Localisation d'un véhicule à l'aide d'un SLAM visuel contraint

Dorra Larnaout

N° d'ordre : D.U. 2454
EDSPIC : 652

UNIVERSITÉ BLAISE PASCAL - CLERMONT-FERRAND II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée par :
Dorra LARNAOUT

Formation Doctorale :
Électronique et Système

en vue de l'obtention du grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : Vision pour la Robotique

Localisation d'un véhicule à l'aide d'un SLAM visuel contraint

---

M. Steve BOURGEOIS	Examinateur
M. Vincent GAY-BELLILE	Examinateur
Mme. Marie-Odile BERGER	Rapporteur
M. Michel DHOME	Directeur de thèse
M. Nicolas PAPARODITIS	Président du jury
M. François CHAUMETTE	Rapporteur

Résumé

Pour se localiser en ville, la majorité des solutions commercialisées se base sur les systèmes GPS. Même si ces systèmes offrent une précision suffisante hors agglomération, celle-ci se dégradent considérablement en villes à cause des phénomènes connus sous le nom du canyon urbain (*i.e.* réflexion du signal GPS sur les façades des bâtiments). Pour pallier ce problème, les solutions basées sur un SLAM visuel (Simultaneous Localization And Mapping) semblent une alternative prometteuse. En plus de l'estimation des six degrés de liberté de la caméra mobile, il fournit une carte 3D de la scène observée. Toutefois, la localisation assurée par le SLAM visuel n'est pas géo-référencée et présente souvent des dérives (*e.g.* mauvaise estimation du facteur d'échelle, accumulation des erreurs).

Pour faire face à ces limitations et afin de proposer une solution facile à déployer, nous avons étudié la possibilité d'intégrer au SLAM des informations supplémentaires qui pourraient contraindre l'ensemble de la reconstruction fournie. Ces dernières doivent alors être peu couteuses et disponibles en milieux urbains denses et péri-urbains. C'est pour cette raison que nous avons choisi d'exploiter les contraintes fournies par un GPS standard et celles apportées par des modèles issus des *Systèmes d'Information Géographique*, plus précisément : des modèles 3D des bâtiments et des modèles d'élévation de terrain.

La principale contribution de ces travaux réside en l'intégration de ces contraintes au sein de l'ajustement de faisceaux (*i.e.* processus d'optimisation du SLAM). Ceci n'est pas trivial étant donné que combiner des contraintes agissant sur la trajectoire de la caméra et la reconstruction 3D peut entrainer des problèmes de convergences, en particulier lorsque les informations exploitées ont des incertitudes variées, voire même des données biaisées ou aberrantes (*e.g.* pour les mesures du GPS). Différentes solutions Larnaout et al. (2012, 2013a,b,c) permettant de combiner plusieurs de ces contraintes simultanément tout en limitant les problèmes de convergence ont été développées.

Les solutions proposées ont été validées sur des séquences de synthèse et d'autres réelles de plusieurs kilomètres enregistrées dans des conditions de circulation normale. Les résultats obtenus montrent que la précision atteinte au niveau de l'estimation des six degrés de liberté de la caméra permet d'assurer des nouvelles applications d'aide à la navigation par le biais de la Réalité Augmentée. En plus de leur précision, nos approches ont l'avantage d'être rapides, peu couteuses et faciles à déployer (ne nécessitant pas un matériel sophistiqué).

Mots clés : Localisation et cartographie simultanées par vision, géolocalisation de véhicule, Système d'Information Géographique, GPS.

Abstract

To ensure a global localization in urban environment, the majority of commercial solutions is based on Global Positioning Systems (GPS). While these systems offer sufficient accuracy in peri-urban or rural areas, their accuracy decreases significantly in cities because of the urban canyon (*i.e.* reflections of the GPS signal through the facades of buildings). To overcome this problem, vision based solutions such as the visual SLAM (Simultaneous Localization And Mapping) seem to be a promising alternative. In addition to the estimation of the six degrees of freedom of the mobile camera, such approach provides a 3D map of the observed scene. However, the localization provided by the visual SLAM is not geo-referenced and is often subject of drifts (*e.g.* poor estimate of the scale factor, accumulations errors).

To address these limitations and to provide a solution easy to deploy, we studied the possibility of integrating to the SLAM algorithm additional information that could constrain the entire reconstruction. These data must then be inexpensive and available in dense urban and peri-urban areas. For these reasons, we chose to exploit the constraints provided by a standard GPS and those provided by models from *Geographic Information Systems*, more precisely, the 3D buildings models and the digital elevation models.

The main contribution of this work lies in the integration of these constraints in the bundle adjustment (*i.e.* the optimization process of the SLAM algorithm). This is not trivial since combining constraints acting on the trajectory of the camera and the 3D reconstruction can lead to convergence problems, especially when the information used have various uncertainties and even outliers (*e.g.* specially GPS measurements). Different solutions Larnaout et al. (2012, 2013a,b,c) to combine these constraints simultaneously while limiting the problems of convergence have been developed.

The proposed solutions have been evaluated on synthetic sequences and large scale real sequences recorded in normal traffic conditions. The results show that the accuracy achieved on the six degrees of freedom of the mobile camera is sufficient to ensure new service of aided navigation through Augmented Reality. In addition to the accuracy, our approaches have the advantage of being fast, inexpensive and easy to deploy (not requiring sophisticated equipment).

Key-words : Simultaneous Localisation and Mapping, vehicle geolocalisation, Geographical Information System, GPS.

Notations et Acronymes

Mathématiques

$\mathbb{E}^n$	Espace E de dimension n
M	Matrice
$\mathbf{v}$	Vecteur
$[\mathbf{t}]_\times$	Matrice antisymétrique créée à partir du vecteur $\mathbf{t}$
M^+	Pseudo-inverse de la matrice M
$\mathcal{F}$	Fonction mathématique

Géométrie euclidienne et projective

$\sim$	Egalité à un facteur non-nul près
$\mathbf{q}$	Point 2D
$\tilde{\mathbf{q}}$	Coordonnées homogènes du point 2D
$\mathcal{Q}$	Point 3D dans le repère monde
$\tilde{\mathcal{Q}}$	Coordonnées homogènes du point 3D
$\mathbf{d}$	Droite de l'espace
$\mathbf{\Pi}$	Plan de l'espace
$\mathcal{R}$	Matrice de rotation
$\mathbf{t}$	Vecteur de translation

Caméras et reconstruction 3D

$\mathcal{C}$	Caméra
$\mathcal{I}$	Image capturée par la caméra
P	Matrice de projection
K	Matrice de calibrage
F	Matrice fondamentale
E	Matrice essentielle
$\mathcal{C}_j$	$j^{\text{ème}}$ caméra reconstruite
$\mathcal{Q}_i$	$i^{\text{ème}}$ point 3D reconstruit
$\mathbf{q}_{i,j}$	Observation du $i^{\text{ème}}$ point 3D par la $j^{\text{ème}}$ caméra

Acronymes

ICP	Iterative Closest Point
MAD	Median Absolute Deviation
SfM	Structure from Motion
SIG	Système d'Information Géographique
MET	Modèle d'Elévation de Terrain
GPS	Global Positioning System
SLAM	Simultaneous Localization and Mapping

Sommaire

Introduction **1**

1 Notions de base et données utilisées **7**
1.1 Caméras perspectives et géométrie associée 7
1.2 Optimisation numérique 12
1.3 Localisation et reconstruction 3D par vision 15
1.4 Algorithmes et données utilisés 23

