que quelle que soit la méthode d'estimation de covariance utilisée, l'estimation n'est plus intègre (la vérité terrain n'est pas dans l'ellipse d'incertitude). Ce phénomène est dû à la dérive du facteur d'échelle au cours de l'estimation. Cette dérive est clairement observable sur le nuage de points reconstruit. Le couloir réel a la même largeur tout au long de la séquence, sur la reconstruction ce n'est plus le cas. Cette dérive est étudiée plus en détail dans le chapitre suivant (Chap. 7). Le facteur d'échelle n'est pas observable en monoculaire. La propagation d'erreur utilise la même fonction que l'estimation, et donc ici aussi la dérive du facteur d'échelle n'est pas prise en compte dans l'estimation de la covariance. Malgré cette dérive, la comparaison entre les différentes méthodes confirme les résultats précédents. C'est-à-dire que les deux méthodes propagent une incertitude (l'incertitude croît bien avec le temps). Nous remarquons que sur cette séquence plus réaliste, les formes estimées sont meilleures que sur les séquences précédentes. Le grand axe est surestimé par l'AFL (c'est le comportement attendu et il peut être attribué à l'utilisation de la méthode itérative). Enfin, pour l'AFLP, nous retrouvons la même sous-estimation du grand axe, mais toujours avec ce phénomène de convergence vers une constante.

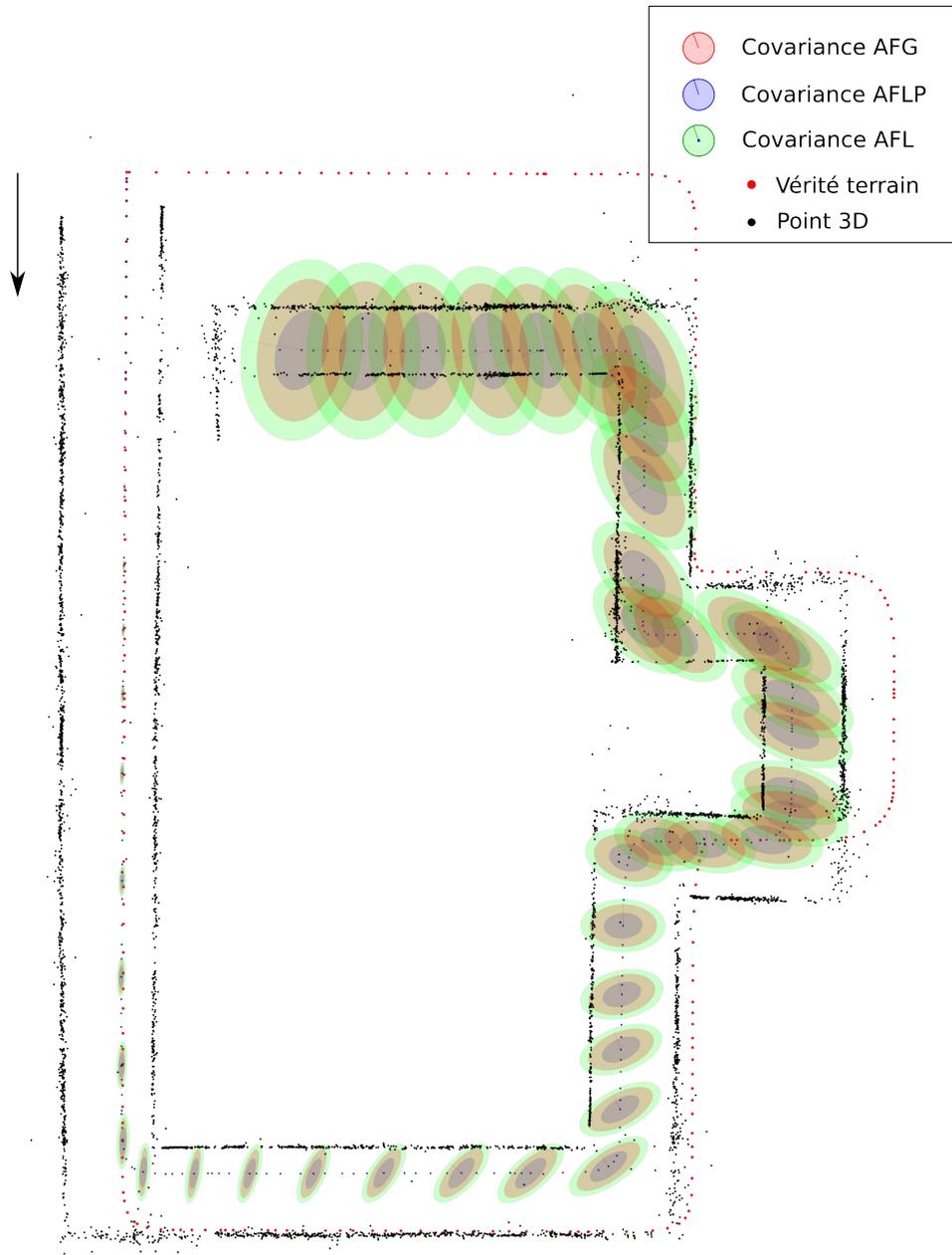


FIGURE 6.8: Vue de dessus de la séquence de synthèse.

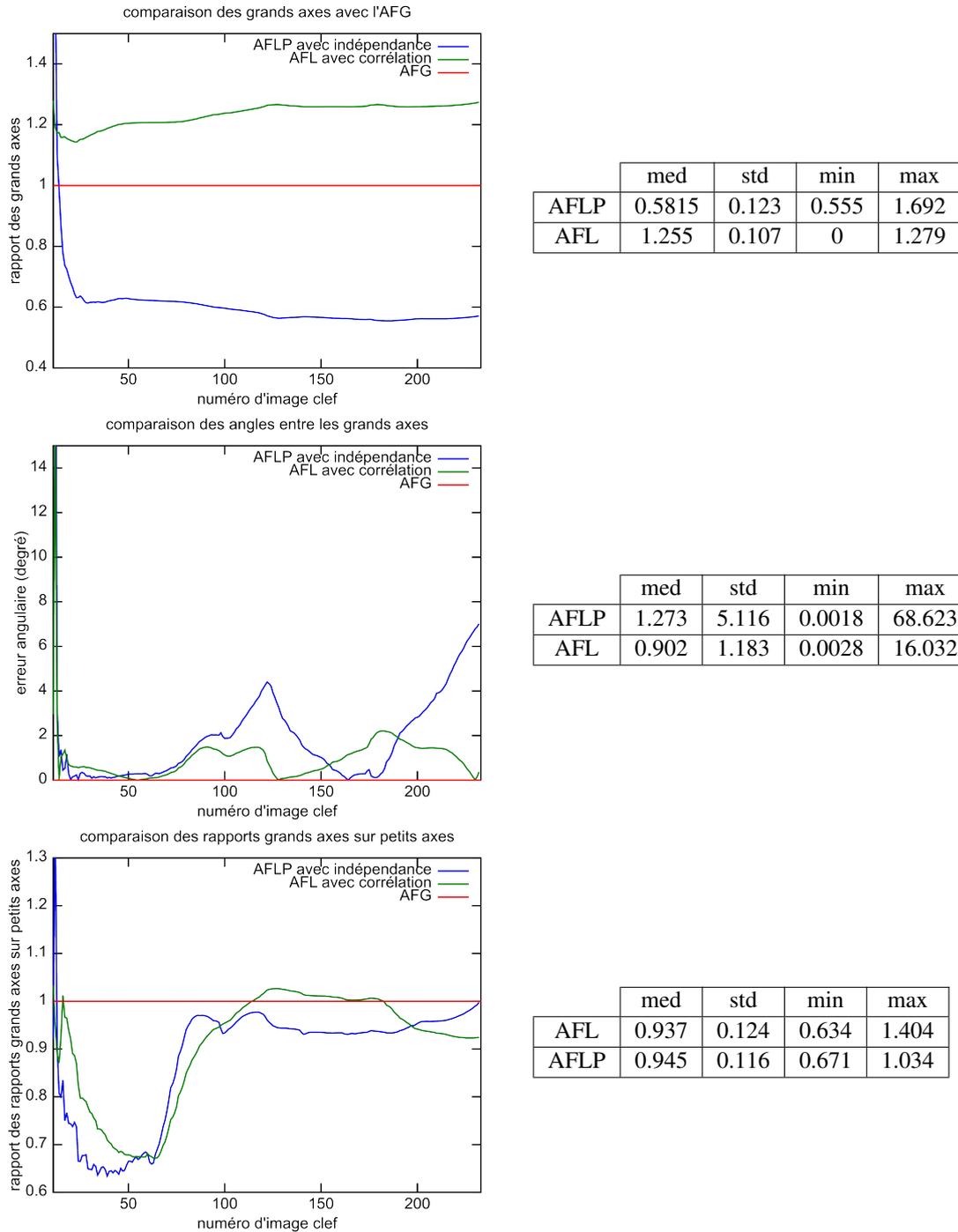


FIGURE 6.9: Comparaison avec la covariance globale sur la séquence d'images de synthèse. Cette figure présente les statistiques sur les différents critères ainsi que leurs évolutions au cours du temps.

6.3.5 Résultat sur une séquence réelle

Nous présentons maintenant les résultats sur une séquence réelle de 4km. Cette séquence a été enregistrée en milieu urbain à St Quentin en Yvelines dans des conditions normales de circulation (véhicules en mouvement dans le champ de vue de la caméra, présence de piétons, arrêts au stop et feu rouge). Cette séquence est une séquence difficile pour le SLAM visuel, car elle traite des environnements urbains et semi-urbains à une vitesse souvent élevée (50km/h). Cela occasionne des perturbations dans les images : du flou dû au mouvement et des changements d'illumination parfois importants, mais aussi au niveau géométrique à cause des occultations et des objets en mouvement. Trois des 16000 images de la séquence sont présentées dans la Fig. 6.11.

La vérité terrain sur cette séquence est connue grâce à un trajectomètre. C'est un capteur couplant un GPS et une centrale inertielle qui permet d'obtenir des trajectoires précises. Le trajectomètre utilisé pour cette séquence est le modèle landINS de chez IXSEA (Fig. 6.10) Cet instrument de mesure permet normalement d'obtenir de grande précision (à 50 cm environ). En pratique, sur nos données réelles, localement, nous avons observé des erreurs allant jusqu'à environ 1m. La documentation du capteur est fournie en annexe.

Sur cette séquence, le SLAM visuel est utilisé conjointement avec l'odomètre pour éviter une trop grande dérive du facteur d'échelle. La méthode de fusion est celle présentée §8.3. Dans cette méthode, l'odomètre est juste utilisé pour initialiser la géométrie. Le raffinement de la géométrie par ajustement de faisceaux est inchangé et donc les méthodes de calculs d'incertitudes (AFL ou AFLP) restent valables. La reconstruction sélectionne 1373 images clef (voir §7.1.3) et reconstruit 37505 points 3D.



FIGURE 6.10: Photographie du trajectomètre utilisé.



FIGURE 6.11: Trois images de la séquence réelle.

De même que précédemment, nous effectuons une comparaison entre les différentes méthodes de calcul de covariance. Pour des raisons de taille mémoire et de temps de calcul, l'incertitude de l'AFG n'a pu être calculée que sur les 600 premières images clefs. Les statistiques seront donc données sur cette première partie dans la Fig. 6.13. Les résultats sur cette séquence réelle sont du même ordre que les séquences d'images de synthèses présentées précédemment.

La Fig. 6.12 présente une vue de dessus de la trajectoire et l'ellipsoïde de confiance pour chacune des méthodes d'estimation utilisées. Elle présente de plus la vérité terrain en rouge. On pourra vérifier que l'estimation est intègre tout au long de la trajectoire pour la méthode AFL contrairement à la méthode AFLP qui n'est pas intègre en certains points. Cette fois-ci, les statistiques montrent que pour les deux méthodes l'erreur de l'orientation de l'ellipse est petite et la forme de l'ellipse est bien mieux respectée que dans les cas précédents. Nous retrouvons aussi les mêmes résultats concernant la valeur du grand axe.

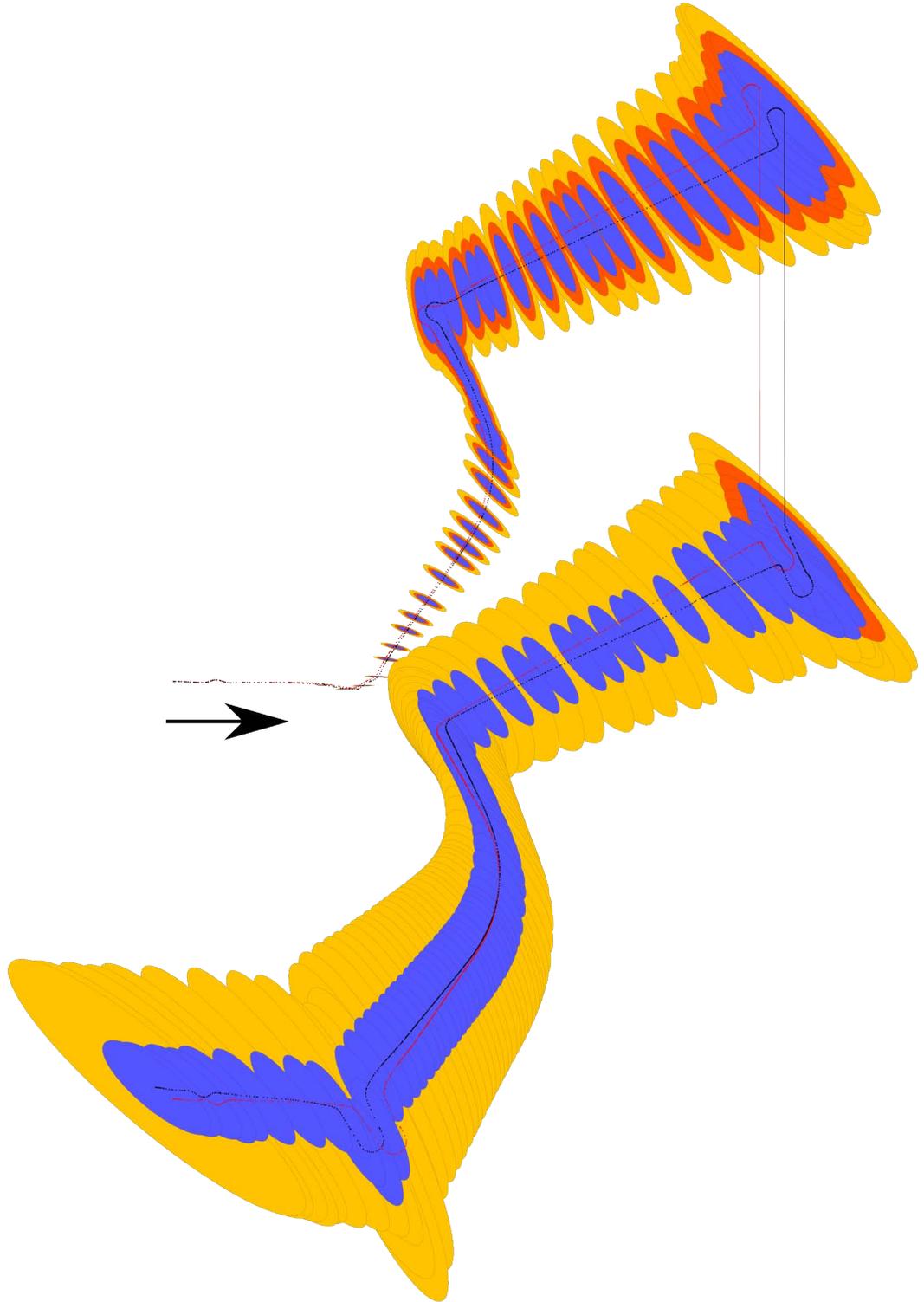
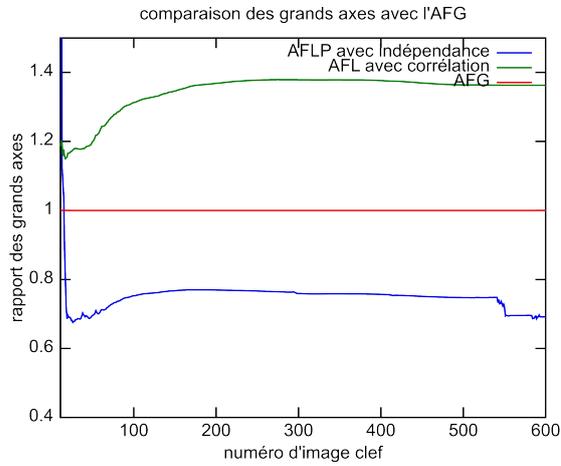
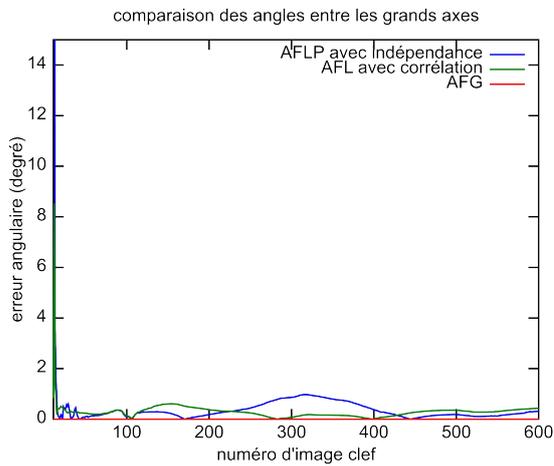


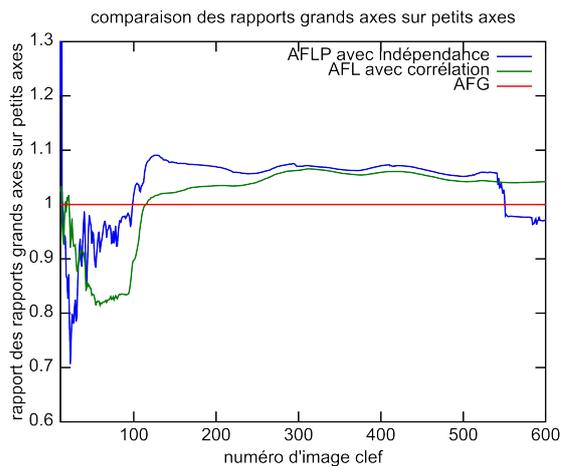
FIGURE 6.12: Vue de dessus de la séquence corrigée grâce à l'utilisation d'un odomètre (§8.3) présentant les ellipses d'incertitude pour les différentes méthodes (AFG en orange, AFLP en mauve, AFL en jaune). Cette séquence a été coupée en deux pour faciliter la lisibilité.



	med	std	min	max
AFLP	0.757	0.1245	0.6755	2.588
AFL	1.365	0.07928	0	1.379



	med	std	min	max
AFLP	0.2465	4.226	0.00300	88.8014
AFL	0.283	0.664	0.0020	13.473



	med	std	min	max
AFLP	1.063	0.0855	0.7061	2.517
AFL	1.042	0.0689	0.814	1.135

FIGURE 6.13: Comparaison avec la covariance globale séquence de réels corrigés par odomètre §8.3.

6.3.6 Temps de calcul

Des statistiques ont été effectuées sur les temps de calcul des différentes méthodes. Ici, nous présentons les temps de calcul moyens pour estimer l'incertitude avec chacune des méthodes implémentées (la méthode AFL avec corrélation n'est pas optimisée) pour une trajectoire 384 images clefs.

Pour la méthode utilisant l'AFG le temps de calcul global a été ramené à une caméra (division du temps par le nombre de caméras) dans un but de comparaison.

Pour une itération, le temps de calcul moyen est de 50s pour l'AFL avec corrélation, 50ms pour l'AFG ramené à une caméra et 6.5 ms pour l'AFLP avec indépendance. Le temps cumulé pour chaque itération de l'AFLP est de 2,4s comparé à celui de l'AFG qui est de 145s.

De plus, il est intéressant de noter que la propagation d'incertitude est en temps presque constant (elle varie suivant le nombre de points 3D observés) pour l'AFLP contrairement à l'AFG qui varie en $O(n^3)$ avec la longueur de la trajectoire (n étant le nombre de caméras depuis le début de la trajectoire).

6.3.7 Conclusion

Dans cette section, nous avons montré que la covariance estimée par nos différentes méthodes est bien propagée au cours des itérations. En comparant les résultats obtenus avec ceux donnés par un ajustement de faisceaux global sur l'ensemble de la trajectoire, nous avons pu déterminer les caractéristiques des deux méthodes étudiées. La méthode issue de l'ajustement de faisceaux local produit une covariance réaliste sur toutes les séquences. Elle est souvent légèrement supérieure à celle estimée par l'AFG mais cela est un résultat attendu du fait de la nature incrémentale du processus d'estimation. La méthode issue de l'AFLP n'estime pas correctement la covariance de l'AFL. Par contre, le rapport des longueurs des grands axes des ellipses reste à peu près constant au cours du temps. Cette propriété sera utilisée en pratique (§6.6). Enfin, nous rappelons que la méthode de calcul de covariance issue de l'AFL n'est pas utilisable en temps réel contrairement à celle de l'AFLP.

6.4 Monte-Carlo

Le but de cette section va être d'estimer la covariance par des méthodes statistiques de manière à valider la version analytique élaborée précédemment et vérifier les résultats précédents provenant de la comparaison avec l'AFG. Pour cela, nous allons utiliser la méthode de Monte-Carlo décrite en section 1.3.2.

6.4.1 Méthodologie pour le test d'une application de l'AFL

Le but de ces expériences est de montrer que la covariance est correctement propagée par une application de l'ajustement de faisceaux local.

On se place à l'instant t . La covariance a été évaluée à tous les instants précédents par la méthode analytique. Donc à partir de l'évaluation de la covariance faite à l'instant $t - 1$, nous connaissons la distribution des variables d'entrées. Nous allons appliquer la méthode de Monte-Carlo à cette distribution de manière à calculer la distribution des variables aux temps t .

Pour cela, nous créons un ensemble de réalisations suivant la distribution d'entrée. Nous propageons cette distribution par la fonction ajustement de faisceau. Enfin, nous effectuons les statistiques sur ces données, de manière à estimer la covariance des variables de sortie. Nous comparons cette covariance estimée avec celle estimée par la méthode analytique précédemment dérivée.

6.4.2 Présentation des résultats

Cette expérience a été effectuée en utilisant la séquence de simulation en virage avec un bruit 2D de $\sigma^2 = 0.5$ à $t = 50$.

Résultat pour l'AFL Les résultats suivants sont le fruit de 400 000 tirages aléatoires. L'espace des paramètres d'entrée est déterminé par 348 paramètres corrélés (les caméras fixes c_f et les points 2D déjà observés) et 292 paramètres indépendants (les nouveaux points 2D observés).

Tout d'abord, nous vérifions que le tirage aléatoire est à l'image de la distribution d'entrée. Pour cela, nous comparons le rapport des coefficients de covariance entre la covariance analytique et la covariance statistique calculée sur la distribution échantillonnée pour la position des 7 caméras fixes (c_f) soit seulement 21 paramètres sur les 600.

La table 6.2 présente les statistiques des rapports des coefficients de ces covariances.

	Moyenne	Médiane	Ecart-Type
Covariances	0.9522	0.9931	0.0754
Variances	0.9987	0.9965	0.0060

TABLE 6.2: Statistiques sur la distribution d'entrée (rapport des coefficients des covariances).

Nous voyons que les coefficients de covariances par Monte-Carlo sont en moyenne (et médiane) similaires à ceux des covariances analytiques.

La figure 6.14 présente le résultat de la propagation d'erreur par la méthode de Monte-Carlo. Elle présente en projection les différents éléments, c'est-à-dire, les positions (x,y,z) de la dernière caméra de l'AFL pour chaque tirage effectué, l'ellipse d'incertitude de la covariance estimée analytiquement et l'ellipse d'incertitude de la méthode de Monte-Carlo estimée numériquement. Les statistiques associées sont présentées table 6.4 et la table 6.3 présente simplement le rapport des coefficients des deux covariances.

	Rapport de la longueur des axes	Erreur en degré
Grand axe	0.9991	0.0296
Moyen axe	1.0050	0.154
Petit axe	1.0045	0.153

TABLE 6.4: Comparaison de l'ellipsoïde (correspondant à la position (x,y,z) de la dernière caméra) estimée analytiquement par rapport à l'estimation numérique fournie par Monte-Carlo.

Ces résultats confirment les résultats obtenus par les premiers test : le grand axe est parfaitement restitué (tant en longueur quand orientation). Les autres axes sont légèrement surestimés et une légère erreur angulaire est présente. On constate ici les très bonnes performances de la méthode AFL avec corrélation.

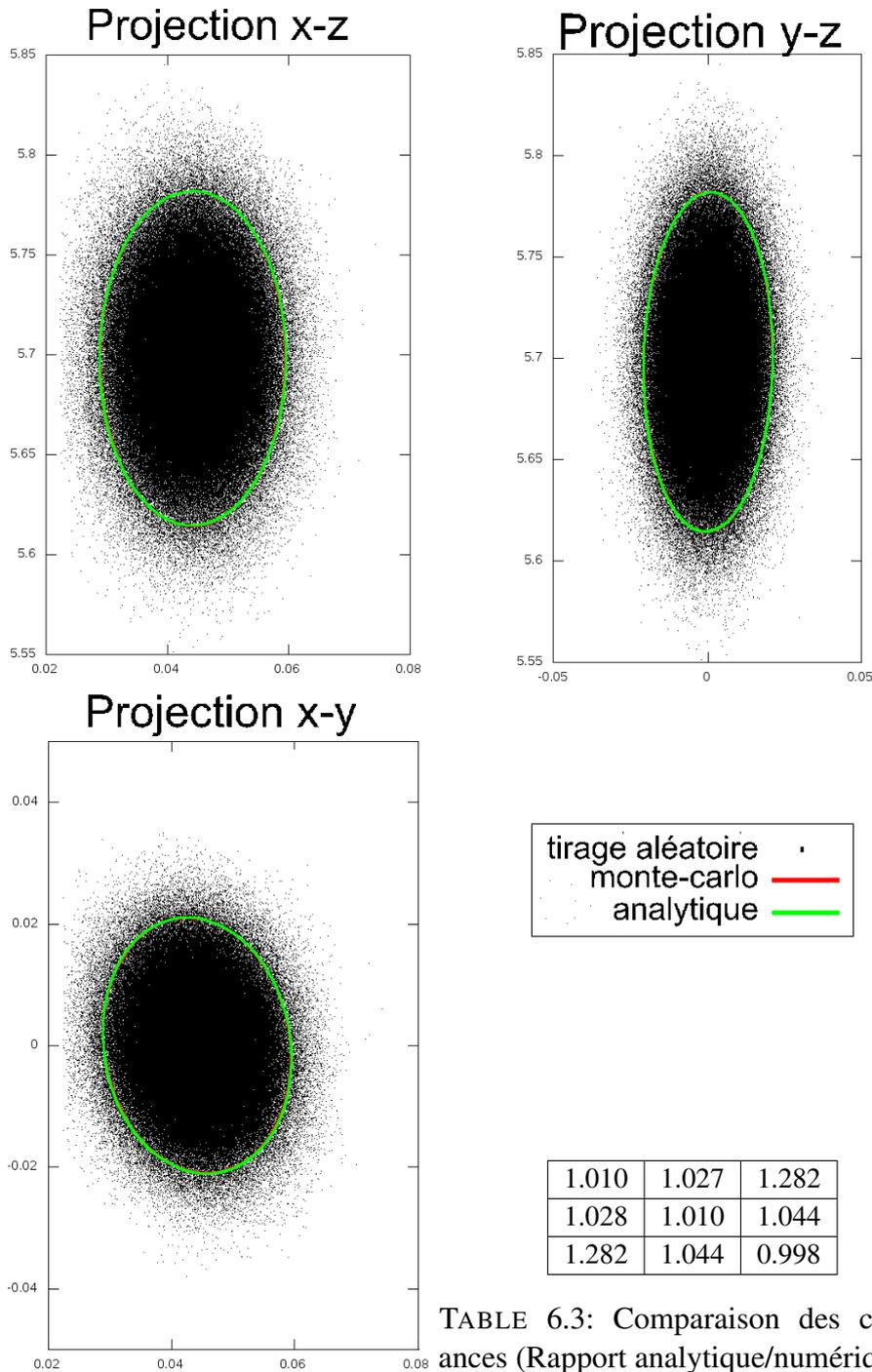


TABLE 6.3: Comparaison des covariances (Rapport analytique/numérique).

FIGURE 6.14: Cette figure présente pour la position de la nouvelle caméra : les différents positions résultats du tirage aléatoire et les ellipses d'incertitude estimées numériquement et analytiquement. Elle présente également le rapport des coefficients de la matrice de covariance 3×3 estimés analytiquement sur ceux de la matrice de covariance estimés numériquement.

Résultat pour l'AFLP Dans le cas de l'AFLP, l'espace des paramètres d'entrée est différent : nous avons 42 paramètres corrélés (les paramètres de caméras c_c) et 598 paramètres indépendants (ici les points 2D).

De la même manière que précédemment, nous présentons, dans la table 6.5, la comparaison de la distribution d'entrée avec le tirage aléatoire obtenu. Comme précédemment, les variances sont très bien respectées. On remarque cette fois-ci que plusieurs coefficients de covariance ne sont pas exacts.

La figure 6.15 et la table 6.6 montrent le résultat de la simulation et la comparaison entre la covariance analytique et numérique de la sortie.

	Moyenne	Médiane	Ecart-Type
Covariances	0.968	1.002	0.7059
Variances	1.001	1.0003	0.0017

TABLE 6.5: Statistiques sur les erreurs de distribution d'entrée (rapport des coefficients des covariances).

	Rapport des longueurs des axes	Erreur en degré
Grand axe	1.00014	0.011
Moyen axe	0.97159	0.720
Petit axe	0.97371	0.720

TABLE 6.6: Statistiques sur les erreurs de l'ellipsoïde correspondant à la position (x,y,z) de la dernière caméra estimée analytiquement par rapport à l'estimation numérique.

Ici, nous nous rendons compte que la propagation analytique estime correctement le grand axe. Par contre, l'estimation des deux autres axes est plus mauvaise, une erreur de 3% et presque un degré d'erreur en orientation est observé.

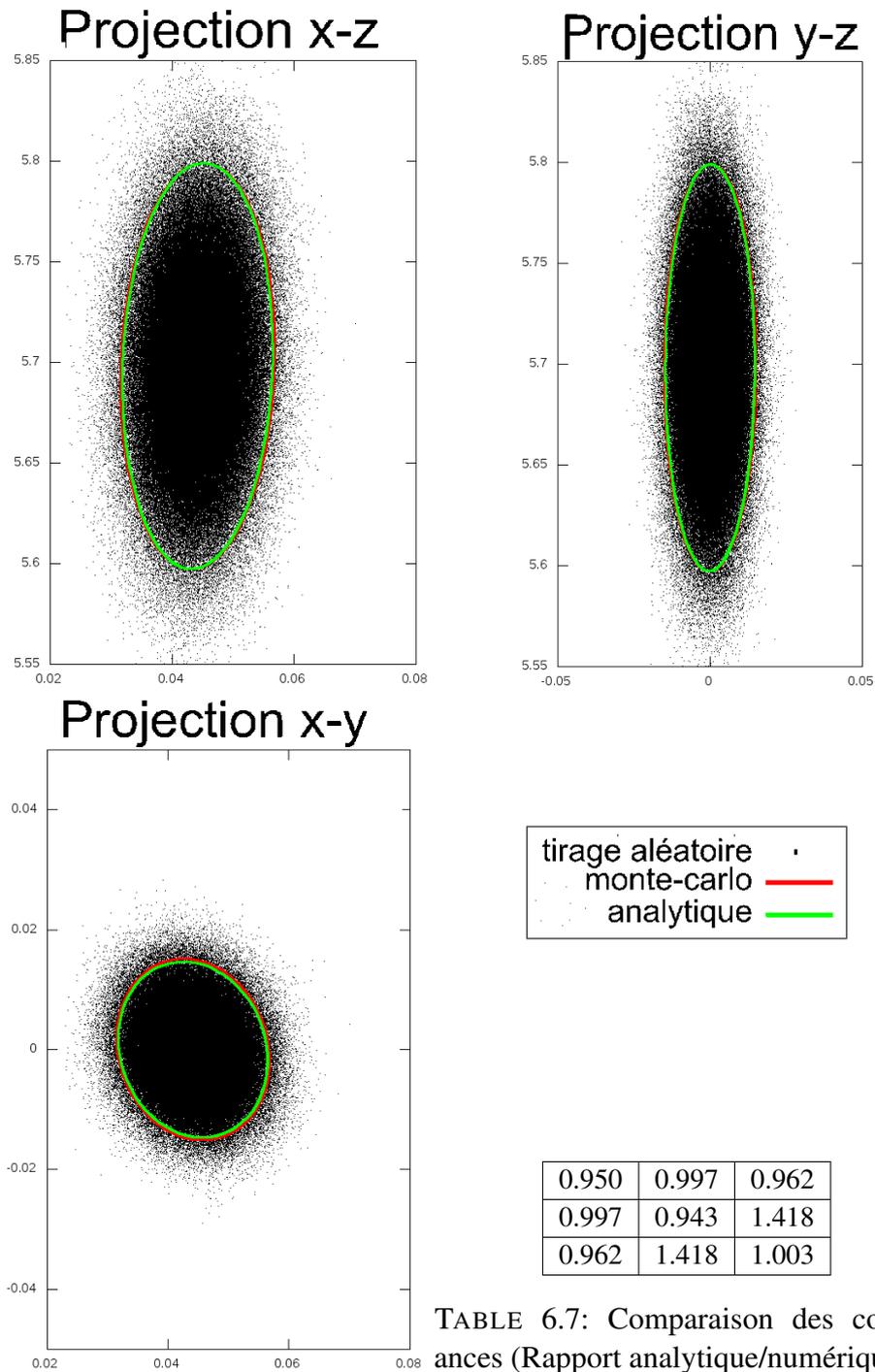


TABLE 6.7: Comparaison des covariances (Rapport analytique/numérique).

FIGURE 6.15: Cette figure présente pour la position de la nouvelle caméra : les différentes positions résultats du tirage aléatoire et les ellipses d'incertitude estimées numériquement et analytiquement. Elle présente également le rapport des coefficients de la matrice de covariance 3×3 estimés analytiquement sur ceux de la matrice de covariance estimés numériquement.

6.4.3 Analyses des résultats

Les expériences effectuées précédemment par la méthode de Monte-Carlo sur les deux méthodes étudiées montrent que la propagation est effectuée d'une application de l'ajustement de faisceaux local à l'autre. Pour la méthode AFL, les résultats de la méthode analytique sont très proches des simulations numériques. Pour la méthode AFLP, les résultats montrent des erreurs plus importantes quant à la propagation.

6.4.4 Complément d'analyse

Les tests de Monte-Carlo permettent de vérifier la bonne propagation de la covariance pour une seule application de l'AFL. Cette procédure est lourde et nécessite de nombreux tirages aléatoires, car notre espace de représentation est grand. En outre, cette procédure n'est pas résistante aux outliers. En effet, l'estimation de la covariance de la distribution de sortie peut être complètement faussée par un tirage malheureux. De plus, la covariance est estimée sur une trajectoire particulière et est comparée à la distribution. Pour montrer la pertinence du calcul de covariance proposé, il serait nécessaire de tester la propagation en chaque point de la trajectoire. Dans la section suivante, nous présentons une autre manière de vérifier que la covariance estimée est conforme à l'erreur réellement existante. Cette méthode permet en plus de tester les résultats sur toute une trajectoire et non plus sur une seule itération comme ici.

6.5 Test de Consistance

6.5.1 Définition

Un estimateur $\hat{\mathbf{x}}$ est dit consistant, si il répond à deux critères :

1. L'estimateur est non biaisé, i.e $E(\hat{\mathbf{x}}) - \bar{\mathbf{x}} = 0$.
2. La variance estimée ($\hat{C}_{\mathbf{x}}$) est supérieure ou égale à la variance réelle ($C_{\mathbf{x}}$), i.e $\hat{C}_{\mathbf{x}} - C_{\mathbf{x}}$ est positive.

Il existe plusieurs tests pour vérifier cette propriété [5]. Ils reposent tous sur le calcul de l'erreur entre la valeur estimée et la valeur réelle. Ici, nous allons utiliser la même méthode que [4] qui a été utilisée pour mettre en lumière les problèmes de consistance de l'EKF. Pour cela, il est nécessaire de connaître avec précision la valeur réelle. Ici, on ne peut pas se contenter d'utiliser une vérité terrain comme le trajectomètre ou un GPS car ces mesures sont bruitées et vont biaiser les résultats des tests. Ces tests sont donc effectués sur des données simulées.

Pour tester la consistance, nous allons utiliser un test statistique utilisant l'erreur NEES (Normalised Estimation Error Square) erreur normalisée d'estimation au carrée, définie par :

$$\epsilon(t) = (\mathbf{x}_t - \bar{\mathbf{x}}_t)^T \hat{C}_{\mathbf{x}_t}^{-1} (\mathbf{x}_t - \bar{\mathbf{x}}_t) \quad (6.2)$$

avec, dans notre cas, t le temps et \mathbf{x}_t la position à l'instant t . Cette distance va refléter les deux critères d'un estimateur consistant.

Dans le cas d'un estimateur linéaire et gaussien consistant \mathbf{x}_t , l'erreur NEES suit une loi de χ^2 à $|\mathbf{x}|$ degrés de liberté.

Nous effectuerons le test suivant :

$$\forall t, \epsilon(t) < \chi_{|\mathbf{x}|, \alpha}^2 \quad (6.3)$$

avec $\chi_{|\mathbf{x}|, \alpha}^2$, le fractile d'ordre α pour une distribution de χ^2 à $|\mathbf{x}|$ degrés de liberté. Si ce test est positif, l'estimateur sera déclaré consistant c'est-à-dire que l'estimateur reflète bien ses propres performances. Dans le cas contraire, il est déclaré inconsistant.

Si nous utilisons une seule trajectoire pour effectuer ce test, les données sont temporellement corrélées, puisque l'estimation courante est effectuée à partir de l'instant précédent. Le risque est de trouver une trajectoire qui montre qu'un estimateur consistant ne le soit pas ou qu'un estimateur inconsistant le soit. Pour palier ce problème, la méthode habituellement utilisée ([5] page 234) consiste à

effectuer une simulation de Monte-Carlo pour obtenir plusieurs tirages de \mathbf{x} indépendants. Ainsi, nous utiliserons plutôt :

$$\bar{\epsilon}(t) = \frac{1}{N} \sum_{i=1}^N \epsilon_i(t) \quad (6.4)$$

Et donc, on effectuera le test suivant :

$$\bar{\epsilon}(t) < \frac{1}{N} \chi^2_{N \times n, \alpha} \quad (6.5)$$

avec $n = |\mathbf{x}|$ et α le seuil d'acceptation de l'hypothèse.

6.5.2 Résultat

Les tests ont été effectués sur les deux séquences de simulation présentées précédemment. Pour chaque expérience, la trajectoire de simulation est artificiellement bruitée. Pour cela, nous ajoutons sur chaque point 2D un bruit gaussien centré de variance $\sigma^2 = 0.5$. Puis, nous optimisons la trajectoire, en utilisant l'AFL. La covariance correspondant à cette trajectoire bruitée est calculée pour les deux méthodes (AFL et AFLP).

Ici, nous utilisons 50 expériences pour nous assurer de la qualité du résultat (la figure 6.16 présente une vue de dessus des 50 trajectoires en virage résultant du tirage aléatoire). Pour chaque trajectoire i , à chaque instant t , la distance NEES $\epsilon_i(t)$ est calculée à partir de l'estimation de la covariance $C_{\mathbf{x}_t}$ et de la position \mathbf{x}_t de la dernière caméra de l'ajustement de faisceaux local à t obtenues par la méthode analytique sur la i -ème trajectoire. Puis la distances NEES moyenne $\bar{\epsilon}(t)$ est la moyenne des $\epsilon_i(t)$. C'est l'évolution de ce critère en fonction du temps qui sera représenté pour chacun des essais suivants et comparé par rapport au seuil du χ^2 . Le seuil du χ^2 sera représenté en rouge sur les figures suivantes. Le seuil d'acceptation a été fixé à 97.5%, $N = 50$ et $|\mathbf{x}| = 3$. La valeur du seuil sera donc de 3,72.

Résultat pour l'AFL avec corrélation Sur les deux séquences de test, nous voyons que la méthode est consistante (voir figure 6.17). Cela confirme les résultats des deux précédents essais.

AFLP avec indépendance Pour l'AFLP, sur la figure 6.18, la méthode n'est pas consistante sur les deux séquences de tests.

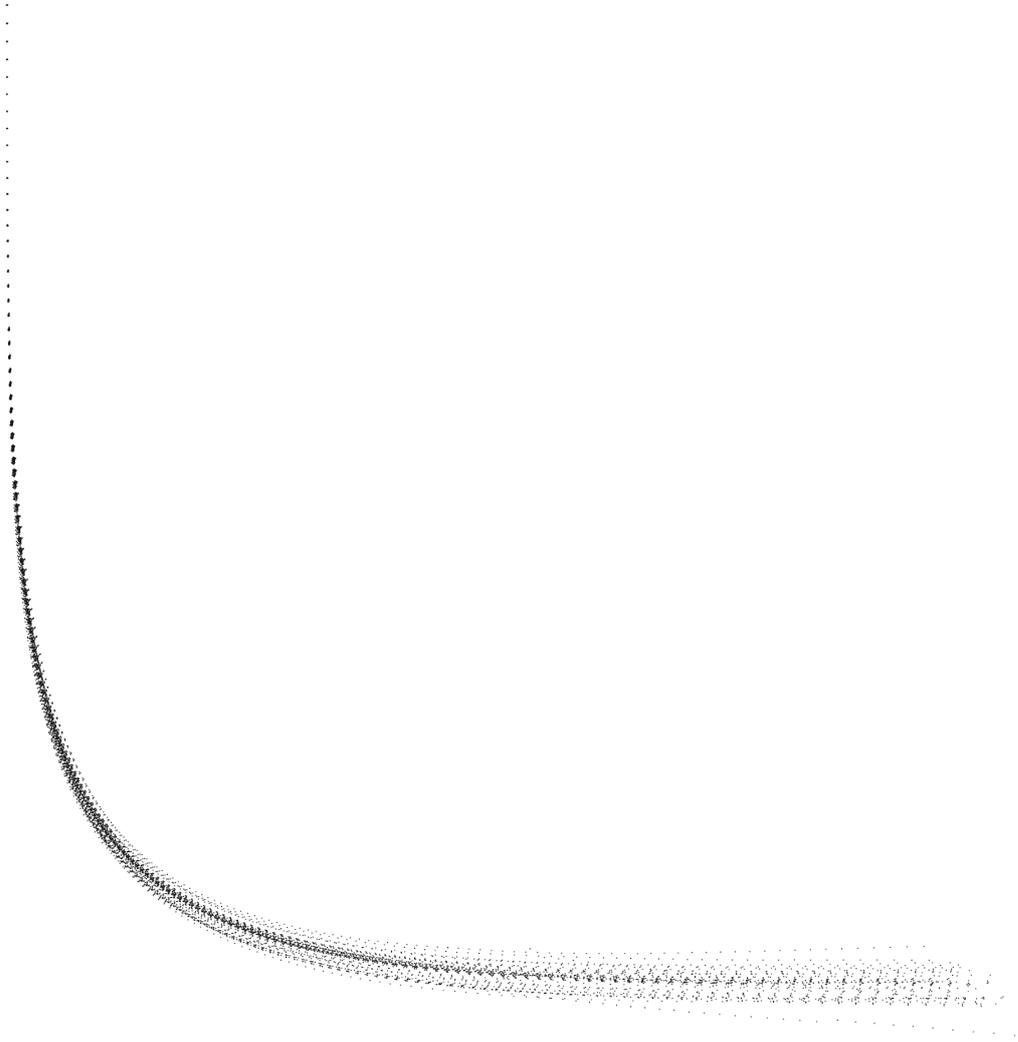


FIGURE 6.16: Cette figure présente les 50 trajectoires bruitées pour le calcul de consistance sur la séquence en virage.

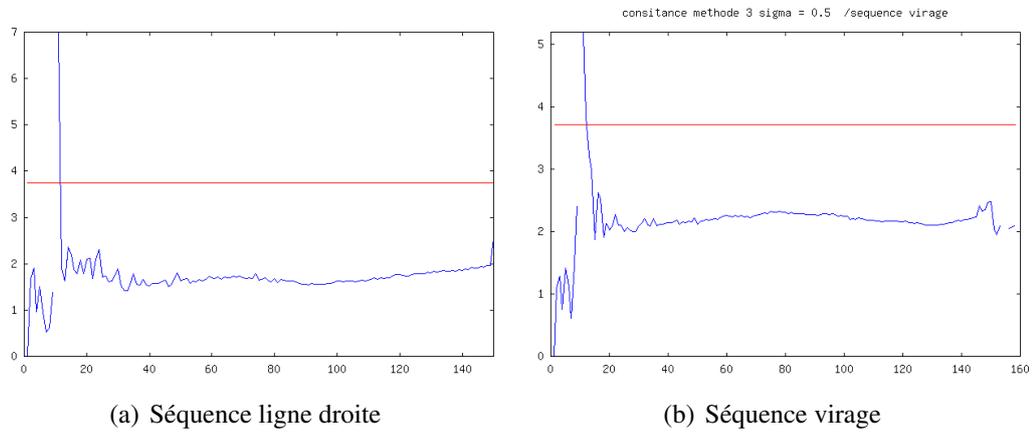


FIGURE 6.17: Consistance de l'AFL avec corrélation. Les deux figures représentent la distance NEES en fonction du temps en images clef. Sur la séquence en ligne droite (a) et sur la séquence en virage (b).