2 Etat de l'art **33**
2.1 Localisation basée vision sans a priori 33
2.2 Localisation basée vision avec ajout d'informations additionnelles 39
2.3 Bilan 45

3 État de l'art : Ajustement de faisceaux contraint **47**
3.1 Introduction 47
3.2 SLAM contraint à un modèle géométrique de la ville : Contraindre la reconstruction 48
3.3 SLAM contraint aux données GPS : Contraindre la trajectoire 59
3.4 Bilan 62

I Intégration des contraintes fournies par le MET dans un SLAM contraint pour une géo-localisation plus précise sur les 6DoF de la caméra **65**

Présentation des méthodes proposées **69**

4 SLAM contraint au MET et aux modèles 3D des bâtiments pour une localisation en ligne **71**
4.1 SLAM contraint au MET 71
4.2 SLAM contraint au MET et aux modèles 3D des bâtiments 75
4.3 Segmentation du nuage de points 78

4.4 Évaluation expérimentale . . . 82
4.5 Conclusion et perspectives . . . 95

5 SLAM contraint aux données GPS et au MET 97
5.1 Introduction . . . 97
5.2 Fusion de la contrainte aux données GPS avec *une contrainte dure en altitude* . 98
5.3 Fusion de la contrainte aux données GPS avec *une contrainte douce en altitude* 100
5.4 Évaluation expérimentale . . . 102
5.5 Conclusion et perspectives . . . 108

Bilan 109

II Fusion des contraintes apportées par le GPS, le MET et les modèles des bâtiments pour une localisation précise d'une caméra dans un milieu urbain : Application à la Réalité Augmentée 111

Présentation des méthodes proposées 115

6 Fusion hors ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Création d'une base d'amers géo-référencée précise 117
6.1 Introduction . . . 117
6.2 Positionnement et principe de notre approche . . . 118
6.3 Création d'une base d'amers 3D géo-référencée . . . 119
6.4 Évaluation expérimentale . . . 122
6.5 Conclusion et perspectives . . . 128

7 Fusion en ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Correction du biais du GPS 135
7.1 Positionnement et principe de notre approche . . . 135
7.2 GPS différentiel basé sur les modèles 3D des bâtiments . . . 137
7.3 SLAM contraint au MET et aux données du GPS corrigées . . . 142
7.4 Évaluation expérimentale . . . 143
7.5 Conclusion et perspectives . . . 152

8 Applications de Réalité Augmentée en milieu urbain 155
8.1 Introduction . . . 155
8.2 Approche proposée . . . 157
8.3 Navigation en zone disposant d'une base d'amers . . . 157
8.4 Navigation en zone dépourvue de base d'amers et mise à jour de la base . . . 159
8.5 Transition entre zone avec base d'amers vers une zone sans base d'amers valide 160
8.6 Évaluation expérimentale . . . 160
8.7 Discussion . . . 161

Conclusion 167

Annexes 171

A Ajustement de faisceaux contraint **173**
A.1 Ajustement de faisceaux basé sur la fonction de coût standard 173
A.2 Ajustement de faisceaux basé sur une fonction de coût avec une contrainte dure 177
A.3 Ajustement de faisceaux basé sur une fonction de coût avec une contrainte d'inégalité 182

Bibliographie **188**

Table des figures **197**

Liste des tableaux **199**

Liste des algorithmes **201**

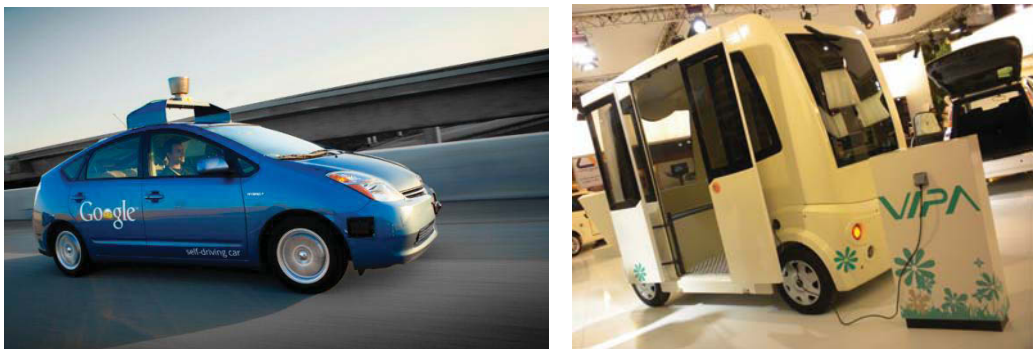
Table des matières **207**

Introduction

Problématique

Actuellement, la problématique de véhicule autonome représente un axe de recherche et développement majeur dans l'industrie de l'automobile. Les recherches effectuées jusqu'à présent ont été motivées par les différents avantages potentiels que pourrait apporter le véhicule autonome dans notre quotidien. En effet, la généralisation d'une telle technologie permettrait de réduire le nombre d'accidents en réduisant notamment le temps de réaction. Elle permettrait également de réduire les embouteillages et l'homogénéisation quasi-instantanée du trafic. Nous pouvons aussi citer la diminution du besoin des parking principalement en centres villes (*i.e.* la voiture peut déposer ses occupants et se garer toute seule en dehors de la ville), livraison automatique *etc.* Malgré les nombreux avantages du véhicule autonome, ce système implique des problématiques importantes de localisation à grande échelle avec des défis en terme de qualité de service (*i.e.* précision et robustesse élevées) et de sa continuité (*i.e.* le véhicule doit fonctionner tout le temps). De nombreuses tentatives isolées ont vu le jour depuis les années 1970. La montée en puissance des technologies liées aux capteurs a propulsé les recherches axées sur les véhicules autonomes (voir figure 1). Nous notons alors la camionnette automatique de Mercedes-Benz en 1984 (le véhicule a atteint 100Km/h sur un réseau routier sans trafic). Le véhicule autonome de Daimler-Benz (1995) était capable de réaliser une conduite en file, un changement de file et un dépassement avec une vitesse de pointe de 175Km/h. La distance maximale parcourue en conduite automatique fut 158Km. Récemment, en 2010, les voitures autonomes développées par Google ont parcouru plus de 800000Km en Californie sans avoir provoqué d'accident. Malgré ce progrès prometteur, les solutions actuelles restent difficilement déployables en raison des coût du matériel nécessaire à leur fonctionnement (*e.g.* caméras, radars, sonars, lidars, GPS différentiel *etc.*).

D'un autre côté, l'apparition de système tel que le système d'affichage à tête haute et des premiers pare-brises augmentés ont créé un besoin de services d'aides à la navigation améliorés à travers la Réalité Augmentée (voir figure 2). Ces services exploitent le flux vidéo fourni par une caméra embarquée sur le véhicule en rajoutant en surimpression des éléments d'aide à la navigation tel que les panneaux routiers, les passages piétons ou les bâtiments. Pour ceci, ces services ont des besoins assez proches du véhicule autonome mais ils sont plus exigeants en termes de facilité de déploiement et plus tolérants aux erreurs de localisation. Il peuvent donc être considérés comme une étape intermédiaire pour le développement et le déploiement



(a) (b)

FIGURE 1 – **Exemples de voitures autonomes.**(a) Véhicule développée par Google, (b) Véhicule VIPA développée par l'Institut Pascal.