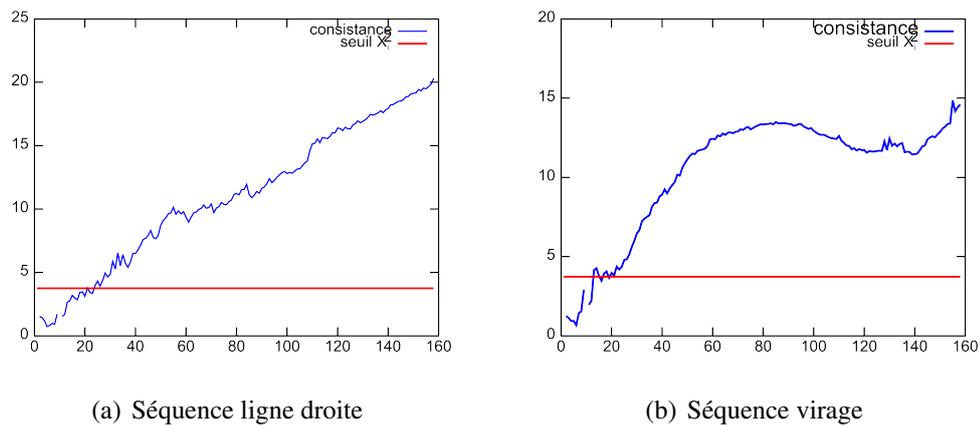


FIGURE 6.18: Consistance de l'AFLP avec indépendance. Les deux figures représentent la distance NEES en fonction du temps en images clefs. Sur la séquence en ligne droite (b) et sur la séquence en virage (a).

6.6 Correction de la méthode AFLP

Dans les résultats précédents, nous avons montré que la méthode AFLP propage une incertitude au cours du temps. Malheureusement, cette incertitude est sous-estimée par rapport à la réalité. Nous avons mis en lumière le fait que malgré cette sous-estimation, la propagation reste cohérente à un facteur près. De ce fait, nous proposons d'introduire un facteur correctif permettant de tout de même utiliser les covariances fournies par l'AFLP pour estimer l'erreur effectuée par l'AFL. Ce facteur a été estimé en fonction du rapport entre les covariances de la méthode AFL et celle de la méthode AFLP. Une expérimentation a été conduite sur plusieurs séquences. Les valeurs varient entre 1.53 et 2.2 autour d'une moyenne de 1.87 avec un écart-type de 0.20. Nous proposons donc pour pouvoir tout de même utiliser la méthode AFLP de multiplier les résultats par ce facteur correctif de manière à obtenir des valeurs proches de celle de l'AFL. Nous appliquerons donc un facteur correctif de 1.87. La figure 6.19 montre une vue de dessus des ellipses de l'AFLP corrigée et de l'AFL.

6.7 Conclusion

Dans ce chapitre, nous avons estimé les performances des deux méthodes retenues pour propager l'incertitude dans l'ajustement de faisceaux local. La première, l'AFL avec prise en compte des corrélations, propage l'incertitude de manière correcte sur toutes les séquences que nous avons utilisées. Malheureusement, son coût en temps de calcul est trop important pour une application embarquée. La seconde méthode proposée est l'AFLP avec hypothèse d'indépendance. Même si cette hypothèse est en toute rigueur très simplificatrice, la méthode reste utilisable et permet de propager une incertitude au cours des itérations. Nous avons vu que cette méthode sous-estime la covariance, mais cette sous-estimation a la propriété de rester proportionnelle au cours de la séquence. Nous avons donc proposé d'utiliser un facteur correctif pour rendre cette méthode utilisable et pouvoir propager l'incertitude en temps réel.

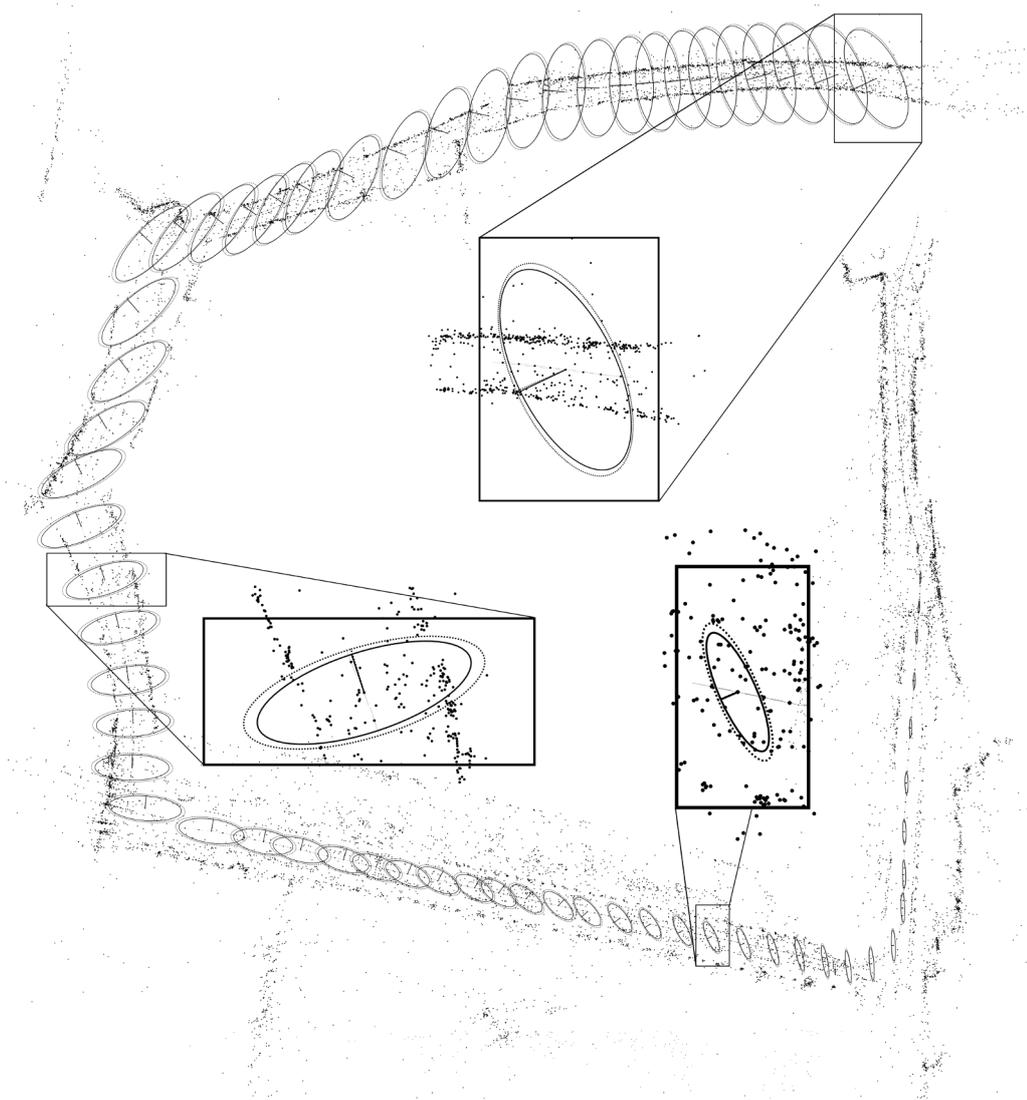


FIGURE 6.19: Cette figure présente une représentation en vue de dessus des ellipses d'incertitude calculée par deux méthodes : la méthode AFG (trait plein) et la méthode AFLP corrigée (pointillés).

Chapitre 7

Application au SLAM Visuel

Sommaire

7.1	SLAM Visuel	110
7.1.1	Schéma général	110
7.1.2	Localisation rapide	111
7.1.3	Choix de l'image clef	113
7.1.4	Mise à jour du nuage de points 3D	114
7.1.5	Initialisation	115
7.2	Utilisation de l'ajustement de faisceau local pondéré	116
7.2.1	Ajustement de faisceaux local pondéré	116
7.2.2	Evaluation expérimentale	117
7.2.3	Résultats sur la séquence de synthèse	118
7.2.4	Résultat sur une séquence réelle	122
7.3	Conclusion	125

Ce chapitre détaille, dans un premier temps, la méthode de SLAM visuel développée lors de la thèse d'Etienne Mouragnon [79] qui sert de base à nos travaux. Dans un second temps, nous présentons l'utilisation de différents ajustements de faisceaux en remplacement de celui utilisé à l'origine. Ces travaux ont été publiés dans [37].

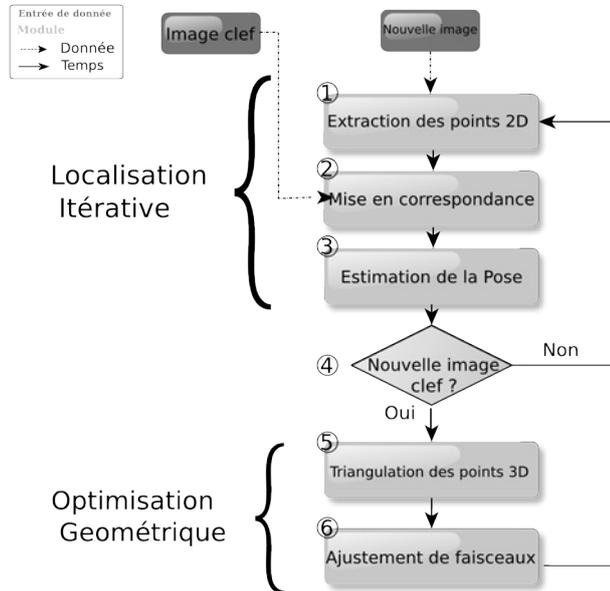


FIGURE 7.1: Schéma général de l'algorithme utilisé.

7.1 SLAM Visuel

Cette section a pour but d'expliquer le fonctionnement du SLAM visuel utilisé ici et qui a été développé durant la thèse d'Etienne Mouragnon.

7.1.1 Schéma général

Le schéma général de la méthode est présenté dans la Fig. 7.1.

Cette approche peut être découpée en deux phases. La première étape consiste en une localisation rapide pour chaque image. Ici, le but est de recalculer rapidement la pose correspondante à la nouvelle image. Pour cela, un suivi de points entre les images est nécessaire. Ensuite, un calcul de pose peut être effectué pour déterminer la position de la caméra. Chacune de ces étapes est décrite en section 7.1.2

Une fois cette première étape terminée, il y a deux possibilités. Soit la pose calculée à une qualité suffisante, alors l'étape précédente est répétée. Soit, la qualité de la pose est jugée insuffisante et alors la deuxième partie de l'algorithme est utilisée. Il met en oeuvre la notion d'image clef ; ces images clefs forment une sous-séquence de la séquence d'origine. Ce choix ainsi que la notion d'image clef

sont détaillés section 7.1.3.

La seconde partie de l'algorithme décrite en section 7.1.4 a pour but de mettre à jour le nuage de point 3D qui sert à la localisation. Pour cela, deux étapes sont utilisées : d'abord une étape d'extension du nuage de point 3D qui utilise la triangulation. Puis une étape d'optimisation pour réduire les erreurs introduites par les calculs précédents.

7.1.2 Localisation rapide

La première phase de l'algorithme a pour but de déterminer la pose qui correspond à la dernière image acquise. Pour ce faire, il est nécessaire de trouver des correspondances entre des points 3D du nuage et leurs observations dans l'image.

Points d'intérêt Le but de cette étape est de trouver des points particuliers de l'environnement qui seront facilement reconnaissables même en cas de changement de point de vue. Ils sont appelés points d'intérêt et seront détectés directement dans l'image. Ici, le détecteur de Harris [47] a été choisi pour ses très bonnes performances et pour son faible coût en temps de calcul. Nous n'utilisons pas les détecteurs plus récents invariants à l'échelle ou aux changements affines [75], car leurs coûts en temps de calcul est trop important.

Descripteur La seconde étape consiste à construire une signature caractéristique du point dans le but de pouvoir reconnaître un point parmi les autres. Pour cela, un voisinage autour du point est utilisé. Énormément de descripteurs ont été proposés et ce domaine de recherche reste très actif. Dans les travaux initiaux d'Etienne Mouragnon, le descripteur utilisé était la ZNCC (Zero-mean normalized cross-correlation). Cette technique consiste à utiliser directement l'image pour construire le descripteur.

Mon sujet de stage de fin d'études de master effectué au CEA avait pour but l'adaptation de la méthode dans un cadre automobile en milieu urbain. La méthode avait auparavant été testée à petite vitesse (15 km/h maximum). Une vitesse plus importante (jusqu'à 50km/h) entraîne principalement deux problèmes : une distance plus importante entre deux images et une augmentation du flou de bougé.

La mise en correspondance devenant plus difficile, la ZNCC [16] n'était plus suffisante. Pour pallier ce problème, d'autres descripteurs ont été testés (SIFT [73], SURF [6]).

Le descripteur finalement choisi est un SURF [6] modifié de manière à fonc-

tionner à une seule échelle et ne prenant pas en compte l'invariance à la rotation. Cette simplification a été effectuée pour deux raisons. La première pour des raisons dues à l'application : dans le cadre automobile, il n'y a pas de rotation importante dans les images et les changements de point de vue introduits ne sont pas suffisants pour entraîner des changements d'échelle importants. De plus, l'utilisation du multiéchelle ou de l'invariance à la rotation rend les descripteurs moins discriminants. La seconde raison est la volonté de réduire le coût en temps de calcul de l'utilisation de SURF à la place de la ZNCC de manière à pouvoir traiter le flux vidéo en temps réel. Pour diminuer encore ce coût, des stratégies de parallélisation du code et de portage sur GPU ont été mises en place. J'ai continué ces travaux durant ma thèse notamment dans le cadre du projet ODIAAC.

Mise en correspondance La mise en correspondance des points détectés dans les deux images repose sur le fait de chercher le point de l'image 2 qui ressemble le plus à celui de l'image 1, la ressemblance étant calculée grâce à une distance entre descripteurs.

Pour cela, il existe plusieurs techniques suivant les connaissances a priori sur le mouvement, ou les hypothèses effectuées. La plus simple consiste à effectuer une recherche exhaustive dans l'image 2 et à garder le point qui a le meilleur score de ressemblance pour chaque point de l'image 1. Pour éviter de conserver des points qui auront un trop mauvais score de ressemblance, un seuil fixe le score au-dessus duquel l'appariement n'est pas conservé. De même, pour éviter les problèmes dus aux structures répétitives, un deuxième seuil vérifie que la différence de ressemblance entre les deux meilleurs appariements possibles est suffisante (supérieure à un seuil). Construire la carte de distance entre tous les points de la première image et ceux de la seconde est très coûteux en temps de calcul. Pour réduire ce coût, le mouvement de la caméra est supposé avoir une vitesse bornée. Cela nous permet de limiter la recherche dans une fenêtre en pixel centrée autour de la position du point de l'image 1.

En sortie de cette étape, des appariements 2D/2D entre deux images sont obtenus.

Localisation 3D

La pose de la caméra courante est maintenant recherchée. Pour cela, il faut tout d'abord déterminer des correspondances entre des points 3D existants et leurs observations dans l'image courante.

Pour chaque association 2D/2D entre l'image courante et l'image clef précé-

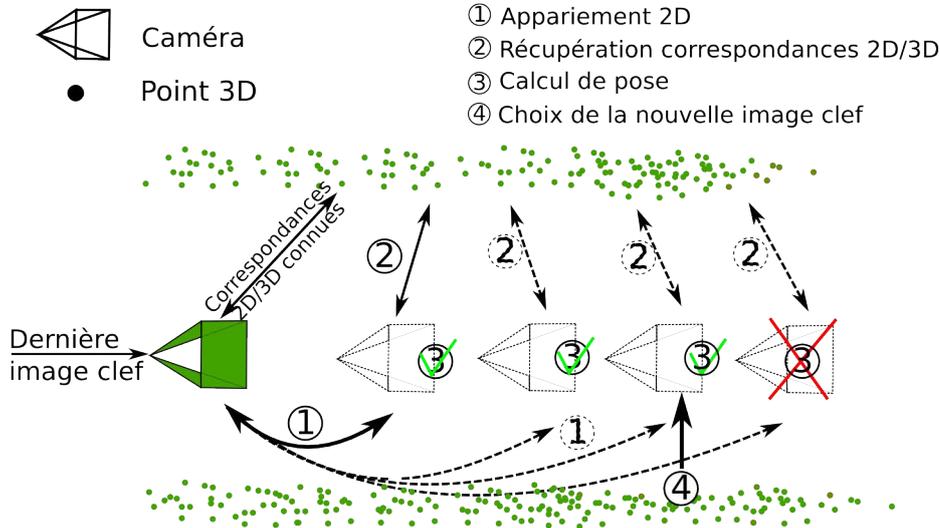


FIGURE 7.2: Phase de localisation.

dente, seules les correspondances 2D qui sont associées dans l'image précédente à un point 3D déjà triangulé sont conservées. À partir des associations 3D/2D ainsi obtenues, la pose courante de la caméra peut être calculée en utilisant l'algorithme des 3 points (§1.1.3). Pour obtenir un résultat robuste à de mauvais appariements, l'algorithme de RANSAC est utilisé (§1.1.3).

7.1.3 Choix de l'image clef

Cette étape permet d'évaluer la qualité de la localisation de la caméra qui vient d'être effectuée. Pour cela, deux critères sont utilisés. Le premier critère est purement 2D, il concerne le nombre d'appariements entre l'image courante et la dernière image clef. Si ce nombre descend en dessous d'un certain seuil, cela signifie que les deux images sont trop éloignées l'une de l'autre et le processus de mise en correspondance est mis en défaut. Le second critère est 3D, il vérifie le nombre de points 3D correctement reprojétés dans l'image courante (la projection du point 3D doit être proche de son observation 2D). Si ce nombre descend en dessous d'un certain seuil, la qualité de la pose reconstruite par l'algorithme des 3 points ne peut plus être garantie. Ces deux critères doivent être validés pour déclarer la pose courante de qualité suffisante.

Ensuite, suivant la qualité de la pose, deux choix sont possibles.

Soit, la pose est de bonne qualité. Dans ce cas, le nuage de point 3D est consid-

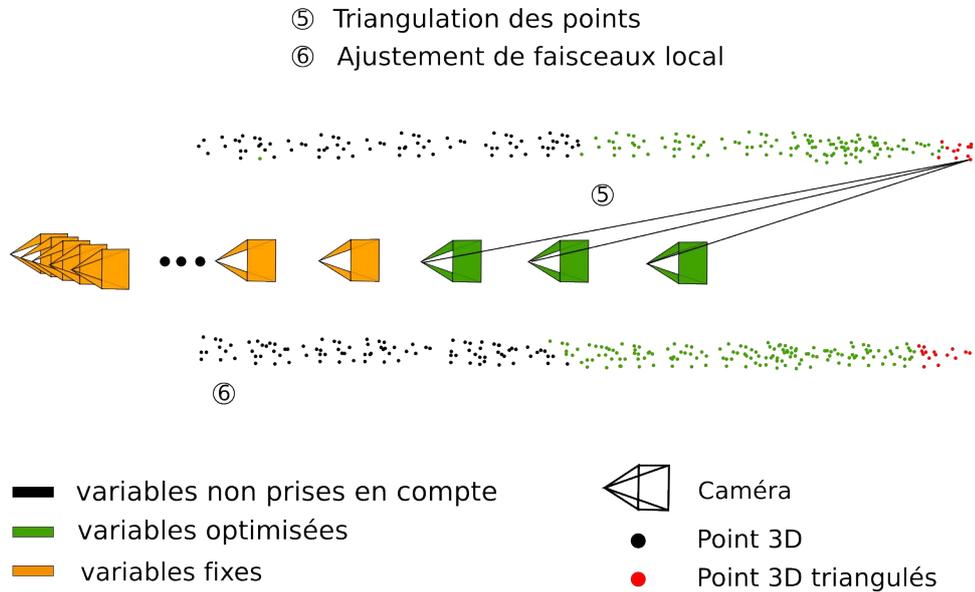


FIGURE 7.3: Reconstruction 3D et optimisation géométrique.

éré comme suffisant pour se localiser. L'algorithme se poursuit, en répétant l'étape de localisation avec l'image suivante (étape 1, fig.7.1).

Soit, dans le cas contraire, la qualité de la pose courante est jugée insuffisante. Donc cette pose est oubliée et la pose précédente est considérée comme une nouvelle image clef. Ce fait signifie que le nuage de point 3D n'est plus suffisant pour se localiser. Une procédure de mise à jour du nuage de point, décrite dans la section suivante, est alors utilisée. Une fois le nuage mis à jour (étape 5/6), il sera de nouveau possible d'utiliser la procédure de localisation (étape 1).

7.1.4 Mise à jour du nuage de points 3D

Cette seconde partie de l'algorithme a pour but de mettre à jour le nuage de point 3D pour pouvoir continuer le processus de localisation dans de bonnes conditions. Elle se découpe en deux étapes : la mise à jour proprement dite (étape 4), puis une étape d'optimisation de la géométrie (étape 5).

Triangulation 3D

Depuis la dernière mise à jour, une nouvelle partie de l'environnement a été explorée : il est nécessaire de compléter le nuage de points 3D avec de nouveaux

points pour représenter cette partie.

De nouveaux points vont donc être triangulés à partir de correspondance 2D trouvée entre les 3 dernières images clefs ($t, t-1, t-2$). En effet, le fait de suivre un élément sur 3 images assure plus de robustesse et limite la construction de mauvais points.

Lors de la procédure de triangulation, la profondeur des points 3D est estimée. Pour que cette estimation soit de bonne qualité, il est nécessaire d'avoir un changement de point de vue suffisant entre les différentes caméras servant à la triangulation. L'utilisation d'image clef assure que ce changement de point de vue soit aussi grand que possible pour que les points reconstruits soient de qualité suffisante pour servir de support à la localisation.

Optimisation géométrique

Cette dernière étape a pour but de diminuer l'erreur géométrique : une procédure d'optimisation non linéaire est utilisée pour raffiner tous les nouveaux paramètres introduits. L'historique est utilisé pour apporter des contraintes sur ces nouveaux paramètres. Pour cela, un ajustement de faisceaux local tel que décrit au chap.4 est effectué.