(a) (b)

(c)

FIGURE 2 – **Exemples de systèmes d'aide à la navigation la Réalité Augmentée.** (a) Pioneer Cyber Navi, (b) Follow Me et (c) Wikitude Drive.

d'un système de localisation fine à grande échelle. Malheureusement, les solutions actuellement commercialisées par le monde industriel offrant un tel service, tels que *Wikitude Drive* [1], *Follow Me* [2] et *Pioneer Cyber Navi* [3] exploitent principalement l'orientation fournie par une centrale inertielle et la géo-localisation apportée par un GPS dont la précision restent insuffisante en milieux urbains denses. Par conséquent, les éléments ajoutés en surimpression sur le flux vidéo sont mal recalés et flottant, ce qui les rend difficiles à interpréter.

L'objectif de la présente thèse consiste à explorer la possibilité d'une solution basée vision intégrant des données issues de capteurs standard et d'informations disponibles et peu couteuses. Cette solution doit être facilement déployable et doit offrir une qualité de localisation et une robustesse acceptable pour une application d'aide à la navigation en Réalité Augmentée.

Contexte de la thèse

Cette thèse a été effectuée entre janvier 2011 et février 2014 au Laboratoire Vision et Ingénierie des Contenus (LVIC) du CEA LIST, à Saclay et à l'Institut Pascal, à Clermont-Ferrand.

Contributions

L'objectif de nos travaux est de proposer des solutions de localisation par vision dans des milieux urbains qui soient à la fois précises, robustes et faciles à déployer. Pour ceci, nous nous basons sur un algorithme de localisation et cartographie simultanée communément identifié par l'acronyme SLAM. En plus de fournir une estimation de la trajectoire de la caméra, un tel algorithme reconstruit en 3D l'environnement parcouru. Toutefois, en l'état, le processus SLAM est sujet de plusieurs dérives (*e.g.* accumulation d'erreurs et dérive en facteur d'échelle). Pour éviter ces limitations, nous avons étudié la possibilité d'intégrer au SLAM des informations supplémentaires qui pourraient contraindre l'ensemble de la reconstruction fournie. Ces dernières doivent alors être peu couteuses et disponibles en milieux urbains denses et péri-urbains. C'est pour cette raison que nous avons choisi d'exploiter les contraintes fournies par un GPS standard et celles apportées par des modèles issus des *Systèmes d'Information Géographique*, plus précisément : des modèles 3D des bâtiments et des modèles d'élévation de terrain.

La principale contribution de nos travaux de thèse réside en l'intégration de ces contraintes au sein de l'ajustement de faisceaux (*ie* processus d'optimisation du SLAM). Ceci n'est pas trivial étant donné que combiner des contraintes agissant sur la trajectoire de la caméra et la reconstruction 3D peut entrainer des problèmes de convergences, en particulier lorsque les informations exploitées ont des incertitudes variées, des données aberrantes ou biaisées (*eg* pour les mesures du GPS) ou fournit des contraintes contradictoires. Dans la première partie de la thèse, nous nous sommes inspirés des solutions existantes dans la littérature se basant sur des ajustements de faisceaux contraints à savoir les méthodes proposée par Lothe et al. (2009), Tamaazousti et al. (2011) et Lhuillier (2012). Plusieurs améliorations ont été proposées pour donner naissance à deux approches d'estimation des six degrés de liberté d'une caméra mobile soit en milieux urbains denses soit en milieux péri-urbains :

- ▷ **Localisation dans un milieu urbain dense :** Dans les milieux urbains denses, les bâtiments présentent la structure géométrique la plus observée. Il semble alors intuitif d'ex-

1. www.wikitude.org/drive-2
2. www.66.com/US/navigate/follow-me/
3. http://pioneer.jp/press-e/2013/0508-1.html

ploiter les contraintes que les modèles des bâtiments peuvent apporter. Pour ceci, nous améliorons la méthode proposée par Tamaazousti et al. (2011) et nous l'étendons en intégrant une contrainte supplémentaire fournie par le modèle d'élévation de terrain. Tous les degrés de liberté de la reconstruction SLAM sont contraints ce qui a permis l'amélioration de la localisation basée vision dans un environnent urbain dense. Pour améliorer la convergence du processus d'optimisation, nous avons également proposé une nouvelle méthode de segmentation du nuage de points reconstruit qui identifie d'une façon plus précise l'ensemble de points 3D concerné par la contrainte aux bâtiments.

- ▷ **Localisation dans un milieu péri-urbain :** Dans les milieux péri-urbains, la présence des bâtiments est beaucoup moins importante qu'en ville. Les contraintes fournies par de tels modèles ne sont plus désormais suffisantes pour contraindre la reconstruction SLAM. Cependant, le manque de bâtiments implique une précision plus élevée des systèmes GPS. L'exploitation de ce capteur permet ainsi de contraindre la position en deux dimensions de la caméra. Pour améliorer la précision de la localisation sur les degrés de liberté restant, nous proposons d'étendre la fusion SLAM/GPS introduite pas Lhuillier (2012) en lui intégrant les contraintes apportés par le modèle d'élévation de terrain qui est également disponible en milieux péri-urbains.

La deuxième partie de ce mémoire explore la possibilité de combiner les différentes contraintes fournies par le GPS, le modèle d'élévation de terrain et les modèles 3D des bâtiments dans un même processus d'optimisation pour obtenir une solution plus précise et fonctionnant dans un plus vaste domaine d'applications (urbain dense et péri urbain, *i.e.* les lieux où la précision est nécessaire). Nos contributions dans cette partie s'articulent autour de deux manières distinctes pour fusionner les contraintes en question.

- ▷ **Fusion hors ligne des contraintes :** Afin d'éviter tout problème de convergence, nous avons proposé, dans cette partie, une approche *coarse to fine* fonctionnant hors ligne. Une première localisation est obtenue à travers la fusion du SLAM avec le GPS et le modèle d'élévation de terrain. La reconstruction SLAM résultante est, par la suite, raffinée en exploitant les contraintes aux bâtiments et au modèle d'élévation de terrain. Il en résulte une base d'amers géo-référencée précise pouvant être exploitée via un processus de re-localisation en ligne.
- ▷ **Fusion en ligne des contraintes :** Pour la fusion en ligne des contraintes, les bâtiments sont utilisés pour corriger les imprécisions importantes du GPS. Cette correction est réalisée en s'inspirant du principe de fonctionnement du GPS différentiel. En effet en comparant le nuage de points obtenu en ligne à travers une fusion du SLAM/GPS avec les modèles des bâtiments, nous montrerons qu'il est possible d'estimer les imprécisions du GPS. Les mesures GPS, ainsi corrigées, sont intégrées par la suite dans l'ajustement de faisceaux intégrant en plus la contrainte au modèle d'élévation de terrain.
- ▷ **Application à la réalité augmentée** Pour mettre en évidence la précision de nos deux précédentes approches de fusion de contraintes, des exemples d'application d'aide à la navigation basé sur la Réalité Augmentée ont été présentés.

Les travaux réalisés au cours de cette thèse ont donné lieu à plusieurs publications internationales (Larnaout et al. (2012), Larnaout et al. (2013a), Larnaout et al. (2013b), Larnaout et al. (2013c)) et une soumission de brevet.

Organisation du mémoire

En premier lieu, le chapitre 1 présente l'ensemble des notations et des outils de base nécessaires à la bonne compréhension du mémoire. Le chapitre 2 fait un tour d'horizon des méthodes classiques de localisation basée vision puis présente les travaux intégrant à ces approches des informations supplémentaires fournies soit par un capteur additionnel soit via la connaissance a priori de l'environnement parcouru. Un deuxième chapitre dédié à l'état de l'art détaille les méthodes les plus pertinentes dans notre cas d'étude et sur lesquelles se basent nos approches (chapitre 3).

Dans la partie I sont présentées nos deux approches permettant d'estimer les six degrés de liberté d'une caméra mobile dans un milieu urbain dense en exploitant les contraintes aux bâtiments et aux modèles d'élévation de terrain (chapitre 4) et dans un milieu péri-urbain en exploitant les contraintes GPS et aux modèles d'élévation de terrain (chapitre 5). La partie II, quant à elle, détaille les approches proposées pour fusionner les contraintes multi-vues avec les contraintes fournies par le GPS, le modèle d'élévation de terrain et les modèles 3D des bâtiments : le chapitre 6 est consacré à la fusion hors ligne des contraintes tandis que le chapitre 7 assure une fusion en ligne des contraintes en question. Les différentes applications d'aide à la navigation via la Réalité Augmentée sont présentées dans le chapitre 8.

Nous dressons enfin un bilan des travaux réalisés et présentons différentes perspectives envisageables pour les travaux futurs.

CHAPITRE 1

Notions de base et données utilisées

Dans ce chapitre, nous introduisons les notions de base et les notations nécessaires à la compréhension du mémoire. Après avoir présenté les caméras perspectives et géométrie associée, nous rappellerons les plus importantes techniques d'optimisation utilisées. Plus de détails sur ces notions peuvent être trouvées dans le livre de Hartley and Zisserman (2004). Nous décrirons par la suite les méthodes et les données d'entrée de nos travaux, notamment un algorithme de localisation et cartographie simultanées par vision monoculaire (SLAM), les données GPS ainsi que les modèles SIG.

1.1 Caméras perspectives et géométrie associée

1.1.1 Géométrie projective

Sur l'espace vectoriel $\mathbb{R}^{n+1}$, il est possible de définir la relation d'équivalence suivante :

$$\mathbf{u} \sim \mathbf{v} \Longleftrightarrow \exists \lambda \in \mathbb{R}^* / \mathbf{u} = \lambda \mathbf{v}. \tag{1.1}$$

L'ensemble des classes d'équivalence de $\mathbb{R}^{n+1}$ pour cette relation "$\sim$ "définit un espace appelé *espace projectif*. Cet espace, de dimension n, sera noté $\mathbb{P}^n$. Si des études théoriques de ces espaces existent, nous nous intéresserons dans nos travaux à la *géométrie projective* qui leur est associée et qui permet en particulier de formaliser la notion de point à l'infini dans les espaces affines.

Un vecteur de l'espace projectif $\mathbb{P}^n$ aura pour coordonnées :

$$\tilde{\mathbf{x}} = (x_1 \ \dots \ x_{n+1})^\mathsf{T}, \tag{1.2}$$

avec les x_i non tous nuls. Si x_{n+1} est non nul, ce vecteur $\tilde{\mathbf{x}}$ représente le vecteur $\mathbf{x}$ de $\mathbb{R}^n$ avec $\mathbf{x} = (x_1/x_{n+1} \ \dots \ x_n/x_{n+1})^\mathsf{T}$. Dans le cas contraire, le vecteur $\tilde{\mathbf{x}}$ décrit un point à l'infini. Les coordonnées $\tilde{\mathbf{x}}$ sont appelées *coordonnées homogènes* de $\mathbf{x}$. Dans l'ensemble du mémoire, l'utilisation du tilde indiquera que les coordonnées utilisées sont les coordonnées homogènes. L'ensemble des notations sont répertoriées à la page vii. Nous appellerons π la

fonction permettant de passer des coordonnées homogènes aux coordonnées euclidiennes, à savoir :

$$\begin{array}{rccc} \pi & : & \mathbb{P}^n & \rightarrow & \mathbb{R}^n \\ & & (x_1 \ \ldots \ x_{n+1})^\mathsf{T} & \mapsto & (x_1/x_{n+1} \ \ldots \ x_n/x_{n+1})^\mathsf{T} \end{array} . \tag{1.3}$$

1.1.2 Représentation d'une caméra perspective

Dans le cadre de nos travaux, nous utiliserons des caméras perspectives respectant le modèle des *caméras sténopé* idéales. Ce modèle considère que l'ensemble des rayons lumineux passent par un seul et unique point avant d'atteindre le capteur (voir figure 1.1).

Dans la suite, nous présenterons tout d'abord les différents paramètres caractérisant ce type de caméras. Nous présenterons alors la géométrie reliant les images observées par plusieurs caméras. Enfin, nous étudierons les méthodes permettant de retrouver le déplacement de ces caméras ainsi que la structure de l'environnement à partir des images qu'elles observent.

1.1.2.1 Projection perspective

La projection perspective vise à calculer, pour tout point $\mathcal{Q}$ de $\mathbb{R}^3$, la position 2D $\mathbf{q}$ de sa projection dans l'image. Basée sur la projection centrale, cette transformation consiste à calculer l'intersection du plan de la *rétine* de la caméra (*i.e.* le capteur) avec le *rayon de projection* de $\mathcal{Q}$. Ce dernier est défini comme étant la droite reliant le point $\mathcal{Q}$ au centre de la caméra (figure 1.1).

Cette projection peut être vue comme un enchaînement de trois transformations géométriques (figure 1.1) :

- ▷ La première transformation est un changement de repère qui consiste à exprimer les coordonnées de $\mathcal{Q}$ dans le repère lié à la caméra. Ce changement de repère est défini par les *paramètres extrinsèques* de la caméra. Dans la suite du mémoire, en cas d'ambiguité, les exposant $\mathcal{W}$ ou $\mathcal{C}$ indiqueront le repère dans lequel les coordonnées du point 3D sont exprimées (respectivement monde ou caméra).

- ▷ La deuxième transformation est la *projection centrale* du point 3D. Elle revient à passer du point 3D (exprimé dans le repère caméra) au point d'intersection du rayon de projection et du capteur. Les coordonnées 2D du point résultant sont alors exprimées dans le plan de la rétine (en mm).

- ▷ La troisième transformation est un changement de repère 2D qui vise à passer du repère rétine (repère lié à la physique du capteur et où les coordonnées sont exprimées en mm) au repère *image* de la caméra (repère géométrique où les coordonnées sont exprimées en pixels). Cette transformation est définie par les *paramètres intrinsèques* de la caméra.

La projection perspective est donc une transformation projective de $\mathbb{P}^3 \rightarrow \mathbb{P}^2$. En pratique, elle sera représentée par une *matrice de projection* P de dimension (3×4). La projection perspective s'exprime alors par la relation matricielle suivante :

$$\tilde{\mathbf{q}} \sim \mathsf{P}\tilde{\mathcal{Q}}^{\mathcal{W}}. \tag{1.4}$$