Une fois cette étape terminée, un nuage de points à jour et un ensemble de poses optimisées sont obtenus. La procédure de localisation peut maintenant être de nouveau utilisée pour localiser les images suivantes (étape 1).

L'étape d'optimisation géométrique est la plus coûteuse en temps de calcul. Notons que l'utilisation des images clefs permet de réduire le nombre d'appels à l'ajustement de faisceaux tout en conservant une précision de localisation importante.

7.1.5 Initialisation

Le processus itératif présenté requiert une phase d'initialisation fournissant les trois premières poses de caméras et un nuage de points 3D initial. Cette initialisation est réalisée au moyen de l'algorithme des 5 points (section 1.1.3) pour le calcul des poses et de la triangulation pour obtenir les points 3D. Pour plus de détail sur la construction de cette première initialisation, le lecteur est invité à se référer à la thèse d'Etienne Mouragnon [79].

Il est important de noter que c'est lors de cette procédure d'initialisation que le repère de la reconstruction est choisi. Il est arbitrairement fixé sur la première

caméra ($R_0 = I_3$, $T_0 = 0_{3 \times 1}$). C'est aussi ici que l'échelle de la reconstruction est fixée, la distance entre la première et la troisième caméra est utilisée comme étalon de distance.

Cette reconstruction initiale est ensuite optimisée par un ajustement de faisceaux global (Chap.3). On notera que pour réduire les erreurs dues à la reconstruction initiale qui peuvent être importantes, l'optimisation géométrique dans le processus itératif pour les 7 images clefs suivantes sera réalisée par un ajustement de faisceau global et non pas local.

7.2 Utilisation de l'ajustement de faisceau local pondéré

Dans cette section, nous proposons de remplacer dans le SLAM visuel l'ajustement de faisceaux local jusqu'à présent utilisé par l'ajustement de faisceaux local pondéré (chap.5), et nous montrons l'apport de cette nouvelle approche.

7.2.1 Ajustement de faisceaux local pondéré

Dans le chapitre 5, nous avons défini un nouvel ajustement de faisceaux local dit pondéré noté AFLP. Contrairement à l'AFL, celui-ci optimise toutes les caméras présentes dans la fenêtre glissante. Il utilise une pondération pour contraindre les anciennes caméras à rester proche de leurs positions précédemment estimées. Ici, nous utiliserons l'AFLP avec hypothèse d'indépendance sous deux formes décrites ci-après.

Ajustement de faisceaux local pondéré avec covariance globale

Dans l'AFLP décrit en section 5.2, C_{c^-} définit la pondération appliquée sur les caméras contraintes. Cette pondération est dans ce cas la covariance globale. Cette covariance représente l'incertitude des poses courantes par rapport au début de la trajectoire. Elle tient compte de toute la trajectoire grâce à un calcul par récurrence et croît avec le temps de la même façon qu'une covariance calculée sur toute la trajectoire par un ajustement de faisceau global.

Ajustement de faisceaux local pondéré avec covariance locale

Cet ajustement de faisceau est le même que celui du chapitre 5 mais nous proposons ici d'utiliser comme pondération une covariance locale. Cette covari-

ance locale représentera la covariance des caméras exprimée dans le repère de la caméra la plus ancienne de la fenêtre glissante.

Pour cela, on considère un ajustement de faisceau global sur la fenêtre glissante utilisée. Cette covariance est donc calculée grâce à la technique standard d'estimation de la covariance en calculant l'inverse de l'approximation de la hessienne de la fonction d'AFG.

Pour inverser la hessienne, il est nécessaire de fixer la jauge. Nous effectuons cette opération en enlevant les lignes et les colonnes de la matrice hessienne correspondante aux paramètres fixes choisis. Ici, nous avons choisi la pose la première caméra (\mathbf{c}_0) et pour le dernier paramètre la plus grande coordonnée de la pose $\mathbf{c}_{|\mathbf{c}_c|-1}$.

Cette covariance est évaluée à l'itération précédente (sur les données à $t - 1$). Nous utilisons ensuite le même procédé que pour la covariance globale pour extraire de cette matrice l'information nécessaire pour la récurrence. Seule la covariance des poses correspondant à \mathbf{c}_c est conservée. Cette nouvelle covariance, dite locale, peut maintenant être utilisée dans l'ajustement de faisceau local pondéré pour optimiser l'itération suivante.

7.2.2 Evaluation expérimentale

Dans ce chapitre, nous présentons une évaluation des différents ajustements de faisceaux appliqués au SLAM visuel. Pour cela, les différentes trajectoires ont été comparées à une vérité terrain.

Comparaison des trajectoires

Pour recaler les trajectoires, il est nécessaire de trouver la translation, la rotation et le changement d'échelle permettant de passer du repère trajectoire au repère vérité terrain.

Les trajectoires estimées par le SLAM visuel monoculaire sont obtenues avec un facteur d'échelle arbitraire dans le repère de la première caméra. Pour les comparer à la vérité terrain, il convient de les remettre à la même échelle que la vérité terrain. Cependant, une limite bien connue du SLAM monoculaire est la dérive du facteur d'échelle dans le temps que souhaitons justement évaluer. Pour cette raison, nous n'avons pas retenu les techniques de best-fit sur les deux trajectoires qui ne permettent pas de mettre en lumière les phénomènes d'accumulation d'erreurs et de dérive. Nous avons choisi de remettre à l'échelle les trajectoires du SLAM visuel en utilisant une portion du début de la trajectoire (les 10 premières

caméras) sur laquelle l'échelle peut être considérée comme sans dérive et nous avons appliqué cette nouvelle échelle à l'ensemble de la trajectoire estimée. Pour la rotation et la translation nous avons recalé le début de la trajectoire sur le début de la vérité terrain et nous avons aligné les repères.

Critère d'évaluation

Pour comparer les trajectoires, plusieurs erreurs sont calculées. Notons $C_t = \Delta \mathbf{c}_{t_{gt}}$ la position de la caméra à l'instant t dans le repère de la vérité terrain (gt) et GT_t la vérité terrain à la position correspondante. Ainsi, nous calculons :

- l'erreur de position : $\|C_t - GT_t\|_2$;
- l'erreur angulaire : $\arg(C_{t-1} - C_t, GT_{t-1} - GT_t)$;
- le ratio des distances inter-caméra $\frac{\|C_{t-1} - C_t\|_2}{\|GT_{t-1} - GT_t\|_2}$.

Cette dernière est une mesure relative et va nous permettre en plus de l'erreur sur la translation de mesurer la précision de l'échelle de la reconstruction. Pour chacune des erreurs, nous donnerons la médiane, l'écart type et le minimum ou le maximum.

7.2.3 Résultats sur la séquence de synthèse

La première séquence utilisée est la séquence synthèse (2900 images, 365m). Dans la figure 7.4, nous présentons les différentes trajectoires obtenues en vue de dessus :

- l'ajustement de faisceau local original (noté original en violet) ;
- l'ajustement de faisceau local pondéré avec covariance globale (noté global en noir) ;
- l'ajustement de faisceau local pondéré avec covariance locale (noté local en bleu) ;
- la vérité terrain (en rouge).

Sur cette figure, nous observons clairement l'accumulation d'erreur pour les trois trajectoires et nous nous rendons compte que l'ajustement de faisceau local pondéré avec covariance locale est moins sensible à l'accumulation d'erreur.

La figure 7.5 présente l'évolution des critères quantitatifs pour les différentes méthodes.

Elle présente les trois critères définis dans la section précédente. L'erreur angulaire et l'erreur de position sont deux critères absolus, ils traduisent de façon chiffrée ce qui est observable sur la trajectoire. Elle montre que l'erreur angulaire est légèrement plus importante pour les méthodes utilisant l'AFLP. Elle montre

aussi que l'erreur de position est similaire pour le global et l'original et meilleure dans le cas de l'AFLP avec covariance locale. Le rapport des distances inter-caméra est un critère local, il ne devrait pas montrer d'accumulation d'erreur et devrait ressembler à un bruit blanc gaussien. Ici ce n'est pas le cas, nous observons clairement une tendance. Cette tendance est le signe de la dérive du facteur d'échelle au cours du temps. Nous nous rendons compte sur cette figure que cette dérive varie linéairement avec la distance. Ici encore, nous remarquons les meilleures performances de l'AFLP avec covariance locale.

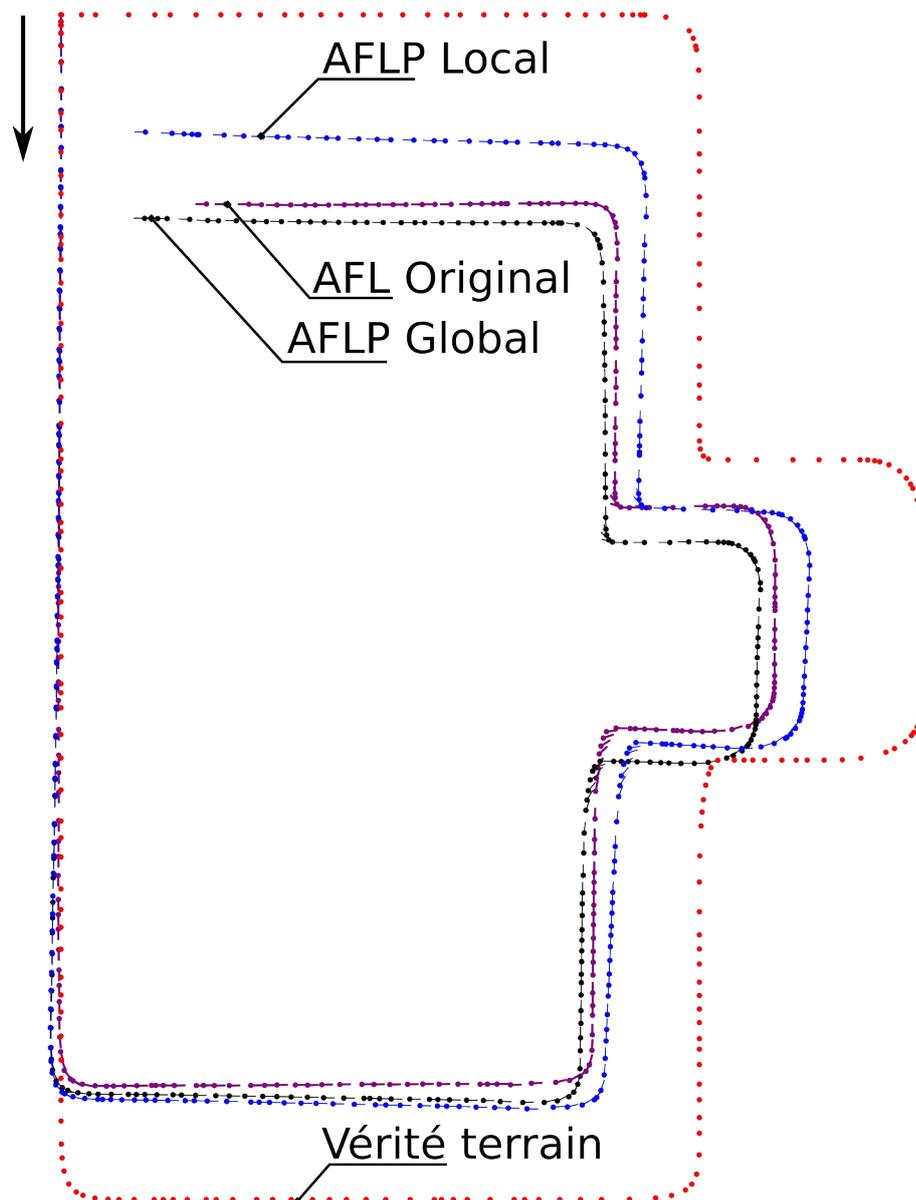
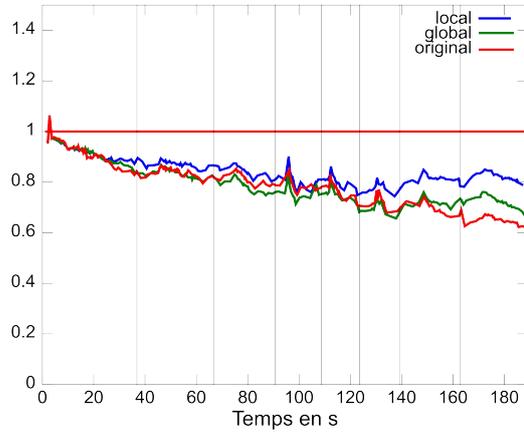
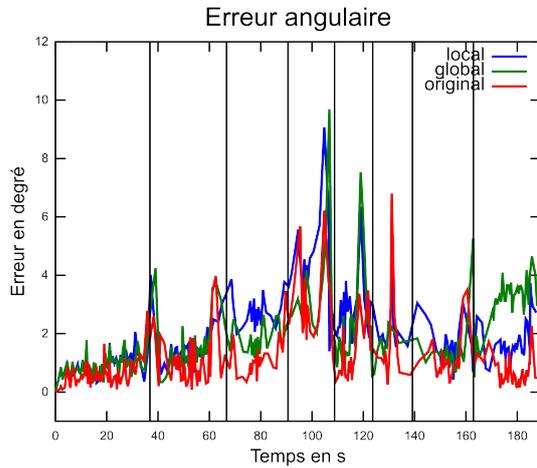


FIGURE 7.4: Vue de dessus des trajectoires reconstruites par AFL, AFLP local et global et la vérité terrain de la séquence de synthèse.

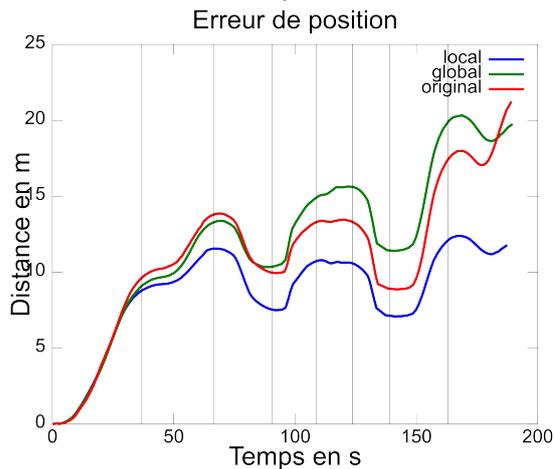
Comparaison du rapport des distances inter-caméra



	med	std	min
AFL	0.793	0.108	0.797
AFLP global	0.799	0.099	0.656
AFLP local	0.848	0.076	0.848



	med	std	max
AFL	1.104	1.013	6.801
AFLP global	1.868	1.276	9.687
AFLP local	1.906	1.220	9.074



	med	std	max
AFL	10.558	5.148	21.248
AFLP global	11.324	5.777	20.332
AFLP local	8.288	3.431	12.394

FIGURE 7.5: Evolution du rapport inter-caméra, de l'erreur angulaire et de l'erreur de position au cours du temps pour les différentes méthodes sur la séquence de synthèse.

7.2.4 Résultat sur une séquence réelle

La seconde séquence utilisée est une séquence réelle en milieu urbain en condition réelle (16000 images, 4km). Dans la figure 7.6, nous présentons les différentes reconstructions vues de dessus avec le même code couleur que précédemment.

Comme précédemment, cette figure permet de voir que le facteur d'échelle dérive au cours du temps pour les trois méthodes. La reconstruction donnée par l'AFLP avec covariance locale est sensiblement de meilleure qualité et celle obtenue par l'AFLP avec covariance globale se dégrade sur la fin de la trajectoire.

La figure 7.7 présente l'évolution des critères quantitatifs pour les différentes méthodes.

Dans cette figure, l'erreur de position après le demi-tour ($t > 300s$) n'est plus significative du fait de l'écrasement du facteur d'échelle. Cette courbe confirme l'examen visuel, l'AFLP local donne les meilleurs résultats, son erreur de position est réduite d'environ 250m au maximum de la trajectoire (à $t = 300s$). Nous pouvons remarquer que la version globale est moins stable que l'AFL et présente des erreurs importantes après la variation rapide du facteur d'échelle à $t = 125s$.

Le rapport des distances inter-caméra est intéressant sur cette figure. On retrouve la dérive linéaire due à l'accumulation d'erreur. Par contre sur le début, on observe des variations brusques de ce facteur. Ces variations sont dues à de mauvaises situations rencontrées lors du trajet (mauvais appariement, véhicule en mouvement, changement d'illumination brusque). L'AFLP évite certaines de ces variations ou atténue leurs effets.

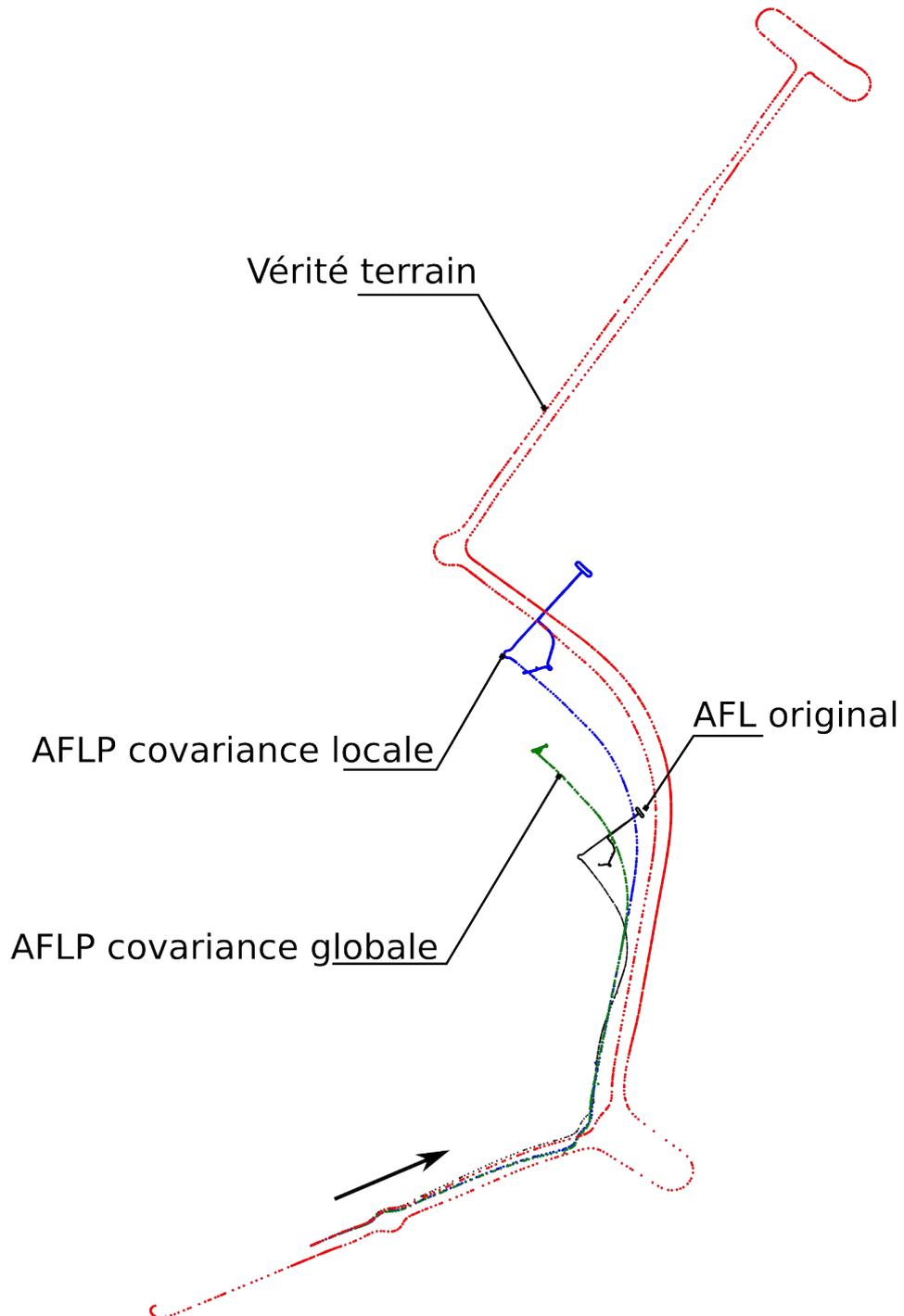


FIGURE 7.6: Vue de dessus des trajectoires reconstruites grâce à l'AFL, L'AFLP local et global et la vérité terrain de la séquence réelle.

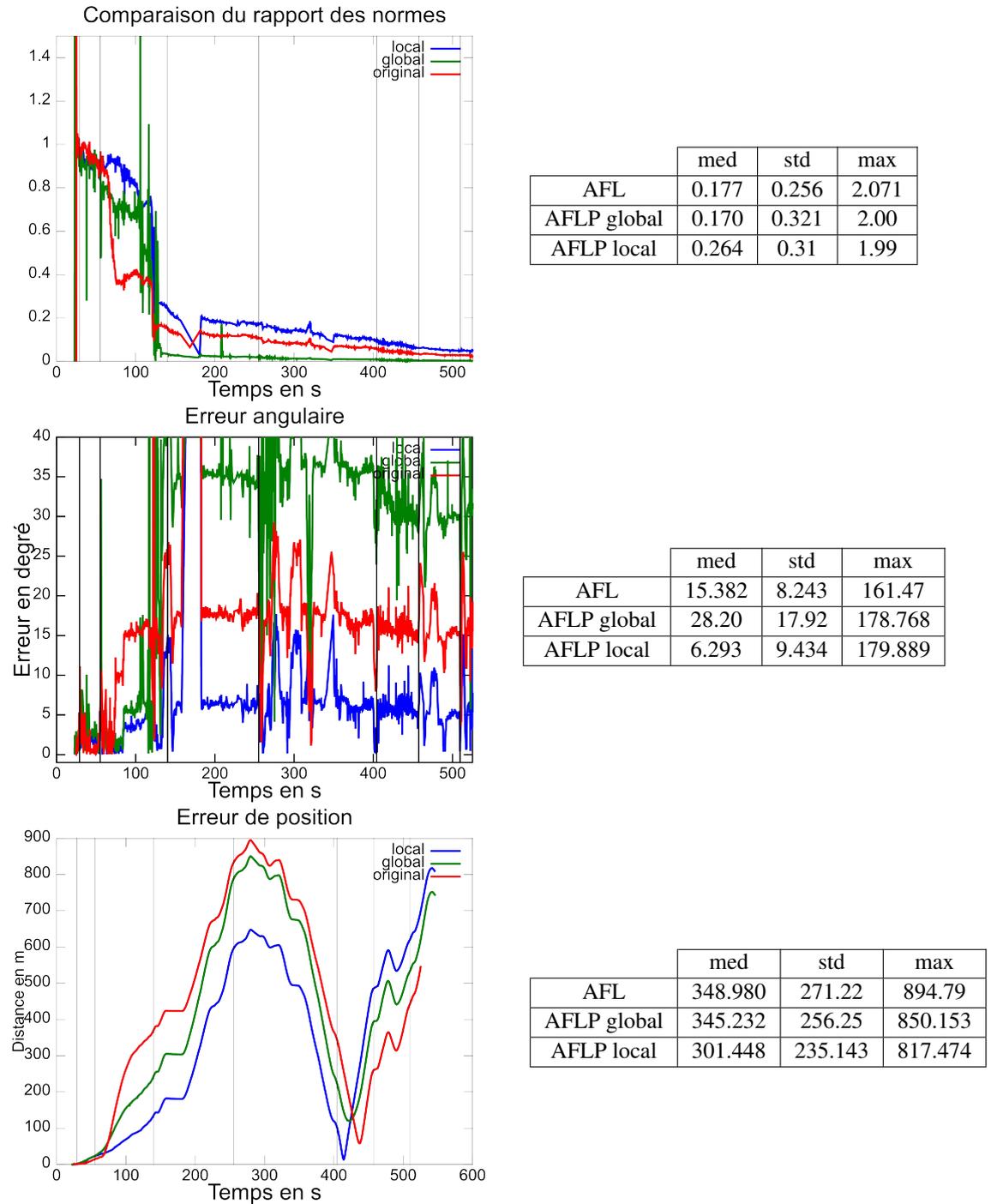


FIGURE 7.7: Evolution du rapport inter-caméra, de l'erreur angulaire et de l'erreur de position au cours du temps pour les différentes méthodes sur la séquence réel.