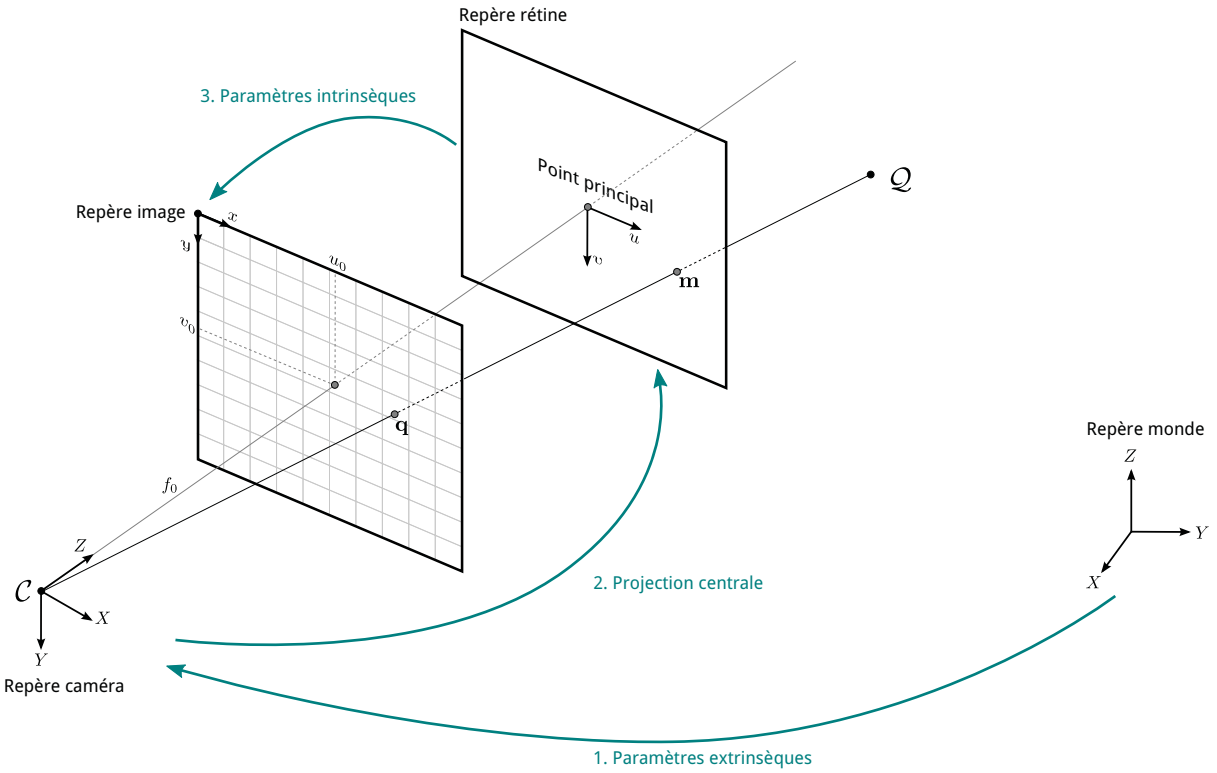


FIGURE 1.1 – **Projection perspective.** La projection perspective peut être vue comme trois transformations géométriques consécutives pour les points 3D.

La matrice P se décompose selon les trois transformations citées précédemment :

$$\mathsf{P} = \mathsf{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathcal{R}^\mathsf{T} & -\mathcal{R}^\mathsf{T}\mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix}, \tag{1.5}$$

où K est la *matrice de calibrage* (de taille 3×3) de la caméra, $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ la matrice de *projection centrale* et $\begin{pmatrix} \mathcal{R}^\mathsf{T} & -\mathcal{R}^\mathsf{T}\mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix}$ la *matrice de pose*.

Paramètres extrinsèques Les paramètres extrinsèques d'une caméra caractérisent la pose de celle-ci dans le repère monde. La pose d'une caméra possède six degrés de liberté :

- ▷ La position 3D du centre optique, décrit par le vecteur $\mathbf{t} = ((\mathbf{t})_x\ (\mathbf{t})_y\ (\mathbf{t})_z)^\mathsf{T}$.
- ▷ L'orientation 3D de la caméra. En pratique, cette orientation sera représentée sous la forme d'une matrice de rotation R, cette matrice pouvant être obtenue à partir des trois angles roulis, tangage et lacet $\mathbf{r} = (\alpha\ \beta\ \gamma)^\mathsf{T}$ (voir figure 1.2).

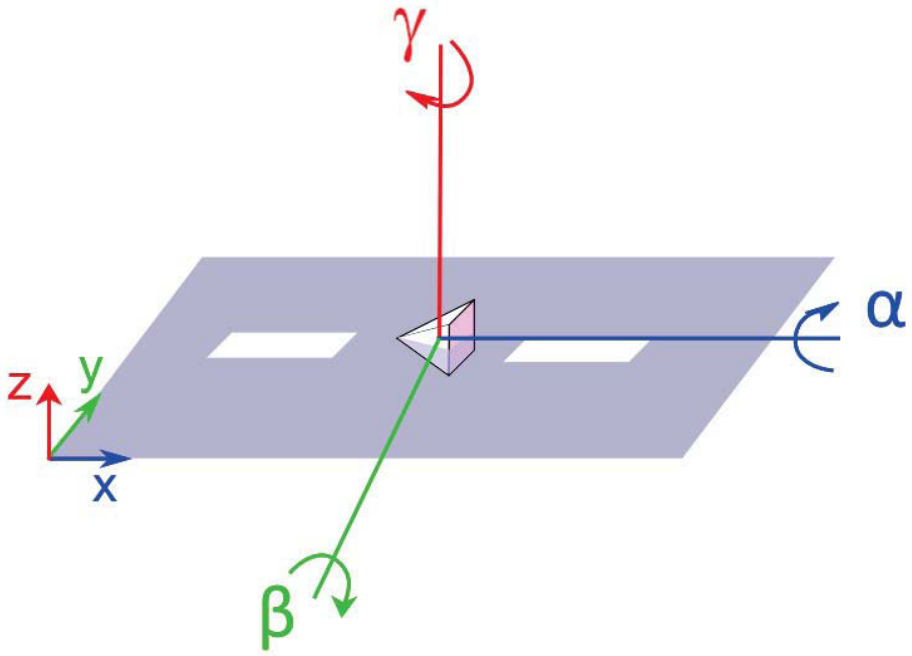


FIGURE 1.2 – **Définition des angles roulis, tangage et lacet.**

Les paramètres extrinsèques de la caméra permettent d'établir les changements de repère monde/caméra, à savoir :

$$\tilde{\mathcal{Q}}^{\mathcal{C}} \sim \begin{pmatrix} \mathcal{R}^{\mathsf{T}} & -\mathcal{R}^{\mathsf{T}}\mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{W}} \tag{1.6}$$

$$\tilde{\mathcal{Q}}^{\mathcal{W}} \sim \begin{pmatrix} \mathcal{R} & \mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{C}} \tag{1.7}$$

Dans le contexte de localisation de véhicule, étant donné que la caméra est rigidement embarquée sur le véhicule à une hauteur fixe par rapport au sol, sa trajectoire appartient théoriquement à un plan parallèle au plan de la route qui représente le plan de déplacement de la caméra. Dans ce cas, les six degrés de liberté de la caméra peuvent être regroupés différemment :

- ▷ Les degrés de liberté dans le plan de déplacement : $((\mathbf{t})_x, (\mathbf{t})_y, \gamma)$. Dans la suite du mémoire nous parlerons de paramètres *dans le plan*.
- ▷ Les degrés de liberté en dehors du plan de déplacement :$((\mathbf{t})_z, \alpha, \beta)$. Dans la suite du mémoire nous parlerons de paramètres *hors plan*.