7.3 Conclusion

Nous avons décrit ici le processus de localisation itératif d'Etienne Mouragnon utilisé pour réaliser du SLAM visuel monoculaire sur de longues séquences. Nous avons ensuite proposé de remplacer l'ajustement de faisceaux local (AFL) par un ajustement de faisceaux local pondéré (AFLP). Le principe est de relâcher les caméras fixes dans l'AFL en leur laissant la possibilité de varier dans un espace contraint défini par la pondération utilisée. Deux AFLP ont été évalués :

- **L'AFLP global**, c'est-à-dire pondéré par une covariance globale. Nous avons montré que bien qu'améliorant les performances au niveau de la stabilité du facteur d'échelle, il est moins stable en cas de forte perturbation que la version proposée par Mouragnon et al.
- **L'AFLP local**, c'est-à-dire pondéré par une covariance locale. Les essais ont montré qu'il augmente sensiblement la qualité de la reconstruction et la stabilité du facteur d'échelle.

Dans tous les cas, les solutions proposées ne permettent que de réduire les erreurs engendrées par la localisation itérative, mais ne permettent pas de résoudre totalement l'accumulation d'erreur ainsi que l'instabilité du facteur d'échelle. Pour pallier ce problème, l'apport d'une métrique par une donnée extérieure est une solution pertinente. Dans le chapitre suivant, nous allons donc explorer les possibilités offertes par la fusion avec d'autres données.

Chapitre 8

Vers la fusion

Sommaire

8.1	Introduction	128
8.2	Fusion avec un odomètre	128
8.3	Intégration simple	130
8.3.1	Résultats	132
8.4	Fusion par AFLP	140
8.4.1	Processus de correction odométrique	141
8.4.2	Intégration du processus de fusion	143
8.4.3	Résultats	144
8.5	Conclusion	150

Ce chapitre présente deux méthodes de fusion utilisant un odomètre pour pallier le problème de la dérive du facteur d'échelle mis en lumière dans le chapitre précédent. Ces méthodes ont fait l'objet de plusieurs publications ([36] et [37]).

8.1 Introduction

Nous avons vu précédemment que le SLAM monoculaire décrit (§7.1) permet de se localiser précisément sur de courtes distances, mais sur des distances plus importantes, il souffre d'une accumulation d'erreurs au cours du temps. Cette accumulation se traduit sur les résultats de deux manières : une erreur sur la pose et une erreur sur le facteur d'échelle. Ces deux problèmes rendent les résultats difficilement exploitables, surtout en cas de saut rapide du facteur d'échelle (dû à de mauvaises observations).

Dans ce chapitre, nous étudions l'apport de la fusion de données issues d'autres capteurs, pour limiter la dérive du facteur d'échelle et ainsi augmenter la précision de localisation. Nous proposons notamment deux solutions différentes pour introduire des données provenant d'un odomètre dans un SLAM visuel utilisant un ajustement de faisceaux.

8.2 Fusion avec un odomètre

La méthode de SLAM monoculaire présentée précédemment permet de reconstruire une trajectoire et son environnement à une échelle près. Normalement, à partir d'une mesure unique de distance entre deux poses, il est possible de mettre à l'échelle la reconstruction. Malheureusement, ce facteur d'échelle théoriquement constant varie au cours de la reconstruction du fait de sa non observabilité et donc de l'absence de contrainte le concernant. Cette dérive rend la reconstruction finale difficilement utilisable en l'état.

Les techniques de localisation par vision les plus couramment utilisées ne rencontrent pas ce problème d'estimation du facteur d'échelle, car elles utilisent plusieurs caméras. En effet, en utilisant plusieurs caméras rigidement liées, la métrique est fournie par la ou les transformations rigides connues entre les caméras. Ainsi, on obtient une reconstruction aux dimensions réelles du monde¹. Par contre, ces techniques engendrent un surcoût en terme de traitement, de matériel et de communication et n'ont pas été retenues ici pour les applications visées. De plus, le système de caméras doit être calibré et le rester.

D'autres approches ont récemment été proposées pour estimer le facteur d'échelle en utilisant une unique caméra, on peut citer les approches de :

- Scaramuzza *et al.* [90] qui utilise le fait que des véhicules non holonomes

1. à la condition d'éviter la configuration centrale où tous les centres optiques sont confondus, qui revient au cas d'une caméra unique

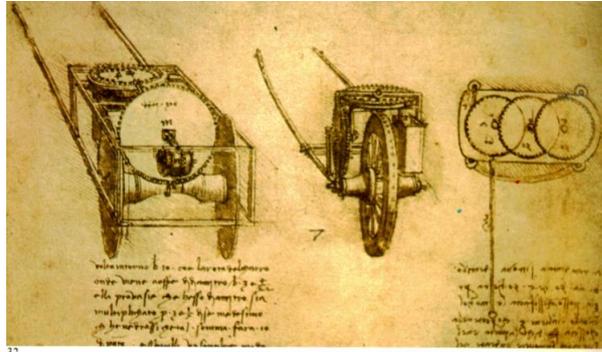


FIGURE 8.1: Odomètre de Leonard de Vinci (Codex Atlanticus f.1.r).

suivent des trajectoires circulaires par morceaux pour déterminer le facteur d'échelle.

- Lothe *et al.* [72] qui suppose connue la distance entre la caméra et le plan du sol.

Malheureusement, ces techniques ne sont pas toujours fiables et ne donnent pas d'information à chaque instant, mais seulement en certains points de la trajectoire.

La dernière possibilité, celle que nous avons choisi d'étudier, est l'utilisation d'un autre capteur pour fournir une information complémentaire à celle estimée par vision. Nous nous sommes plus particulièrement intéressés à l'odomètre.

L'odomètre (Fig. 8.1) est un instrument permettant de mesurer une distance. Deux exemples d'odomètre sont quotidiennement utilisés :

- sur une voiture le compteur de kilomètre est un odomètre ;
- en informatique, une souris est une interface utilisateur constituée d'un couple d'odomètres qui permettent de déplacer un curseur sur l'écran.

Il est intéressant de noter qu'avec le concours de plusieurs odomètres, par exemple un par roue sur un véhicule, il est théoriquement possible d'estimer la trajectoire du véhicule. Mais en pratique, la trajectoire est de mauvaise qualité, surtout à cause des erreurs d'estimation angulaire. Dans notre cas, nous nous concentrerons sur l'utilisation d'un unique odomètre qui nous donne une information assez fiable sur la distance parcourue. Des biais apparaissent en cas de glissement, mais dans notre cas d'étude, ces glissements restent négligeables. Ce capteur est d'autant plus intéressant que c'est un capteur qui est facilement disponible dans les applications automobiles et nous permet de récupérer l'information métrique manquante en vision monoculaire.

Quelques travaux ont déjà été effectués pour fusionner un odomètre avec un

système de vision monoculaire.

Cumani *et al.* [24] reconstruit la trajectoire en estimant pour chaque image clef, un triplet de caméra. Le tenseur trifocal est estimé grâce à l'algorithme des 6 points [88], à partir de correspondance dans les trois dernières images clef. Le nouveau triplet de caméras est ensuite chaîné avec les caméras précédemment estimées. Pour cela, il est nécessaire de déterminer le changement de repère entre les deux reconstructions (le nouveau triplet et la reconstruction précédemment obtenues). L'échelle relative doit notamment être estimée, mais cette estimation peut être de mauvaise qualité en utilisant uniquement la vision. Il propose donc d'utiliser l'odomètre pour mettre à l'échelle les triplets de caméras. Nous exploiterons une idée similaire dans la section 8.3, mais appliqué au SLAM visuel proposé par Mouragnon *et al.* .

Kaess *et al.* [53] propose de fusionner directement l'odomètre dans un ajustement de faisceau local. Dans la section 8.4 nous proposons d'effectuer une fusion de données externes et de l'intégrer ensuite dans l'ajustement de faisceau.

Dans cette section, nous présentons deux techniques pour intégrer la mesure donnée par l'odomètre dans le processus de SLAM visuel. La première est une modification simple permettant d'intégrer la mesure odométrique dans le processus de SLAM sans en modifier le schéma global et le processus d'optimisation. La seconde utilise l'AFLP décrit dans le chapitre 5 pour imposer des contraintes supplémentaires sur les caméras.

8.3 Intégration simple

La première méthode est une méthode simple permettant d'intégrer un facteur d'échelle provenant d'un odomètre à l'intérieur du SLAM visuel.

L'idée de la méthode est de modifier l'initialisation de l'AFL en substituant la pose estimée de la nouvelle image clef t par une pose corrigée. La figure 8.2 illustre la structure du SLAM avec ce nouveau module de correction du facteur d'échelle. La correction appliquée consiste à remettre à l'échelle la pose estimée par la vision grâce à l'information de distance parcourue entre $t - 1$ et t fournie par l'odomètre.

Cette correction illustrée par la figure 8.3, est définie par :

$$\Delta_{\tilde{\mathbf{c}}_t} = \Delta_{\mathbf{c}_{t-1}} + \frac{d_{t-1}}{\|\Delta_{\mathbf{c}_t} - \Delta_{\mathbf{c}_{t-1}}\|_2} (\Delta_{\mathbf{c}_t} - \Delta_{\mathbf{c}_{t-1}}) \quad (8.1)$$

$$\omega_{\tilde{\mathbf{c}}_t} = \omega_{\mathbf{c}_t} \quad (8.2)$$

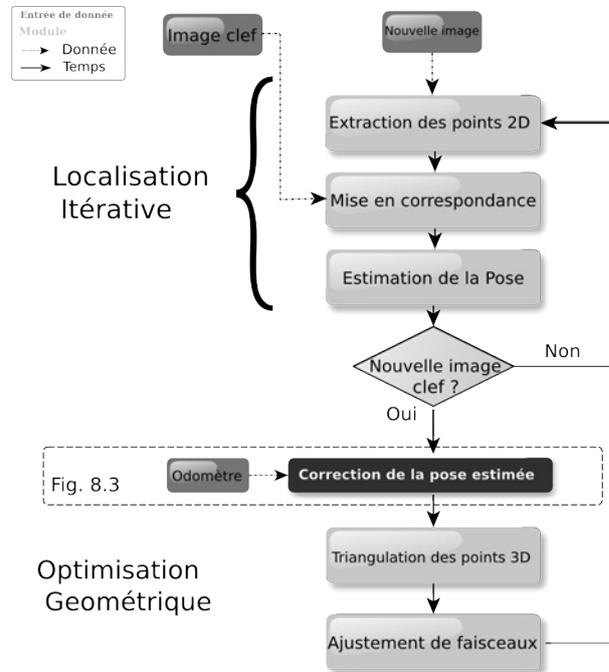


FIGURE 8.2: Positionnement du module correction de la pose estimée dans le schéma du SLAM.

avec les notations suivantes

$t - 1$: image clef précédente

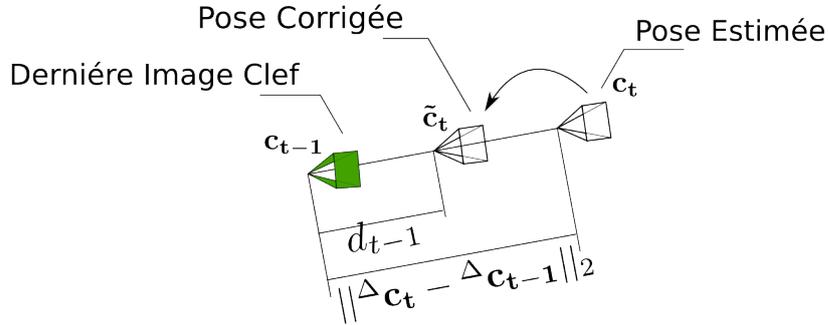
$\Delta \mathbf{c}_t$: position de la caméra au temps t

$\omega \mathbf{c}_t$: orientation de la caméra au temps t

d_{t-1} : la distance odométrique entre $t - 1$ et t

La pose \mathbf{c}_t est modifiée après son estimation et avant la triangulation des nouveaux points 3D. Il est important de noter que cette modification impacte non seulement la pose, mais aussi les nouveaux points 3D lors de l'étape de triangulation. Enfin, la pose corrigée et les nouveaux points 3D sont optimisés par l'AFL avec les anciennes poses et les anciens points 3D comme dans la méthode non modifiée.

La même correction est appliquée à l'initialisation des trois premières caméras de manière à ce que la reconstruction débute avec une échelle correcte.



$$\Delta \tilde{\mathbf{c}}_t = \Delta \mathbf{c}_{t-1} + \frac{d_{t-1}}{\|\Delta \mathbf{c}_t - \Delta \mathbf{c}_{t-1}\|_2} (\Delta \mathbf{c}_t - \Delta \mathbf{c}_{t-1})$$

$$d_{t-1} \quad \text{Distance estimée par l'odomètre}$$

$$\|\Delta \mathbf{c}_t - \Delta \mathbf{c}_{t-1}\|_2 \quad \text{Distance estimée par le SLAM visuel}$$

FIGURE 8.3: Correction de la pose estimée par vision à l'image clef précédente grâce à la distance estimée par l'odomètre.

8.3.1 Résultats

Dans cette section nous présentons les résultats sur deux séquences, une séquence synthétique et une séquence réelle. Nous comparons le résultat de cette méthode intégrant l'odomètre à la version non modifiée du SLAM visuel.

Les trajectoires sont recalées par rapport à la vérité terrain. Pour cela, nous suivons la procédure présentée dans le chapitre précédent (cf section 7.2.2). Pour la nouvelle méthode, le facteur d'échelle n'est pas recalé, mais laissé tel que reconstruit par la méthode.

Pour effectuer la comparaison, nous utilisons les mêmes critères que pour le chapitre précédent (cf section 7.2.2) : l'erreur de position, l'erreur angulaire et le ratio inter-caméra.

Deux types de résultats sont donnés : les vues de dessus des trajectoires reconstruites et les graphiques d'évolution des critères quantitatifs. Des repères communs ont été ajoutés entre les figures pour en faciliter la lecture (les points sur la trajectoire correspondent aux traits verticaux des graphiques).

Résultat sur séquence de synthèse

La séquence utilisée ici est la séquence de synthèse d'un couloir texturé de 365m et 2900 images (§6.3.4). Un odomètre synthétique a été généré à partir de la vérité terrain pour tester la méthode. La mesure odométrique n'est donc pas bruitée.

La Fig. 8.4 présente une vue de dessus des différentes trajectoires :

- la méthode originale purement visuelle notée AFL SLAM (en noir) ;
- la méthode améliorée en utilisant l'odomètre noté AFL SLAM avec odomètre (en bleu) ;
- la vérité terrain (en rouge).

Sur cette figure, nous observons que la méthode corrigée par l'odomètre ferme la boucle pratiquement sans erreur contrairement à la méthode originale.

La figure 8.5 présente les statistiques associées à cette expérience. Elle montre que l'AFL SLAM avec odomètre corrige la dérive du facteur d'échelle. En effet, la dérive du facteur d'échelle, observable par la diminution du rapport de distance inter-caméra au cours du temps, est bien visible pour la méthode originale. Avec la nouvelle méthode, ce rapport reste bien autour de 1. Nous observons également que la contrainte donnée par l'odomètre n'est pas toujours respectée par l'ajustement de faisceaux suivant la correction (par exemple $t=40s$), mais l'ajout de la correction du facteur d'échelle à chaque image clef assure que malgré une possible dérive en position, l'algorithme reviendra tout de même à une échelle correcte.

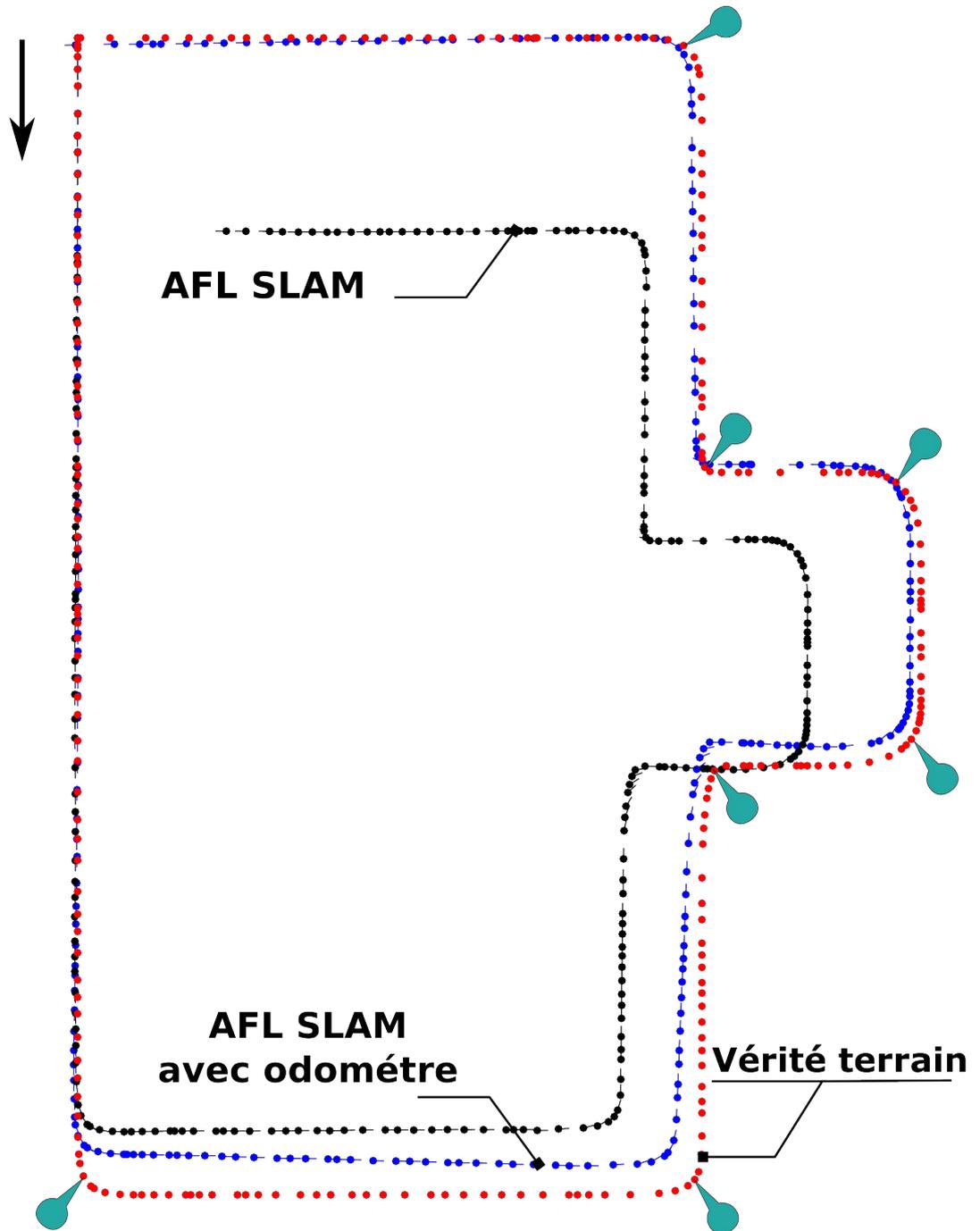


FIGURE 8.4: Trajectoire des différentes méthodes vues de dessus de la séquence synthétique (365m) : vérité terrain (rouge), AFL SLAM (noir), AFL SLAM avec odomètre (bleu).

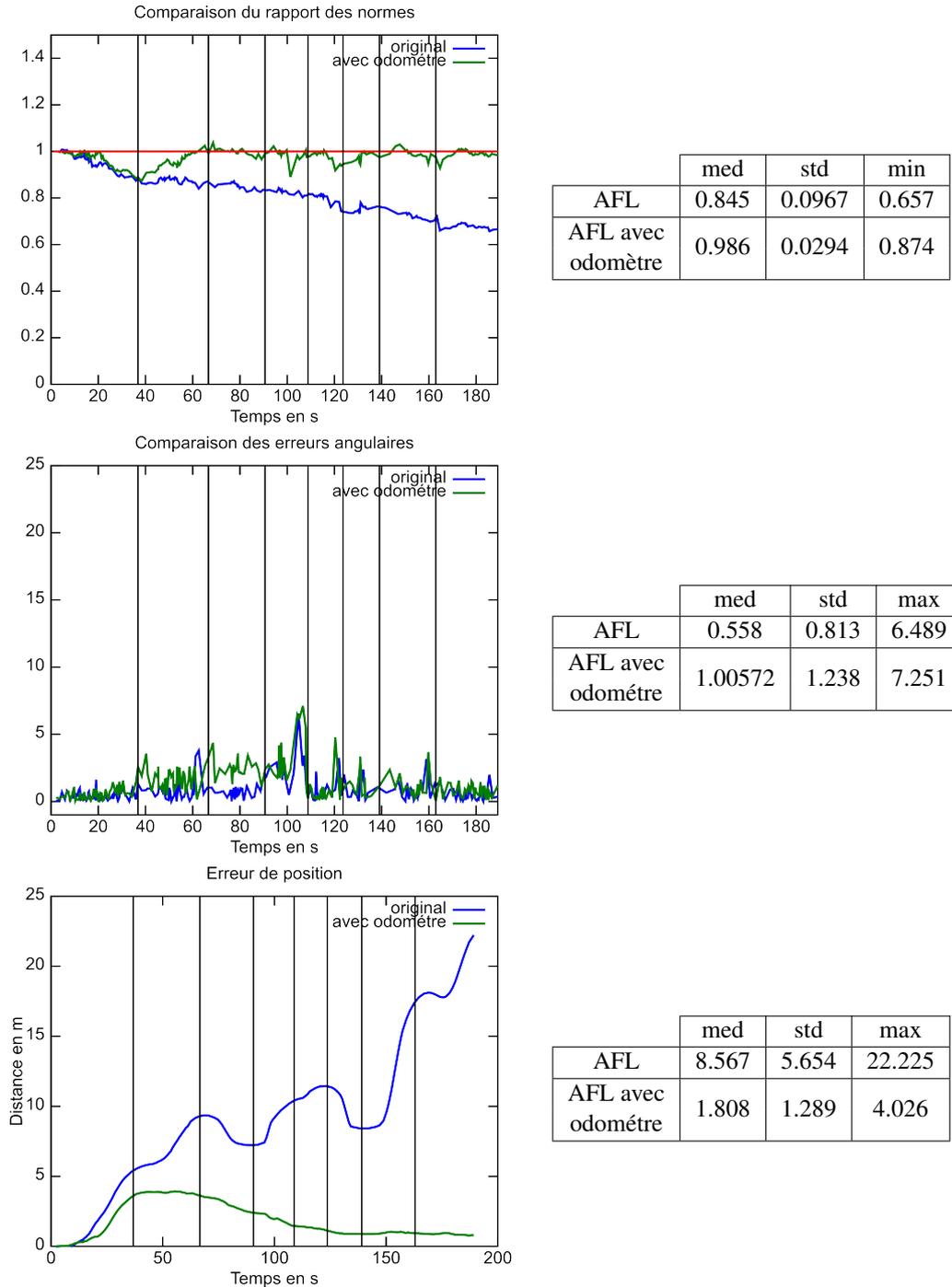


FIGURE 8.5: Evolution du rapport inter-caméra, de l'erreur angulaire et de l'erreur de localisation pour l'AFL SLAM et l'AFL SLAM avec l'odomètre.