Projection centrale Lorsqu'on utilise les coordonnées homogènes, la projection centrale d'un point $\mathcal{Q}_{\mathcal{C}}$ est une fonction linéaire de $\mathbb{P}^3 \mapsto \mathbb{P}^2$ caractérisée par la matrice de dimension 3×4 :

$$\tilde{\mathbf{q}} \sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{C}}. \tag{1.8}$$

Dans de nombreuses publications, le changement de repère 3D et la projection centrale sont vus comme une unique projection centrale à partir d'un point 3D dans le repère monde. Cela

s'écrit matriciellement :

$$\begin{aligned}\tilde{\mathbf{q}} &\sim \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}\mathcal{R}^{\mathsf{T}} & -\mathcal{R}^{\mathsf{T}}\mathbf{t}\\ 0_{1\times 3} & 1\end{pmatrix}\tilde{\mathcal{Q}}^{\mathcal{W}}\\ &\sim \begin{pmatrix}\mathcal{R}^{\mathsf{T}} & -\mathcal{R}^{\mathsf{T}}\mathbf{t}\end{pmatrix}\tilde{\mathcal{Q}}^{\mathcal{W}}.\end{aligned} \tag{1.9}$$

La projection d'un point exprimé dans le repère monde dans l'image (équation 1.4) s'écrit donc généralement sous cette forme :

$$\tilde{\mathbf{q}} \sim \mathsf{K}\begin{pmatrix}\mathcal{R}^{\mathsf{T}} & -\mathcal{R}^{\mathsf{T}}\mathbf{t}\end{pmatrix}\tilde{\mathcal{Q}}^{\mathcal{W}}. \tag{1.10}$$

Paramètres intrinsèques et distorsion Les paramètres intrinsèques définissent les propriétés géométriques du capteur de la caméra. Dans notre étude, nous considérons que les pixels sont carrés. La *matrice de calibrage* K peut alors s'exprimer sous la forme :

$$\mathsf{K} = \begin{pmatrix}f_0 & 0 & u_0\\ 0 & f_0 & v_0\\ 0 & 0 & 1\end{pmatrix}. \tag{1.11}$$

Nous retrouvons dans la matrice de calibrage les différents paramètres intrinsèques, à savoir :

- ▷ f_0 la *distance focale*. Exprimée en pixel par unité de mesure, elle décrit la distance orthogonale entre le centre et la rétine de la caméra.
- ▷ $(u_0\ v_0)^{\mathsf{T}}$ le *point principal*. Souvent approximé comme étant le centre du capteur, il est plus précisément l'intersection entre l'axe optique et la rétine de la caméra (figure 1.1).

Il est important de noter que les capteurs à courte focale peuvent présenter un phénomène de distorsion important. Ceci se traduit visuellement par une déformation des lignes droites dans l'image sous forme de courbes. Pour corriger cela, il est possible d'ajouter au calibrage de la caméra des paramètres de distorsion permettant de passer de la position observée d'un point 2D dans l'image à sa position réelle, c'est-à-dire corrigée de toute distorsion.

Dans le cadre de ce mémoire, nous considérerons à la fois que la matrice de calibrage est connue et que les entrées de nos algorithmes ont été préalablement corrigées en distorsion. En pratique, la distorsion radiale est modélisée en utilisant 5 coefficients. La distorsion tangentielle étant beaucoup plus faible, elle sera négligée dans nos travaux. Pour plus de renseignements sur le calibrage des caméras, nous invitons le lecteur à se référer à l'article de Lavest et al. (1998).

1.1.2.2 Notion de rétroprojection

La *rétroprojection* peut être vue comme l'opération inverse de la projection. Son but est d'inférer la position d'un point 3D $\mathcal{Q}$ à partir de son observation $\mathbf{q}$ dans l'image. Néanmoins, à partir d'un seule image, il est impossible d'obtenir la position exacte du point 3D. En effet, l'utilisation d'une seule caméra ne permet pas de retrouver la profondeur à laquelle se situe ce point. La rétroprojection d'un point de l'image se traduit sous la forme du rayon optique qui passe à la fois par le centre de la caméra $\mathcal{C}$ et par l'observation $\mathbf{q}$. La position du point 3D est donc exprimée à un facteur λ près qui reflète la profondeur du point sur ce rayon :

$$\tilde{\mathcal{Q}}(\lambda) \sim \mathsf{P}^{+}\tilde{\mathbf{q}} + \lambda\tilde{\mathcal{C}}, \tag{1.12}$$

où P^+ désigne la pseudo-inverse de la matrice P :

$$\mathsf{P}^+ \;=\; \mathsf{P}^\mathsf{T}(\tilde{\mathsf{P}}\mathsf{P}^\mathsf{T})^{-1}. \tag{1.13}$$

1.2 Optimisation numérique

Dans cette section, nous introduisons les notions et méthodes mathématiques relatives à la résolution des problèmes numériques rencontrés dans la plupart des problèmes de vision et en particulier dans nos travaux. La vocation de cette section n'est pas de détailler les théories mathématiques sous-jacentes mais de faire un tour d'horizon des méthodes utiles. Après avoir détaillé le cadre d'étude de cette section, nous présenterons successivement les méthodes de résolution linéaires et non-linéaires. Nous finirons en présentant différentes approches permettant d'améliorer la robustesse de ces méthodes.

1.2.1 Moindres carrés

En vision par ordinateur en particulier, le problème à résoudre peut souvent être vu comme la recherche d'un ensemble de paramètres $\widehat{\mathbf{x}} = \{\widehat{x}_1, \ldots, \widehat{x}_K\}$ tel que :

$$\mathcal{F}(\widehat{\mathbf{x}}) = \mathbf{y}, \tag{1.14}$$

où $\mathcal{F}$ est la fonction modélisant le problème étudié et $\mathbf{y} = \{y_1, \ldots, y_M\}$ un ensemble de mesures connues. Dans la pratique, cette égalité stricte ne peut pas être obtenue. Ceci est dû aux erreurs de mesure et de calcul numérique par exemple. On définit dans ce cas l'*erreur résiduelle* comme étant la différence entre les mesures et le modèle appliqué aux paramètres estimés :

$$\mathbf{r}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) - \mathbf{y}. \tag{1.15}$$

L'approche couramment utilisée pour résoudre le problème posé est alors la *méthode des moindres carrés*. Cela revient à trouver les paramètres qui vérifient l'équation :

$$\widehat{\mathbf{x}} = \underset{\mathbf{x}}{argmin}\, \varepsilon(\mathbf{x}), \tag{1.16}$$

où la *fonction de coût* ε à minimiser est :

$$\begin{aligned} \varepsilon(\mathbf{x}) &= ||\mathcal{F}(\mathbf{x}) - \mathbf{y}||^2 \\ &= ||\mathbf{r}(\mathbf{x})||^2 \\ &= \sum_i ||r_i(\mathbf{x})||^2. \end{aligned} \tag{1.17}$$

En particulier, dans le cas où la distribution des erreurs est gaussienne, l'estimation aux moindres carrés correspond à l'estimation du maximum de vraisemblance. Dans ce cas, la solution obtenue est optimale au sens statistique du terme.

L'approche utilisée pour résoudre ce problème dépend alors de la linéarité de la fonction $\mathcal{F}$.

1.2.2 Méthodes de résolution linéaires

Lorsque la fonction $\mathcal{F}$ est linéaire, il existe une matrice F telle que pour tout $\mathbf{x}$, $\mathcal{F}(\mathbf{x}) = \mathsf{F}\mathbf{x}$. Deux cas de figure sont alors à différencier.

Système linéaire homogène. Lorsque le vecteur des mesures $\mathbf{y}$ est nul, on parle de *système homogène*. La fonction de coût s'écrit alors :

$$\varepsilon(\mathbf{x}) = ||\mathsf{F}\mathbf{x}||^2. \tag{1.18}$$

La solution aux moindres carrés de cette équation, si on prend la contrainte que $||\mathbf{x}|| = 1$, correspond au vecteur propre associé à la plus petite valeur propre de la matrice F. Ce vecteur propre peut être facilement obtenu en utilisant la décomposition SVD de F.