Résultats sur la séquence réelle

La séquence réelle est une séquence de 4km en milieu urbain (§6.3.5). La vérité terrain est donnée par un trajectomètre. Ici, nous utilisons directement l'odomètre du véhicule.

La figure 8.6 donne une vue de dessus des trois trajectoires (la vérité terrain, la méthode originale et la méthode utilisant l'odomètre). La reconstruction de 4km montre clairement la dérive de l'échelle de la méthode originale et met en évidence l'apport de la méthode avec odomètre sur la correction de cette dérive (la trajectoire ne s'écrase plus comme précédemment).

Dans la figure Fig. 8.7, nous observons la réduction importante de l'erreur de position (de 345m à 34m) et de l'erreur angulaire (de 16 à 4 degrés). Avec la méthode utilisant l'odomètre, l'erreur de fermeture de boucle sur cette séquence de 4 km est de 30m et l'erreur de positionnement maximum est de 80m.

Nous avons montré ici que même avec un odomètre standard et dans des conditions réelles la méthode utilisant l'odomètre permet de corriger le facteur d'échelle et d'améliorer le positionnement et la reconstruction. A certains instants, l'échelle dérive encore malgré la correction (cf par exemple $t=80s$) mais la correction odométrique permet de retourner vers un état cohérent du facteur d'échelle.

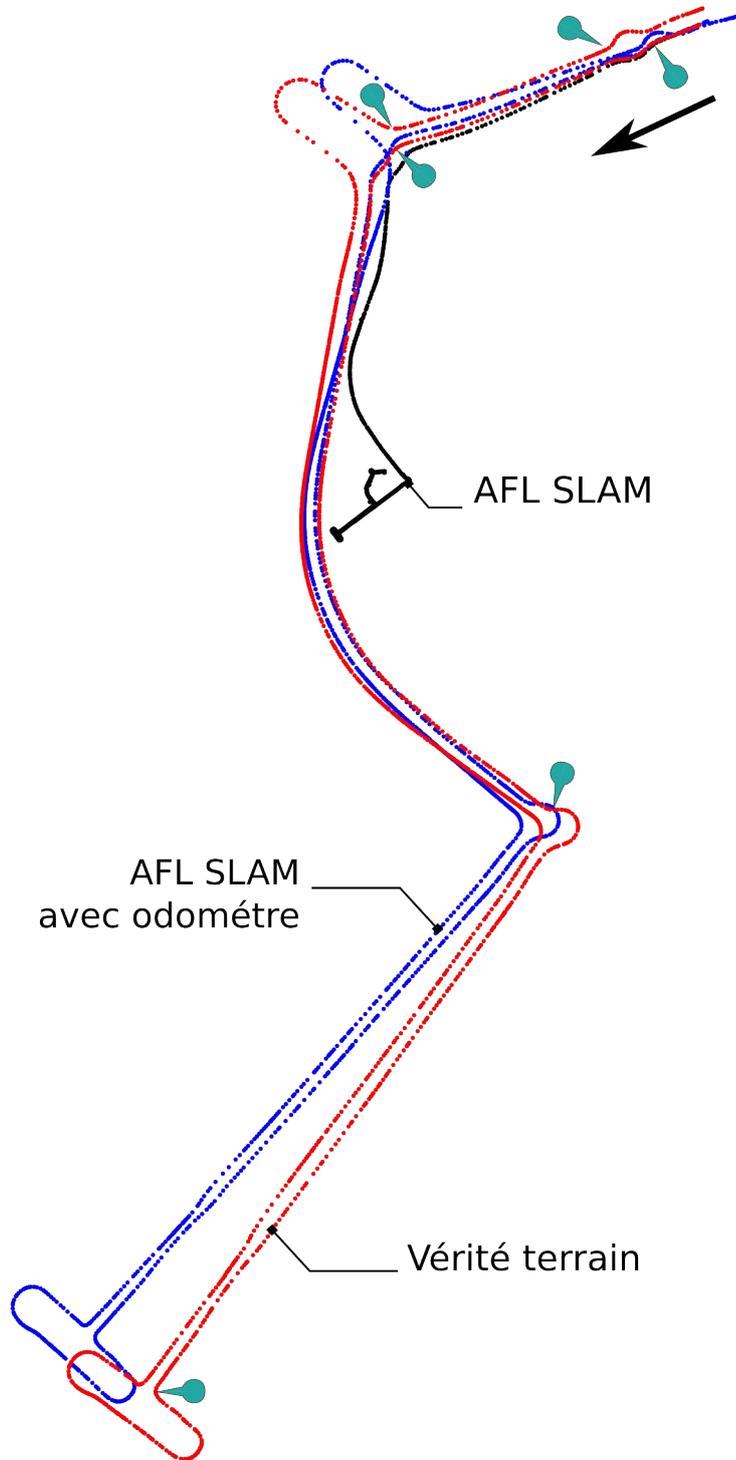


FIGURE 8.6: Vue de dessus des différentes trajectoires.

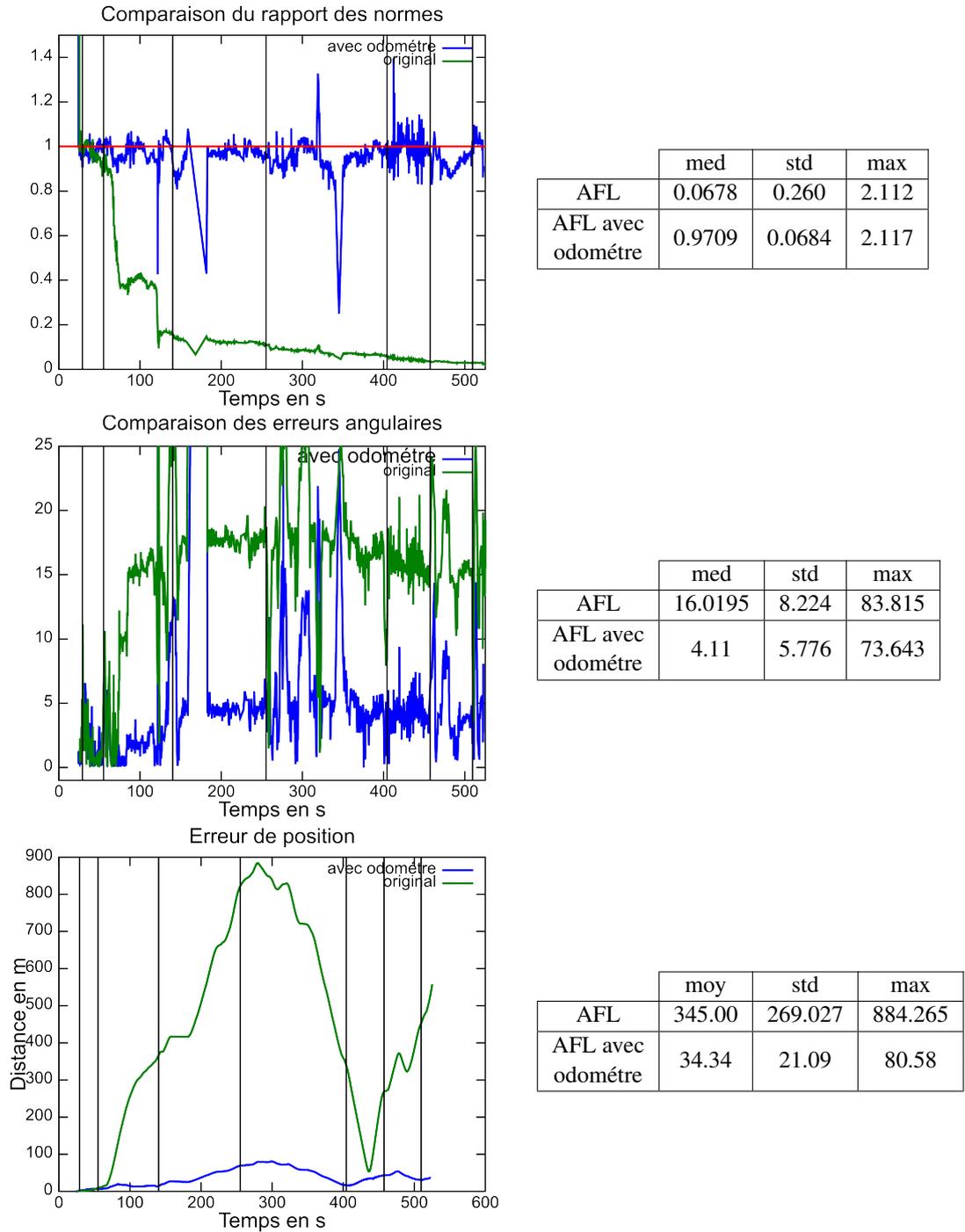


FIGURE 8.7: Evolution des critères avec le temps.

Conclusion

Dans cette section, nous avons démontré la possibilité d'intégrer des données odométriques dans un SLAM visuel monoculaire utilisant un ajustement de faisceau local (AFL). L'odomètre est utilisé pour mettre à l'échelle les nouvelles estimations de pose. Cela se traduit par une mise à l'échelle des points 3D.

Les résultats présentés sur des données réelles et synthétiques montrent que cette méthode améliore significativement les résultats de la méthode originale en stabilisant le facteur d'échelle. Elle permet ainsi d'obtenir une meilleure localisation et reconstruction. Cette trajectoire n'étant plus arbitrairement déformée par la dérive du facteur d'échelle, les techniques de correction comme la fermeture de boucle ou l'utilisation d'une carte pour augmenter la précision de la reconstruction peuvent être utilisées. Il est tout de même à noter que par endroits, le facteur d'échelle continue de dériver du fait que l'ajustement de faisceau ne tient pas toujours correctement compte de la nouvelle position donnée à la caméra contrainte. Un des avantages de cette méthode est sa facilité de mise en oeuvre due aux faibles modifications nécessaires à son application.

Notons que dans cette méthode, nous n'avons pas modifié l'ajustement de faisceau local. Donc, les résultats de la propagation d'erreur restent valables. Ces résultats sont présentés en section 6.3.5.

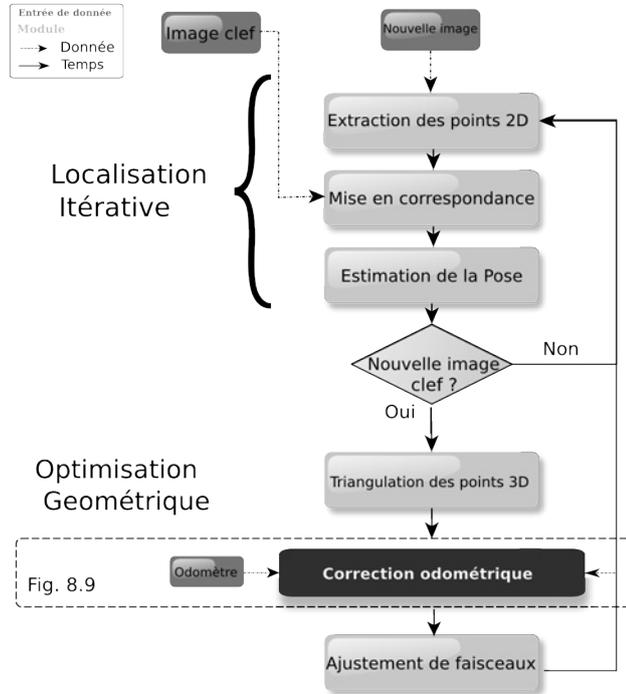


FIGURE 8.8: Positionnement du module correction odométrique dans le schéma du SLAM.

8.4 Fusion par AFLP

Nous avons vu dans la section précédente un moyen pour intégrer une information d'échelle externe simplement en modifiant l'initialisation de l'AFL pour tenir compte de cette information. Mais nous avons vu qu'avec cette méthode, on observe encore ponctuellement une dérive résiduelle du facteur d'échelle. Dans cette section, nous présentons un autre moyen d'intégrer les contraintes provenant de la fusion de données directement dans l'ajustement de faisceau. Dans cette technique, la fusion est faite à l'extérieur de l'ajustement de faisceau et réinjectée dans celui-ci.

Dans l'AFLP, en plus de l'estimation de la position, l'incertitude sur la position est déterminée. Cette information est ensuite utilisée dans le processus d'estimation pour contraindre l'optimisation des anciennes caméras (c_c) à rester proche de leurs estimations précédentes (c_c^-) ceci afin de tenir compte de l'*a priori* sur la position de ces caméras. Pour pouvoir intégrer le processus de vision dans un système de fusion plus large, ces anciennes caméras (c_c^-) ne sont plus seulement

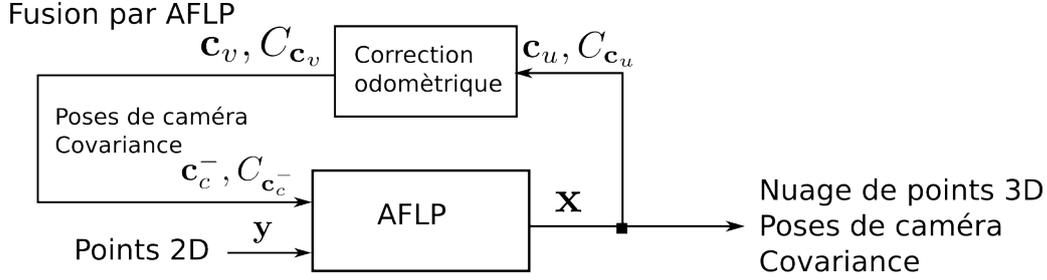


FIGURE 8.9: Schéma de fusion en utilisant l'ajustement de faisceaux local pondéré.

l'estimation des positions par la vision, mais le résultat de la fusion avec d'autres capteurs. Ici, nous proposons de remplacer les anciennes caméras (\mathbf{c}_c^-) par le résultat de la correction par l'odomètre de ces caméras estimées par la vision.

Dans la section suivante, nous décrivons le processus de correction odométrique.

8.4.1 Processus de correction odométrique

Le processus de correction odométrique (voir Fig.8.9) proposé ici prend en entrée un vecteur de poses de caméra (\mathbf{u}) et les distances odométriques correspondantes (\mathbf{d}). Il produit en sortie un ensemble de caméras corrigées (\mathbf{v}). De plus, le processus effectue aussi la propagation d'erreur pour obtenir la covariance des poses corrigées ($C_{\mathbf{v}}$). La connaissance de $C_{\mathbf{u}}$ et $C_{\mathbf{d}}$ sera nécessaire pour cette étape. Chaque caméra va être corrigée par le processus de correction odométrique f^{corr} :

$$f^{corr} : (\mathbf{u}, \mathbf{d}) \mapsto (\mathbf{v}, C_{\mathbf{v}}) = f^{corr}(\mathbf{u}, \mathbf{d}). \quad (8.3)$$

Nous allons maintenant décrire ce processus de correction.

On introduit les notations suivantes :

- n : le nombre de caméras à corriger ($\lfloor \frac{|\mathbf{u}|}{6} \rfloor$).
- $norm_l = \|\Delta \mathbf{u}_{l+1} - \Delta \mathbf{u}_l\|_2$: la distance euclidienne entre la position de \mathbf{u}_l et de \mathbf{u}_{l+1} .
- $\mathbf{dir}_l = \frac{\Delta \mathbf{u}_{l+1} - \Delta \mathbf{u}_l}{norm_l}$: la direction définie par le vecteur $\Delta \mathbf{u}_l \Delta \mathbf{u}_{l+1}$.

Mise à jour de la pose Le processus de correction est un processus récursif : la première caméra est initialisée par $\mathbf{v}_0 = \mathbf{u}_0$.

Puis nous mettons à jour chaque caméra l comme suit :

$$\begin{cases} \omega_{\mathbf{v}_l} = & \omega_{\mathbf{u}_l}. \\ \Delta_{\mathbf{v}_l} = & \Delta_{\mathbf{v}_{l-1}} + d_{l-1} \mathbf{dir}_{l-1}. \end{cases} \quad (8.4)$$

où $\omega_{\mathbf{v}_l}$ (respectivement $\Delta_{\mathbf{v}_l}$) est le sous-vecteur de \mathbf{v}_l correspondant aux paramètres de rotation (respectivement de translation) des poses.

Mise à jour de la covariance La matrice de covariance est déterminée en utilisant la covariance des poses non transformées $C_{\mathbf{u}}$, la covariance de l'odomètre C_d et la transformation que nous venons de définir f^{corr} .

Pour cela, il est nécessaire de calculer le jacobien de \mathbf{v} par rapport à \mathbf{u} et \mathbf{d} . La partie rotation est évidente :

$$J_{\omega_{\mathbf{u}}}^{\omega_{\mathbf{v}}} = \mathbf{I}_{3 \times 3} \quad (8.5)$$

$$J_{\Delta_{\mathbf{u}}}^{\omega_{\mathbf{v}}} = \mathbf{0}_{3 \times 3} \quad (8.6)$$

$$J_{\mathbf{d}}^{\omega_{\mathbf{v}}} = \mathbf{0}_{3 \times 3} \quad (8.7)$$

$$J_{\omega_{\mathbf{u}}}^{\Delta_{\mathbf{v}}} = \mathbf{0}_{3 \times 3} \quad (8.8)$$

Le calcul de $J_{\Delta_{\mathbf{u}}}^{\Delta_{\mathbf{v}}}$ est plus compliqué et nécessite un calcul par récurrence.

A chaque itération, les variables mises en jeu sont $\Delta_{\mathbf{v}_{l-1}}, \Delta_{\mathbf{u}_{l-1}}, \Delta_{\mathbf{u}_l}, d_{l-1}$. Nous calculons donc les jacobiens qui nous permettront de calculer $J_{\Delta_{\mathbf{u}}}^{\Delta_{\mathbf{v}}}$. Ces jacobiens sont :

$$J_{\Delta_{\mathbf{v}_{l-1}}}^{\Delta_{\mathbf{v}_l}} = \mathbf{I} \quad (8.9)$$

$$J_{d_{l-1}}^{\Delta_{\mathbf{v}_l}} = \mathbf{dir}_{l-1} \quad (8.10)$$

$$J_{\Delta_{\mathbf{u}_{l-1}}}^{\mathbf{dir}_{l-1}} = -\frac{1}{norm_{l-1}} (\mathbf{I} - \mathbf{dir}_{l-1} \mathbf{dir}_{l-1}^T) \quad (8.11)$$

$$J_{\Delta_{\mathbf{u}_l}}^{\mathbf{dir}_{l-1}} = \frac{1}{norm_{l-1}} (\mathbf{I} - \mathbf{dir}_{l-1} \mathbf{dir}_{l-1}^T) \quad (8.12)$$

Ainsi nous construisons le jacobien de l'Eq.8.4 par rapport à $\Delta[\mathbf{u}_0 \dots \mathbf{u}_l]$ par récurrence sur l . Le processus de récurrence est :

$$J_{\Delta[\mathbf{u}_0 \dots \mathbf{u}_l]}^{\Delta_{\mathbf{v}_l}} = [J_{\Delta[\mathbf{u}_0 \dots \mathbf{u}_{l-1}]}^{\Delta_{\mathbf{v}_{l-1}}} \mathbf{0}_{3 \times 3}] + d_{l-1} [\mathbf{0}_{3 \times 3(l-1)} J_{\Delta_{\mathbf{u}_{l-1}}}^{\mathbf{dir}_{l-1}} J_{\Delta_{\mathbf{u}_l}}^{\mathbf{dir}_{l-1}}] \quad (8.13)$$

et est initialisé avec $J_{\Delta_{\mathbf{u}_0}}^{\Delta_{\mathbf{v}_0}} = \mathbf{I}_{3 \times 3}$.

Maintenant que tous les jacobiens sont calculés, nous pouvons calculer la covariance de \mathbf{v} :

$$C_{\omega_{\mathbf{v}}} = C_{\omega_{\mathbf{u}}} \quad (8.14)$$

$$C_{\Delta_{\mathbf{v}}} = J_{\Delta_{\mathbf{u}}}^{\Delta_{\mathbf{v}}} C_{\Delta_{\mathbf{u}}} J_{\Delta_{\mathbf{u}}}^{\Delta_{\mathbf{v}T}} + C_d \begin{pmatrix} 0_{3 \times 3} & \dots & \dots & 0_{3 \times 3} & \dots \\ \vdots & J_{d_0}^{\Delta_{\mathbf{v}1T}} J_{d_0}^{\Delta_{\mathbf{v}1}} & \dots & 0_{3 \times 3} & \dots \\ \vdots & \vdots & \ddots & \dots & \dots \\ \vdots & 0_{3 \times 3} & \dots & J_{d_{l-1}}^{\Delta_{\mathbf{v}lT}} J_{d_{l-1}}^{\Delta_{\mathbf{v}l}} & \dots \\ 0_{3 \times 3} & \vdots & \dots & \dots & \ddots \end{pmatrix} \quad (8.15)$$

$$C_{\Delta_{\mathbf{v}/\omega_{\mathbf{v}}}} = C_{\omega_{\mathbf{v}/\Delta_{\mathbf{v}}}}^T = J_{\Delta_{\mathbf{u}}}^{\Delta_{\mathbf{v}}} C_{\Delta_{\mathbf{u}/\omega_{\mathbf{u}}}} \quad (8.16)$$

L'équation Eq.8.16 permet la mise à jour de la corrélation entre les translations et les rotations des caméras pour cela nous utilisons le résultat détaillé §1.3.2 : Eq.1.58 page 25.

Le jacobien de $J_{\Delta_{\mathbf{u}}}^{\Delta_{\mathbf{v}}}$ est une matrice triangulaire supérieur par bloc 3×3 , ce qui peut être utilisé pour accélérer les calculs.

8.4.2 Intégration du processus de fusion

Ici, nous nous proposons de remplacer l'*a priori* sur les poses calculées par vision dans l'utilisation classique de l'AFLP par les poses corrigées par le processus de correction odométrique. Notons que la covariance de l'AFLP utilisée ici ($C_{[\mathbf{c}_{t-N+1}^-, \dots, \mathbf{c}_{t-n}^-]}$) est la covariance locale.

Ainsi :

$$\mathbf{c}_c^- = f^{corr}([\mathbf{c}_{t-N+1}^-, \dots, \mathbf{c}_{t-n}^-], \mathbf{d}) \quad (8.17)$$

avec : \mathbf{d} contenant les distances odométriques entre les caméras $[\mathbf{c}_{t-N+1}^-, \dots, \mathbf{c}_{t-n}^-]$.

Notons que les covariances $C_{[\mathbf{c}_{t-N+1}^-, \dots, \mathbf{c}_{t-n}^-]}$ et C_d sont nécessaires pour obtenir $C_{\mathbf{c}_c^-}$. Nous appliquons l'AFLP à la place de l'AFL avec cet *a priori* dans le SLAM visuel. Le schéma global du SLAM est présenté figure 8.8 ; la figure 8.9 présente le détail de l'intégration du module de correction odométrique. Par la suite, cette méthode sera appelée AFLP avec odomètre.

8.4.3 Résultats

Nous utilisons le même processus d’alignement de trajectoires que précédemment et les mêmes statistiques sont calculées (cf 7.2.2) : l’erreur de position, l’erreur angulaire, le rapport des distances inter-caméras.

Résultats sur la séquence de synthèse La figure 8.10 est une vue de dessus des trajectoires suivantes :

- la méthode originale sans odomètre noté AFL (en noir) ;
- la méthode simple utilisant l’odomètre présenté dans la section précédente notée AFL odomètre (en bleu) ;
- la méthode utilisant l’AFLP pour intégrer les données odométriques notées AFLP odomètre (en vert) ;
- la vérité terrain (en rouge).

Elle nous permet de voir que la méthode utilisant l’AFLP corrige la dérive du facteur d’échelle tout comme la méthode précédente. Nous nous rendons compte sur cette figure que la dérive en échelle sur le premier segment a disparu contrairement à la méthode précédente. Par contre, on observe une erreur angulaire plus importante en fin de trajectoire.

Les statistiques de la figure 8.11 confirment cette observation : la courbe des rapports inter-caméra montre que contrairement à la méthode précédente l’AFLP avec odomètre tient compte à tout instant de la contrainte imposée et le rapport inter-caméra est conforme à celui imposé par l’odomètre. Nous observons effectivement une erreur angulaire plus importante en fin de séquence.

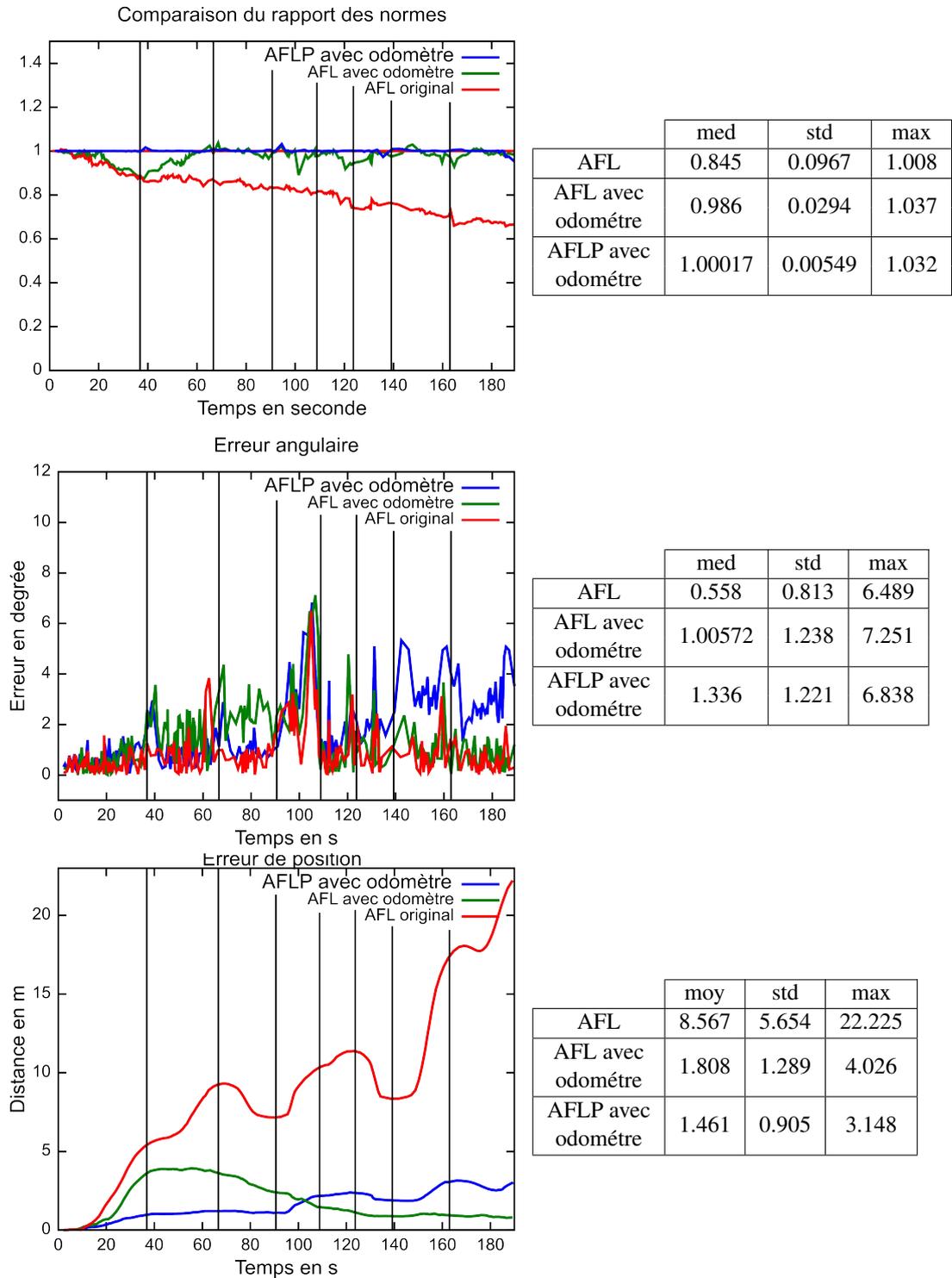


FIGURE 8.11: Evolution du ratio des distances inter-caméra, de l'erreur angulaire et de l'erreur de position sur la séquence de synthèse : comparaison de l'AFL, de l'AFL avec odomètre et l'AFLP avec odomètre.

Résultat sur la séquence réelle La fusion odométrique a été testée sur la séquence réelle de 4 km tournée en milieu urbain. La vérité terrain est un trajectomètre et nous utilisons l'odomètre standard de la voiture. La Fig. 8.12 est une vue de dessus des quatre trajectoires reconstruites :

- la méthode originale sans odomètre noté AFL (en violet).
- la méthode simple utilisant l'odomètre présenté dans la section précédente notée AFL odomètre (en vert).
- la méthode utilisant l'AFLP pour intégrer les données odométriques notées AFLP odomètre (en noir).
- la vérité terrain (en rouge).

Visuellement, la figure 8.12 permet de voir que la méthode utilisant l'AFLP est plus proche de la vérité terrain que la méthode précédente. L'erreur angulaire est diminuée ainsi que l'erreur de fermeture de boucle.

Les statistiques de la figure 8.13 confirment ces observations et montrent que l'erreur en position est de seulement 15m en moyenne soit une réduction de 50% de l'erreur par rapport à la méthode précédente. L'erreur angulaire est sensiblement la même. La courbe des rapports inter-caméra montre que la méthode corrige les dérives du facteur d'échelle : le rapport reste proche de 1 sur toute la séquence et est plus stable que les autres méthodes.

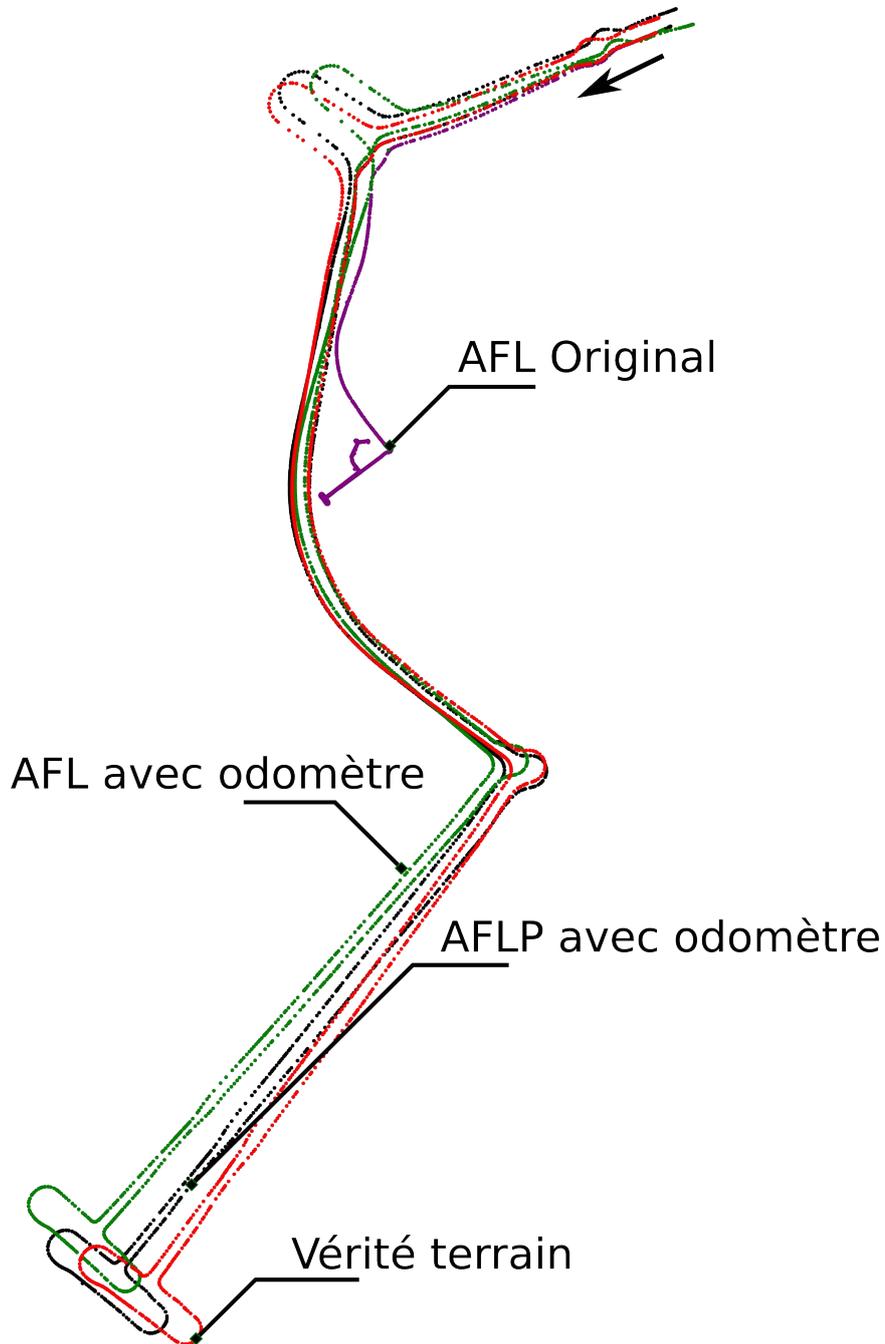


FIGURE 8.12: Vue de dessus de la reconstruction par les différentes méthodes utilisant l'odomètre (AFL et par AFLP), la version non modifiée AFL et la vérité terrain de la séquence réelle.

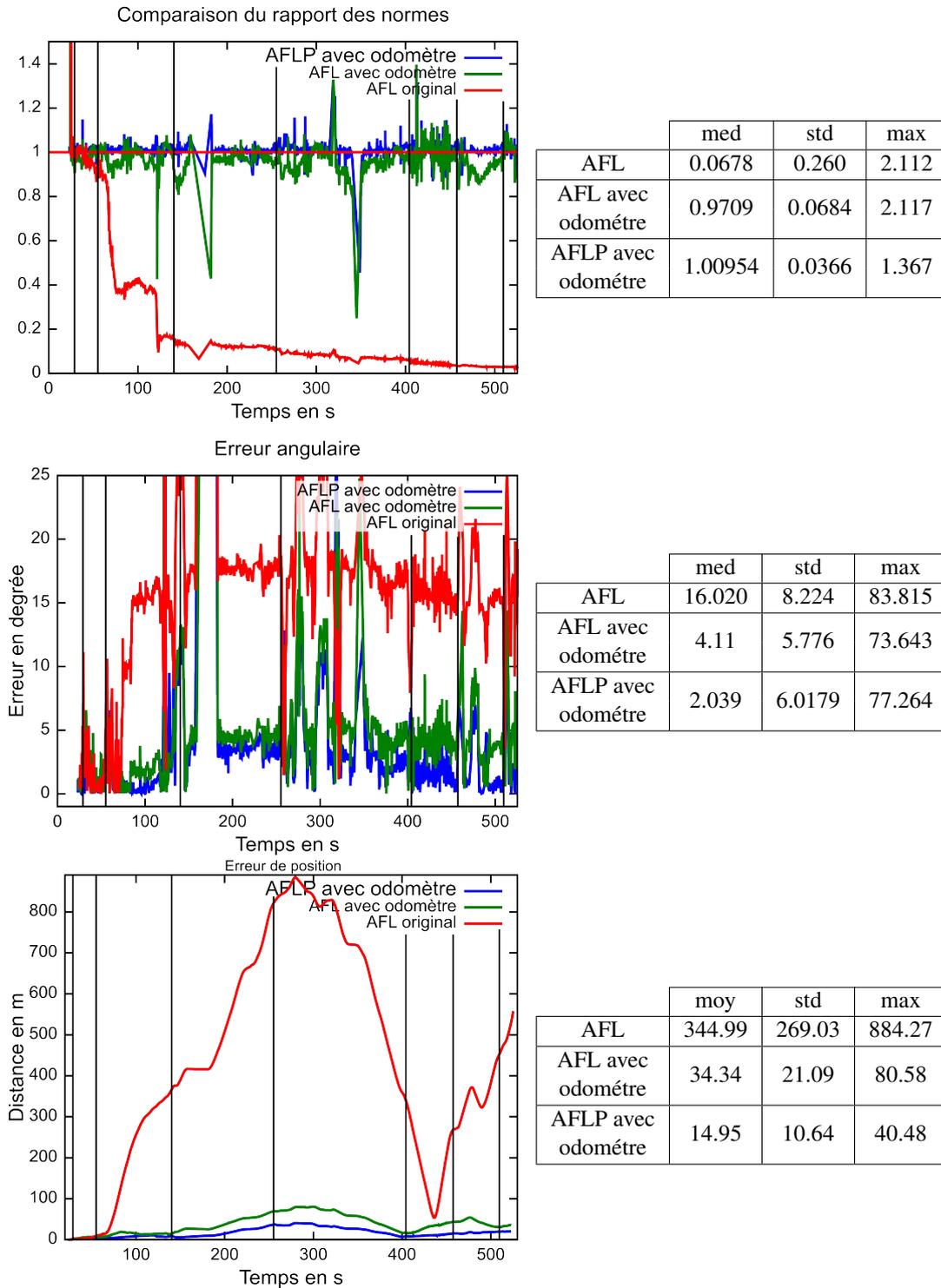


FIGURE 8.13: Evolution du ratio des distances inter-caméra, de l'erreur angulaire et de l'erreur de position sur la séquence réelle de 4 km : comparaison de l'AFL, de l'AFL avec odomètre et l'AFLP avec odomètre.

Conclusion

Dans cette section, nous avons montré que ce processus de fusion est plus efficace que le précédent. Les contraintes apportées par l'odomètre sont mieux intégrées et assurent une stabilité du facteur d'échelle et donc une meilleure précision de localisation.

8.5 Conclusion

Dans ce chapitre nous avons montré que l'utilisation d'un odomètre conjointement à un système de vision monoculaire améliore les performances de la méthode de localisation.

Ici, nous avons proposé deux méthodes permettant d'intégrer le facteur d'échelle provenant d'un odomètre.

La première utilise le système du SLAM monoculaire existant et ajoute une étape de correction odométrique de la pose estimée. La seconde utilise l'AFLP pour intégrer les résultats de la fusion externe entre la vision et un odomètre.

Dans les deux cas, nous avons montré que l'utilisation de cette donnée externe permettait de contraindre le facteur d'échelle et ainsi de rendre la méthode plus robuste aux mauvaises situations et évitant ainsi les sauts du facteur d'échelle existant dans la méthode initiale. Cette stabilisation du facteur d'échelle, nous permet ainsi de diminuer fortement les erreurs de reconstruction et de rendre les trajectoires plus exploitables.

Conclusion

Le travail présenté dans cette thèse s'applique au contexte de localisation de véhicule au moyen d'une caméra embarquée. Il s'appuie sur une technique de SLAM monoculaire reposant sur un ajustement de faisceaux local [80]. Les deux principales limites de la méthode originale appliquée sur des trajectoires de plusieurs kilomètres étaient les suivantes.

- La méthode originale ne fournissait pas d'incertitudes globales c'est-à-dire d'incertitudes relatives au repère du début de la trajectoire. Pour une exploitation optimale des poses des caméras fournies par l'ajustement de faisceaux local, il nous a paru indispensable d'estimer ces incertitudes globales. Cette estimation étant destinée à une application embarquée, elle se doit de ne pas compromettre l'aspect temps réel de la méthode.
- La deuxième limite de cette méthode est une dérive du facteur d'échelle inhérente aux méthodes monoculaires et liée à la non observabilité de l'échelle. Au cours de la séquence, cette dérive est lente dans le cas normal, mais en cas de mauvaises situations (erreur de mise en correspondance, véhicule en mouvement,...), des variations rapides peuvent se produire. Pour pallier ce problème, nous avons étudié la possibilité de fusion avec d'autres capteurs.

Le travail présenté dans cette thèse porte sur ces deux aspects.

Dans un premier temps, nous avons étudié les possibilités de propagations d'erreur dans un ajustement de faisceaux local. Pour obtenir l'incertitude globale des poses de caméras, nous avons étudié et expérimenté une méthode de calcul d'incertitudes reposant sur deux modèles de bruit. Même si l'une de ces méthodes permettait de propager correctement l'incertitude, elle n'était pas adaptée en raison de son coût calculatoire prohibitif.

Nous avons ensuite proposé un nouvel ajustement de faisceaux local, l'ajustement de faisceaux local pondéré. Celui-ci contrairement à celui proposé par Mouragnon *et al.* [80], tient compte de l'incertitude sur les caméras précédemment estimées ; cette incertitude est utilisée comme contrainte dans l'estimation des poses de caméras. Deux méthodes de propagation d'incertitudes ont été étudiées

et expérimentées pour ce nouvel ajustement de faisceaux et nous avons montré que l'une des deux méthodes peut être corrigée et ainsi obtenir une estimation temps réel de l'incertitude.

Pour réduire le problème de la dérive du facteur d'échelle, nous avons proposé deux méthodes d'intégration d'une information provenant d'un odomètre.

La première méthode de fusion permet d'intégrer cette information en modifiant l'initialisation de la méthode d'optimisation du SLAM visuel. L'avantage de cette technique est de stabiliser le facteur d'échelle en imposant qu'une faible modification de la méthode originale. Cependant, elle n'assure pas un facteur d'échelle correct sur toute la trajectoire.

La seconde méthode permet d'imposer les contraintes provenant de l'odomètre. Elle exploite l'ajustement de faisceaux local pondéré pour intégrer des positions corrigées par l'odomètre. Cette méthode permet de supprimer les fortes variations de facteur d'échelle et de réduire fortement les erreurs de localisation.

Pour résumer, nous avons proposé une technique de calcul d'incertitude globale répondant au mieux aux contraintes temps réel de la vision embarquée et des solutions de fusions avec des données odométriques.

Ce travail a mis en évidence plusieurs perspectives qu'il serait souhaitable d'approfondir.

Bien que les méthodes proposées ici permettent de réduire les erreurs de localisation, le processus utilisé est itératif. Il paraît peu probable que ce type de méthode supprime totalement l'accumulation d'erreur. Sur de longues trajectoires, cette erreur va devenir importante. Il est donc nécessaire ici d'ajouter des contraintes absolues. De plus, il est à noter que les méthodes proposées ne donnent pas une pose géoréférencée mais uniquement relative au début de la trajectoire. Plusieurs possibilités s'offrent à nous.

- Nous pouvons utiliser la fermeture des boucles. Cependant, celle-ci nécessite des trajectoires bouclant sur elles même ; ce qui est rarement le cas dans les déplacements de véhicules.
- Il existe aussi les SIG (Système d'information géographique) qui contiennent des informations géographiques sur l'environnement. Ces informations peuvent être des cartes, des images géoréférencées ou encore des modèles 3D plus ou moins précis de l'environnement. Sur ce dernier point, on peut se référer aux travaux de Lothe *et al.* ([71],[72]).
- Enfin, nous pouvons utiliser un GPS. On peut notamment se référer au travail récent de Kume *et al.* [64] sur l'utilisation d'un GPS dans un ajustement

de faisceaux. Sur ce dernier point, notons que l'ajustement de faisceaux local pondéré, semble *a priori* être bien adapté pour intégrer ce type de données et mériterait d'être testé.