Système linéaire non-homogène. Dans le cas où le vecteur des mesures est non-nul, la fonction de coût est de la forme :

$$\varepsilon(\mathbf{x}) = ||\mathsf{F}\mathbf{x} - \mathbf{y}||^2. \tag{1.19}$$

Dans ce cas, la solution au sens des moindres carrés peut être obtenue à l'aide de la pseudo-inverse de la matrice F (voir l'équation 1.13) :

$$\widehat{\mathbf{x}} = \mathsf{F}^{+}\mathbf{y}. \tag{1.20}$$

1.2.3 Méthodes de résolution non-linéaires

Lorsque la fonction ε est non-linéaire, il est possible de résoudre le problème posé en utilisant une méthode itérative. L'hypothèse faite par cette famille de méthodes est que la fonction ε est localement linéaire. Le principe est alors de trouver la *direction* et la *longueur de pas* (c'est-à-dire la distance à parcourir dans cette direction), dans l'espace des paramètres, qui permet de diminuer au mieux l'erreur résiduelle. Les paramètres sont alors modifiés à l'aide de l'incrément ainsi calculé et le processus est réitéré depuis la nouvelle valeur des paramètres.

Chacune des méthodes de résolution non-linéaire se distingue sur ces notions d'optimalité concernant la direction à choisir. Voici les méthodes principalement utilisées en vision par ordinateur :

- ▷ **Descente de gradient.** La descente de gradient est une méthode de résolution du premier ordre. La direction de déplacement choisie est directement liée au gradient de la fonction étudiée. L'avantage de cette approche est qu'elle converge efficacement lorsque la solution initiale est éloignée du minimum recherché. Toutefois, elle peut s'avérer extrêmement lente.

- ▷ **Gauss-Newton.** La méthode de Gauss-Newton est une méthode du second ordre. Elle s'appuie principalement sur la dérivée seconde de la fonction afin d'obtenir la direction à chaque itération. Plus sensible à la condition initiale que la descente de gradient, elle assure néanmoins une convergence plus efficace lorsque les paramètres sont proches de la solution.

- ▷ **Levenberg-Marquardt.** La méthode d'optimisation non-linéaire de Levenberg-Marquardt (Levenberg (1944)) est la méthode la plus couramment utilisée pour les problèmes rencontrés dans ce mémoire. L'idée sous-jacente à cette méthode est de combiner les deux approches précédemment citées afin de profiter de leur avantage respectif. Ainsi, lorsque la solution est éloignée, c'est l'algorithme de descente de gradient qui sera privilégié. En se rapprochant de la solution, c'est la méthode de Gauss-Newton qui sera prépondérante afin d'accélérer la convergence.

Il est important de noter que les méthodes présentées ci-dessus n'assurent pas la convergence vers le minimum global de la fonction ε. En effet, ces méthodes itératives sont particulièrement sensibles aux minima locaux. Cela implique que la condition initiale (c'est-à-dire le jeu de paramètres initial) doit être aussi proche que possible de la solution recherchée.

1.2.4 Optimisation robuste

Les méthodes de résolution numérique ci-avant ont été présentées dans le cadre où les différentes données sont supposées correctes, c'est-à-dire que la distribution des erreurs est gaussienne. Dans la pratique, de nombreuses mesures peuvent être erronées : on parle alors de *données aberrantes* (ou *outliers* en anglais). L'apparition de données aberrantes est généralement due au fait que les données mesurées ne suivent pas la modélisation du problème étudié. On peut par exemple penser à une mauvaise association de points d'intérêt lors de l'estimation de la matrice essentielle, ou à la présence d'un point qui ne se situe pas sur le plan 3D Π lors de l'estimation d'une homographie 2D.

Afin d'être robuste à ces données aberrantes, une des solutions la plus utilisée est les M-estimateurs.

Comme nous l'avons mentionné, la résolution d'un problème aux moindres carrés s'écrit sous la forme $\varepsilon(\mathbf{x}) = \sum_i ||r_i(\mathbf{x})||^2$. La figure 1.3(a) montre que, dans la fonction ε, la contribution de chacun des résidus est quadratique. Cela implique que plus un point sera aberrant (et donc plus son résidu sera important), plus son influence dans la fonction de coût sera grande. Pour éviter cela, il est possible de pondérer les résidus avec un estimateur robuste, dans notre cas un *M-estimateur* ρ, dont le but est de réduire l'influence des points aberrants. La fonction de coût se réécrit alors :

$$\varepsilon(\mathbf{x}) = \sum_i \rho(r_i(\mathbf{x}), c). \tag{1.21}$$

De nombreux M-estimateurs ont été proposés dans la littérature (Huber (1981)). Les trois que nous allons présenter ici ont été choisis car ils sont courants en vision par ordinateur et ils présentent tous les trois une gestion différente des résidus aberrants :

▷ Le M-estimateur de Tukey (figure 1.3(b)) rend la valeur des résidus constante pour tous les points aberrants :

$$\rho_{Tukey}(x) = \begin{cases} \dfrac{c^2}{2}\left(1 - \left[1 - \left(\dfrac{x}{c}\right)^2\right]^3\right) & \text{si } |x| \leq c \\ \dfrac{c^2}{6} & \text{sinon} \end{cases} \tag{1.22}$$

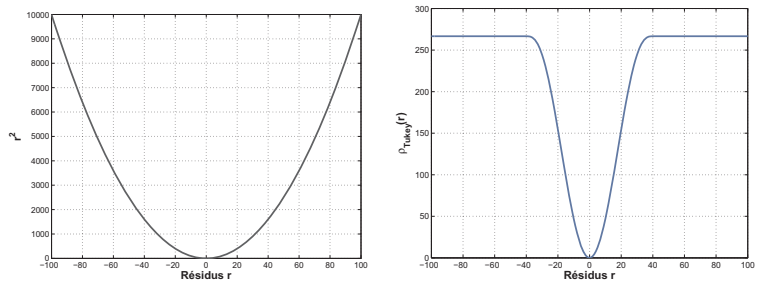
▷ Avec le M-estimateur de Huber (figure 1.3(c)), l'évolution des résidus des points aberrants est linéaire :

$$\rho_{Huber}(x) = \begin{cases} \dfrac{x^2}{2} & \text{si } |x| \leq c \\ c(|x| - \dfrac{c}{2}) & \text{sinon} \end{cases} \tag{1.23}$$

▷ Enfin, le M-estimateur de Geman-McClure (figure 1.3(d)) donne une évolution asymptotique aux valeurs des résidus des points aberrants :

$$\rho_{Geman}(x) = \frac{x^2}{x^2 + c^2}. \tag{1.24}$$

Pour tous les M-estimateurs, il est nécessaire de régler le seuil c qui correspond à la valeur de résidu à partir de laquelle les points sont considérés comme étant aberrants. S'il est possible dans certains problèmes de fixer ce seuil à une valeur précise, il est intéressant de pouvoir estimer automatiquement cette valeur à partir des résidus mesurés. En particulier, la *médiane des écarts absolus à la médiane* (notée MAD dans le mémoire pour *Median Absolute Deviation*) permet d'estimer ce seuil dans les cas où la distribution des résidus étudiés peut être assimilée à une distribution gaussienne (Malis and Marchand (2006)).



(a) (b)

(c) (d)

FIGURE 1.3 – **Exemples de M-estimateurs.** (a) représente la contribution quadratique des erreurs. Les 3 autres figures représentent ce que devient cette contribution en utilisant les M-estimateurs (b) de Tukey, (c) de Huber et (d) de Geman-McClure pour $c = 40$.