Annexe

Documentation technique du trajectomètre



LANDINS

SPECIFICATIONS

PERFORMANCE*

	DGPS	RTK	PPK
True heading [deg]	0.01	0.01	0.01
Roll/Pitch [deg]	0.005	0.005	0.005
Position X and Y [m]	1	0.035	0.02
Position Z [m]	1.5	0.05	0.05

PERFORMANCE* DURING GPS OUTAGES

Outage duration (sec)	15		60		120		300	
	RTK	PPK	RTK	PPK	RTK	PPK	RTK	PPK
True heading [deg]	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Roll/Pitch [deg]	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005
Position X and Y [m]	0.07	0.05	0.20	0.10	0.30	0.25	0.70	0.60
Position Z [m]	0.05	0.05	0.10	0.07	0.25	0.20	0.50	0.40

* All figures are RMS

Specifications subject to change without notice



INS CHARACTERISTICS

Weight:	4.5 kg	Operating temperature:	-10°C to 50°C
Size:	275 mm x 136 mm x 175 mm	Storage temperature:	-40°C to 80°C
Embedded GNSS¹:	L1/L2, Glonass, RTK	Logging capacity:	up to 48 hrs (IMU, INS & GNSS data)
Power:	consumption: < 18W input range: 12 to 32V DC	MTBF:	> 30,000 Hours

INS INTERFACES

3 "event markers":	100 μ s time stamping accuracy, up to 1,000Hz
Output refreshing rate:	up to 200 Hz
Latency:	< 3 ms
Time tagging:	PPS signal
Ethernet 100 Mbits:	-configuration, monitoring, ftp/http access -5 logical ports
2 serial inputs:	RS232/422
2 serial outputs:	RS232/422
Pulses:	3 in/2 out
USB host:	for real-time logging or data transfer
DMI embedded interface:	-TTL dual-phase or single phase & sign signals -12-24V dual-phase or single phase & sign signals -open/close switch & sign signals

¹ Optional



Bibliographie

- [1] S. Abraham and W. Forstner. Calibration Errors in Structure from Motion. In *Mustererkennung 1998, 20. DAGM-Symposium*, page 124. Springer-Verlag, 1998.
- [2] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski. Building rome in a day. In *IEEE 12th International Conference on Computer Vision, 2009*, pages 72–79. IEEE, 2010.
- [3] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM) : Part II. *Robotics & Automation Magazine, IEEE*, 13(3) :108–117, 2006.
- [4] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-SLAM algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568, 2006.
- [5] Y. Bar-Shalom, X.R. Li, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. Wiley-Interscience, 2001.
- [6] H. Bay, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. *ECCV 2006*, pages 404–417, 2006.
- [7] IEC Bipm, ISO IFCC, and O. IUPAC. Guide to the Expression of Uncertainty in Measurement. *ISO, Geneva*, 1993.
- [8] A. Bos. *Parameter estimation for scientists and engineers*. Wiley-Interscience, 2007.
- [9] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *Sixth International Conference on Computer Vision, 1998.*, pages 790–796. IEEE, 2002.
- [10] G.E.P. Box and M.E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2) :610–611, 1958.
- [11] M.J. Box, D. Davies, and Swann. *Non-linear optimization techniques*. Mathematical and statistical techniques for industry. Published for Imperial Chemical Industries Ltd by Oliver & Boyd, 1969.

- [12] S.S. Brandt. On the probabilistic epipolar geometry. *Image and Vision Computing*, 26(3) :405–414, 2008.
- [13] MJ Brooks, W. Chojnacki, D. Gawley, and A. Van Den Hengel. What value covariance information in estimating vision parameters? In *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001.*, volume 1, 2001.
- [14] G. Casella and C.P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1) :81, 1996.
- [15] J.A. Castellanos, J. Neira, and J.D. Tardos. Limits to the consistency of EKF-based SLAM 1. In *IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.
- [16] B. Charmette, E. Royer, and F. Chausse. Efficient planar features matching for robot localization using GPU. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 16–23. IEEE, 2010.
- [17] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway. Real-time and robust monocular slam using predictive multi-resolution descriptors. *Advances in Visual Computing*, pages 276–285, 2006.
- [18] A.R. Chowdhury and R. Chellappa. Statistical error propagation in 3d modeling from monocular video. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 8, page 89. IEEE, 2008.
- [19] J. Civera, O.G. Grasa, A.J. Davison, and JMM Montiel. 1-point RANSAC for EKF-based structure from motion. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3498–3504. IEEE, 2009.
- [20] AI Comport, E. Malis, and P. Rives. Real-time Quadrifocal Visual Odometry. *International Journal of Robotics Research*, 29(2-3) :245–266, 2010.
- [21] A. Criminisi. *Accurate visual metrology from single and multiple uncalibrated images*. Springer Verlag, 2001.
- [22] M. Csorba. *Simultaneous localisation and map building*. University of Oxford, 1998.
- [23] G. Csurka, C. Zeller, Z. Zhang, and OD Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer vision and image understanding*, 68(1) :18–36, 1997.
- [24] A. Cumani, S. Denasi, A. Guiducci, and G. Quaglia. Integrating monocular vision and odometry for SLAM. *WSEAS Transactions on Computers*, 3(3) :625–630, 2004.

- [25] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.*, pages 1403–1410. IEEE, 2003.
- [26] F. Dellaert and M. Kaess. Square Root SAM : Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12) :1181, 2006.
- [27] A. Doucet, N. de Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. *Sequential Monte Carlo methods in practice*, pages 3–14, 2001.
- [28] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 176–183, 2000.
- [29] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping : part I. *IEEE Robotics & Automation Magazine*, 13(2) :99–110, 2006.
- [30] E. Eade and T. Drummond. Scalable monocular SLAM. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006*, volume 1, pages 469–476. IEEE, 2006.
- [31] C. Engels and D. Nistér. Global uncertainty in epipolar geometry via fully and partially data-driven sampling. In *ISPRS Workshop BenCOS : Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*, pages 17–22, 2005.
- [32] C. Engels, H. Stewénus, and D. Nistér. Bundle adjustment rules. *Photogrammetric Computer Vision*, 2, 2006.
- [33] A. Eudes and Maxime Lhuillier. Error propagations for local bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, Miami, Florida, June 2009.
- [34] A. Eudes and Maxime Lhuillier. Propagations d'erreur pour l'ajustement de faisceaux local. In *Actes de la conférence ORASIS'09*, Tregastel, France, June 2009.
- [35] A. Eudes, Maxime Lhuillier, and Sylvie Naudet-Collette. Propagations d'erreur pour l'ajustement de faisceaux local. In *Actes de la conférence RFIA'10*, Caen, France, January 2010.
- [36] A. Eudes, Maxime Lhuillier, Sylvie Naudet-Collette, and Michel Dhome. Fast odometry integration in local bundle adjustment-based visual slam. In *ICPR*, 2010.

- [37] Alexandre Eudes, Sylvie Naudet-Collette, Maxime Lhuillier, and Michel Dhome. Weighted local bundle adjustment and application to odometry and visual slam fusion. In *Proceedings of the British Machine Vision Conference*, pages 25.1–25.10. BMVA Press, 2010. doi :10.5244/C.24.25.
- [38] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Proceedings of Robotics : Science and Systems*, volume 2, 2005.
- [39] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig ? In *ECCV'92*, pages 563–578. Springer, 1992.
- [40] O. Faugeras. *Three-dimensional computer vision : a geometric viewpoint*. the MIT Press, 1993.
- [41] J.A. Fessler. Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood) : applications to tomography. *IEEE Transactions on Image Processing*, 5(3) :493–506, 1996.
- [42] M.A. Fischler and R.C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.
- [43] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. *ECCV'98*, pages 311–326, 1998.
- [44] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proc. of Robotics : Science and Systems (RSS)*, 2007.
- [45] RM Haralick. Propagating covariance in computer vision. *International journal of pattern recognition and artificial intelligence*, 10(5) :561–572, 1996.
- [46] R.M. Haralick, H. Joo, C.N. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6) :1426–1446, 1989.
- [47] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50, 1988.
- [48] R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *ECCV'92*, pages 579–587. Springer, 1992.
- [49] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.
- [50] R.I. Hartley. In defence of the 8-point algorithm. In *Fifth International Conference on Computer Vision, 1995.*, pages 1064–1070, 2002.

- [51] R.I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2) :146–157, 1997.
- [52] Yekeun Jeong, David Nister, Drew Steedly, Richard Szeliski, and InSo Kweon. Pushing the Envelope of Modern Methods for Bundle Adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [53] M. Kaess and F. Dellaert. Visual slam with a multi-camera rig. *Georgia Institute of Technology, Tech. Rep. GIT-GVU-06-06*, 2006.
- [54] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM : Incremental smoothing and mapping. *IEEE Trans. on Robotics (TRO)*, 24(6) :1365–1378, Dec 2008.
- [55] R.E. Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1) :35–45, 1960.
- [56] K. Kanatani. *Statistical optimization for geometric computation : theory and practice*. Elsevier Science Inc. New York, NY, USA, 1996.
- [57] K. Kanatani. Uncertainty modeling and geometric inference. *Handbook of Geometric Computing*, pages 461–491, 2004.
- [58] K. Kanatani and D.D. Morris. Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 47(5) :2017, 2001.
- [59] K. Kanatani, Y. Sugaya, and H. Niitsuma. Triangulation from two views revisited : Hartley-Sturm vs. optimal correction. In *Proc. 19th British Machine Vision Conf*, pages 173–182, 2008.
- [60] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2008.
- [61] K. Konolige and M. Agrawal. Frameslam : From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on*, 24(5) :1066–1077, 2008.
- [62] K. Konolige, M. Agrawal, and J. Sola. Large scale visual odometry for rough terrain. In *Proc. International Symposium on Robotics Research*, 2007.
- [63] Kurt Konolige. Sparse sparse bundle adjustment. In *Proc. BMVC*, pages 102.1–11, 2010. doi :10.5244/C.24.102.
- [64] Hideyuki Kume, Takafumi Taketomi, Tomokazu Sato, and Naokazu Yokoya. Extrinsic camera parameter estimation using video images and gps considering gps positioning accuracy. In *ICPR*, pages 3923–3926, 2010.

- [65] J.M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration ? *ECCV'98*, page 158, 1998.
- [66] T. Lemaire, C. Berger, I.K. Jung, and S. Lacroix. Vision-based slam : Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3) :343–364, 2007.
- [67] J. Leonard and P. Newman. Consistent, convergent, and constant-time SLAM. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 1143–1150, 2003.
- [68] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp : An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2) :155–166, 2009.
- [69] M. Lhuillier. Effective and generic structure from motion using angular error. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 67–70, 2006.
- [70] M. Lhuillier and M. Perriollat. Uncertainty ellipsoids calculations for complex 3D reconstructions. In *International Conference on Robotic and Automation*, 2006.
- [71] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Monocular SLAM Reconstructions and 3D City Models : Towards a Deep Consistency. *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, pages 201–214, 2010.
- [72] P. Lothe, S. Bourgeois, E. Royer, M. Dhome, and S. Naudet-Collette. Real-time vehicle global localisation with a single camera in dense urban areas : Exploitation of coarse 3D city models. In *Computer Vision and Pattern Recognition (CVPR)*, pages 863–870. IEEE, 2010.
- [73] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.
- [74] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3) :169–186, 2007.
- [75] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1) :63–86, 2004.
- [76] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 1151–1156, 2003.

- [77] DD Morris. *Gauge Freedoms and Uncertainty Modeling for 3D Computer Vision*. PhD thesis, Carnegie Mellon University, the Robotics Institute, 2001.
- [78] D.D. Morris, K. Kanatani, and T. Kanade. Gauge fixing for accurate 3D estimation. *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2, 2001.
- [79] E. Mouragnon. *Reconstruction 3D et localisation simultanée de caméras mobiles : une approche temps-réel par ajustement de faisceaux local*. PhD thesis, Université Blaise Pascal de Clermont-Ferrant, 2006.
- [80] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *CVPR'06*, 2006.
- [81] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *5th Int. Symposium on Robotics Research*, volume 1, 1989.
- [82] D. Nistér. *Automatic dense reconstruction from uncalibrated video sequences*. PhD thesis, KTH, 2001.
- [83] D. Nistér. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6) :756–770, 2004.
- [84] P. Piniés and J.D. Tardós. Scalable SLAM building conditionally independent local maps. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3466–3471. IEEE, 2007.
- [85] P. Piniés and J.D. Tardós. Large-scale slam building conditionally independent local maps : Application to monocular vision. *IEEE Transactions on Robotics*, 24(5) :1094–1106, 2008.
- [86] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Automated reconstruction of 3D scenes from sequences of images. *ISPRS Journal Of Photogrammetry And Remote Sensing*, 55(4) :251–267, 2000.
- [87] M. Pollefeys, D. Nistér, J.M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2) :143–167, 2008.
- [88] L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1) :34–46, 2002.
- [89] E. Royer, M. Lhuillier, M. Dhome, and J.M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3) :237–260, 2007.

- [90] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *IEEE 12th International Conference on Computer Vision, 2009*, pages 1413–1419. IEEE, 2010.
- [91] H.Y. Shum, Q. Ke, and Z. Zhang. Efficient bundle adjustment with virtual key frames : A hierarchical approach to multi-frame structure from motion. In *Computer Vision and Pattern Recognition, 1999.*, volume 2. IEEE, 2002.
- [92] G. Sibley. A Sliding Window Filter for SLAM. *Technical report, University of Southern California, Center for Robotics and Embedded Systems*, 2006.
- [93] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. *The International Journal of Robotics Research*, 29(8) :958, 2010.
- [94] P. Smith, T. Drummond, and K. Roussopoulos. Computing MAP trajectories by representing, propagating and combining PDFs over groups. In *Ninth IEEE International Conference on Computer Vision, 2003.*, pages 1275–1282. IEEE, 2008.
- [95] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4) :56, 1986.
- [96] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism : exploring photo collections in 3D. In *ACM SIGGRAPH 2006*, pages 835–846. ACM, 2006.
- [97] H. Strasdat, JMM Montiel, and A.J. Davison. Real-time monocular SLAM : Why filter ? In *IEEE International Conference on Robotics and Automation (ICRA), 2010*, pages 2657–2664, 2010.
- [98] H. Strasdat, JMM Montiel, and A.J. Davison. Scale drift-aware large scale monocular SLAM. In *Proceedings of Robotics : Science and Systems (RSS)*, 2010.
- [99] Y. Sugaya and K. Kanatani. High accuracy computation of rank-constrained fundamental matrix. In *Proc. 18th British Machine Vision Conf*, volume 1, pages 282–291, 2007.
- [100] F. Sur, N. Noury, M. Berger, and I.N. Grand-est. Computing the uncertainty of the 8 point algorithm for fundamental matrix estimation. In *In Proc. BMVC*, 2008.
- [101] T. Svoboda and P. Sturm. A badly calibrated camera in ego-motion estimation—propagation of uncertainty. In *Computer Analysis of Images and Patterns*, pages 183–190. Springer, 1997.

-
- [102] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithm : Theory and Practice*, 1999.
- [103] J. Wallner, R. Krasauskas, and H. Pottmann. Error propagation in geometric constructions. *Computer Aided Design*, 32(11) :631–642, 2000.
- [104] S. Yi, R.M. Haralick, and L.G. Shapiro. Error propagation in machine vision. *Machine Vision and Applications*, 7(2) :93–114, 1994.
- [105] Z. Zhang. Determining the epipolar geometry and its uncertainty : A review. *International Journal of Computer Vision*, 27(2) :161–195, 1998.
- [106] Z. Zhang and C. Loop. Estimating the fundamental matrix by transforming image points in projective space. *Computer Vision and Image Understanding*, 82(2) :174–180, 2001.

Table des figures

1.1	Echantillonnage , cas gaussien 1D.	22
1.2	Propagation , cas gaussien 1D.	23
1.3	Estimation des paramètres de sortie, cas gaussien 1D.	23
3.1	Exemple d'utilisation d'un ajustement de faisceaux pour la reconstruction de scène 3D. A gauche la reconstruction avant ajustement de faisceaux. A droite après l'ajustement. Cette image a été présentée par Yekeun Jeong <i>et al.</i> [52].	38
3.2	Présentation de l'erreur entre les faisceaux 3D et 2D	39
3.3	Exemple d'optimisation par AFG, présentant graphiquement les variables utilisées.	40
3.4	Structure de la matrice Jacobienne	42
3.5	Structure de l'approximation de la Hessienne	43
3.6	Entrée/sortie du système ajustement de faisceaux global	47
4.1	Exemple d'optimisation par AFL, présentant graphiquement les variables utilisées	53
4.2	Entrées/sorties de l'Ajustement de faisceaux local	55
4.3	Récurrance pour le cas avec corrélation	58
4.4	Algorithme de calcul de la covariance de l'ajustement de faisceaux local (cas avec corrélation).	59
4.5	Récurrance pour le cas avec indépendance	61
4.6	Algorithme de calcul de la covariance de l'ajustement de faisceaux local (cas avec hypothèse d'indépendance).	62
5.1	Exemple d'optimisation par AFLP, présentant graphiquement les variables utilisées.	66

5.2	Entrée/sorties de l'Ajustement de faisceaux local pondéré.	69
5.3	Récurrence pour le cas avec corrélation.	72
5.4	Résumé de l'estimation par l'AFLP et du calcul de covariance, en tenant compte de la corrélation entre observation et caméras déjà estimées.	73
5.5	Récurrence pour le cas avec indépendance.	75
5.6	Résumé de l'estimation par l'AFLP et du calcul de covariance, cas avec hypothèse d'indépendance entre observation et caméras déjà estimées.	76
6.1	Longueur du grand axes des ellipses d'incertitude en log10 pour chacune des méthodes : L'AFL avec hypothèse de corrélation, l'AFLP avec indépendance.	83
6.2	Variation de la covariance en fonction du bruit sur les points 2D.	84
6.3	Vue de dessus des différentes incertitudes	84
6.4	Comparaison avec la covariance globale sur la séquence en ligne droite.	85
6.5	Vue de dessus de la séquence en virage.	86
6.6	Comparaison avec la covariance globale sur la séquence en virage.	88
6.7	Trois images de la séquence de synthèse.	89
6.8	Vue de dessus de la séquence de synthèse.	90
6.9	Comparaison avec la covariance globale sur la séquence d'image de synthèse.	91
6.10	Photographie du trajectomètre utilisé.	92
6.11	Trois images de la séquence réelle.	93
6.12	Vue de dessus de la séquence corrigée par ajout de l'odomètre.	94
6.13	Comparaison avec la covariance globale séquence de réels corrigés par odomètre §8.3.	95
6.14	Cette figure présente pour la position de la nouvelle caméra : les différentes positions résultats du tirage aléatoire et les ellipses d'incertitude estimées numériquement et analytiquement. Elle présente également le rapport des coefficients de la matrice de covariance 3×3 estimés analytiquement sur ceux de la matrice de covariance estimés numériquement.	99

6.15	Cette figure présente pour la position de la nouvelle caméra : les différentes positions résultats du tirage aléatoire et les ellipses d'incertitude estimées numériquement et analytiquement. Elle présente également le rapport des coefficients de la matrice de covariance 3×3 estimés analytiquement sur ceux de la matrice de covariance estimés numériquement.	101
6.16	Cette figure présente les 50 trajectoires bruitées pour le calcul de consistance sur la séquence en virage.	105
6.17	Consistance de l'AFL avec corrélation. Les deux figures représentent la distance NEES en fonction du temps en images clef. Sur la séquence en ligne droite (a) et sur la séquence en virage (b). . . .	106
6.18	Consistance de l'AFLP avec indépendance. Les deux figures représentent la distance NEES en fonction du temps en images clefs. Sur la séquence en ligne droite (b) et sur la séquence en virage (a). . .	106
6.19	Cette figure présente une représentation en vue de dessus des ellipses d'incertitude calculée par deux méthodes : la méthode AFG (trait plein) et la méthode AFLP corrigée (pointillés).	108
7.1	Schéma général de l'algorithme utilisé.	110
7.2	Phase de localisation.	113
7.3	Reconstruction 3D et optimisation géométrique.	114
7.4	Vue de dessus des trajectoires reconstruites par AFL, AFLP local et global et la vérité terrain de la séquence de synthèse.	120
7.5	Evolution du rapport inter-caméra, de l'erreur angulaire et de l'erreur de position au cours du temps pour les différentes méthodes sur la séquence de synthèse.	121
7.6	Vue de dessus des trajectoires reconstruites grâce à l'AFL, L'AFLP local et global et la vérité terrain de la séquence réelle.	123
7.7	Evolution du rapport inter-caméra, de l'erreur angulaire et de l'erreur de position au cours du temps pour les différentes méthodes sur la séquence réel.	124
8.1	Odomètre de Leonard de Vinci (Codex Atlanticus f.1.r).	129
8.2	Positionnement du module correction de la pose estimée dans le schéma du SLAM.	131
8.3	Correction de la pose estimée par vision à l'image clef précédente grâce à la distance estimée par l'odomètre.	132

8.4	Trajectoire des différentes méthodes vues de dessus de la séquence synthétique (365m) : vérité terrain (rouge), AFL SLAM (noir), AFL SLAM avec odomètre (bleu).	134
8.5	Evolution du rapport inter-caméra, de l'erreur angulaire et de l'erreur de localisation pour l'AFL SLAM et l'AFL SLAM avec l'odomètre.	135
8.6	Vue de dessus des différentes trajectoires.	137
8.7	Evolution des critères avec le temps.	138
8.8	Positionnement du module correction odométrique dans le schéma du SLAM.	140
8.9	Schéma de fusion en utilisant l'ajustement de faisceaux local pondéré.	141
8.10	Vue de dessus de la reconstruction par les différentes méthodes utilisant l'odomètre (AFL et par AFLP), la version non modifiée AFL et la vérité terrain de la séquence réelle.	145
8.11	Evolution du ratio des distances inter-caméra, de l'erreur angulaire et de l'erreur de position sur la séquence de synthèse : comparaison de l'AFL, de l'AFL avec odomètre et l'AFLP avec odomètre.	146
8.12	Vue de dessus de la reconstruction par les différentes méthodes utilisant l'odomètre (AFL et par AFLP), la version non modifiée AFL et la vérité terrain de la séquence réelle.	148
8.13	Evolution du ratio des distances inter-caméra, de l'erreur angulaire et de l'erreur de position sur la séquence réelle de 4 km : comparaison de l'AFL, de l'AFL avec odomètre et l'AFLP avec odomètre.	149

Liste des tableaux

6.1	Caractérisation des séquences utilisées.	82
6.2	Statistiques sur la distribution d'entrée (rapport des coefficients des covariances).	98
6.4	Comparaison de l'ellipsoïde (correspondant à la position (x,y,z) de la dernière caméra) estimée analytiquement par rapport à l'estimation numérique fournie par Monte-Carlo.	98
6.3	Comparaison des covariances (Rapport analytique/numérique). . .	99
6.5	Statistiques sur les erreurs de distribution d'entrée (rapport des coefficients des covariances).	100
6.6	Statistiques sur les erreurs de l'ellipsoïde correspondant à la position (x,y,z) de la dernière caméra estimée analytiquement par rapport à l'estimation numérique.	100
6.7	Comparaison des covariances (Rapport analytique/numérique). . .	101