1.3 Localisation et reconstruction 3D par vision

Lorsqu'une même scène est observée par plusieurs vues, il est possible d'estimer le déplacement relatif entre les différentes caméras et de calculer la géométrie 3D de l'environnement observé. Ce cas de figure peut apparaître dans différentes configurations :

▷ **Configuration spatiale.** Cette configuration correspond au cas où plusieurs caméras observent simultanément une même scène à partir de différents points de vue.

▷ **Configuration temporelle.** Dans ce cas, une seule caméra se déplace dans l'environnement. L'ensemble des vues correspond alors aux points de vue de la caméra capturés à des instants différents.

Dans le cadre de ce mémoire, nous nous intéresserons à la configuration temporelle. Néanmoins, il est important de noter que ces deux configurations, dans le cas d'une scène rigide, sont équivalentes et peuvent être traitées de façon identique.

Cette partie se consacrera à l'étude de la géométrie entre deux vues. Des méthodes complémentaires sur 3 et N vues peuvent être trouvées dans le livre de Hartley and Zisserman (2004).

1.3.1 Géométrie épipolaire

La *géométrie épipolaire* décrit les contraintes reliant les observations d'une même scène observée par deux caméras, notées $\mathcal{C}_1$ et $\mathcal{C}_2$ (figure 1.4). Ces contraintes sont directement liées au déplacement relatif (également appelé positionnement relatif) entre les deux caméras mais sont totalement indépendantes de la structure de la scène. Toutefois, il important de rappeler que dans le cas du déplacement d'une caméra (*i.e.* dans le cas de la configuration temporelle), la géométrie épipolaire est uniquement vérifiée si la scène observée est rigide.

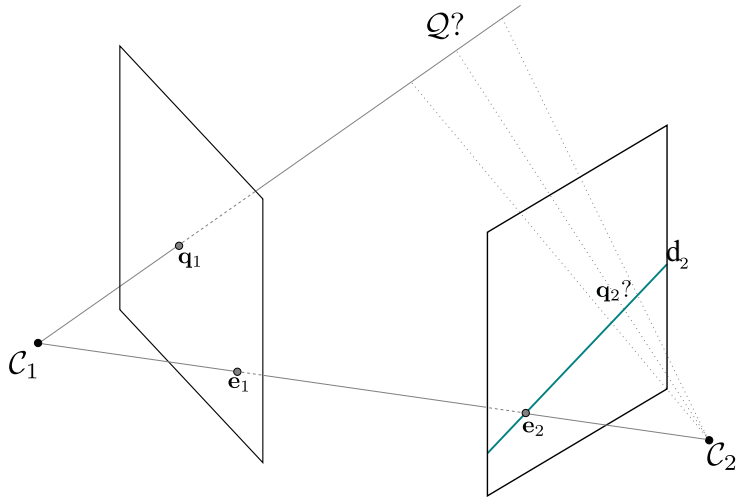


FIGURE 1.4 – **Géométrie épipolaire.** La géométrie épipolaire définit des contraintes géométriques entre les différentes observations d'un même point de l'espace.

Matrice fondamentale La *matrice fondamentale* exprime la relation épipolaire dans le cas où les paramètres internes des caméras sont inconnus. Ainsi, pour un point $\mathbf{q}_1$ de l'image de la première caméra, il est possible de calculer la droite $\mathbf{d}_2$ sur laquelle se situe l'observation correspondante dans la deuxième caméra (figure 1.4) :

$$\mathbf{d}_2 \sim \mathsf{F}\tilde{\mathbf{q}}_1. \tag{1.25}$$

La droite $\mathbf{d}_2$ est appelée *droite épipolaire* associée à $\mathbf{q}_1$. De plus, si deux observations $\mathbf{q}_1$ et $\mathbf{q}_2$ correspondent au même point de l'espace, elles vérifient :

$$\tilde{\mathbf{q}}_2{}^{\mathsf{T}}\mathsf{F}\tilde{\mathbf{q}}_1 = 0. \tag{1.26}$$

Cette relation permet d'estimer la matrice fondamentale à partir d'associations 2D entre deux images. En pratique, F peut se calculer à l'aide de 8 points (Hartley (1997)) ou à partir de 7 points sous certaines hypothèses (Torr and Murray (1997)).

Dans chacune des images, un point joue un rôle particulier. Il s'agit des deux *épipôles* $\mathbf{e}_1$ et $\mathbf{e}_2$. Ils correspondent à la projection dans l'image du centre optique de l'autre caméra. Les épipôles présentent deux caractéristiques intéressantes. Tout d'abord, elles définissent le noyau de F : $\mathsf{F}\tilde{\mathbf{e}}_i = 0, \forall i \in \{1,2\}$. De plus, les épipôles correspondent aux points d'intersection de toutes les droites épipolaires de chacune des images.

Matrice essentielle La *matrice essentielle* E peut être vue comme le cas particulier de la matrice fondamentale dans le cas où le calibrage des caméras (K_1 et K_2) est connu, ce qui est le cas qui nous intéresse en particulier. La relation entre matrice essentielle et matrice fondamentale est la suivante :

$$\mathsf{E} \sim {\mathsf{K}_2}^{\mathsf{T}} \mathsf{F} \mathsf{K}_1. \tag{1.27}$$

L'équation 1.26 devient dans ce cas :

$$\tilde{\mathbf{q}}_2{}^{\mathsf{T}} \left({\mathsf{K}_2}^{-\mathsf{T}} \mathsf{E} {\mathsf{K}_1}^{-1} \right) \tilde{\mathbf{q}}_1 = 0, \tag{1.28}$$

où ${\mathsf{K}_2}^{-\mathsf{T}}$ est la transposée inverse de K_2. Pour estimer la matrice essentielle, Nister (2004) a proposé un algorithme efficace appelé *algorithme des 5 points*.

Relation entre matrice essentielle et déplacement relatif En fonction des cas d'étude, la matrice essentielle peut avoir différentes utilisations. En effet, il existe une relation qui lie la matrice essentielle du couple de caméras ($\mathcal{C}_1$, $\mathcal{C}_2$) au déplacement relatif entre ces caméras. Le déplacement relatif est défini par le couple $(\mathcal{R}_{1\to2}, \mathbf{t}_{1\to2})$. Une formalisation en sera faite à la section 1.3.2.1. La relation entre E, $\mathcal{R}_{1\to2}$ et $\mathbf{t}_{1\to2}$ s'écrit :

$$\mathsf{E} = [\mathbf{t}_{1\to2}]_\times \mathcal{R}_{1\to2}, \tag{1.29}$$

où $[\mathbf{t}]_\times$ est la matrice antisymétrique construite à partir du vecteur $\mathbf{t}$, à savoir :

$$[\mathbf{t}]_\times = \begin{pmatrix} 0 & -(\mathbf{t})_z & (\mathbf{t})_y \\ (\mathbf{t})_z & 0 & -(\mathbf{t})_x \\ -(\mathbf{t})_y & (\mathbf{t})_x & 0 \end{pmatrix}. \tag{1.30}$$

Dès lors, deux cas de figure sont possibles. Si le déplacement entre les caméras est connu, la matrice essentielle permet de réduire la recherche de point d'intérêt correspondant à 1 dimension (le long de la droite épipolaire). Dans le cas contraire, une estimation de la matrice essentielle (grâce à l'appariement d'au moins 5 points) permet de retrouver le déplacement relatif entre les caméras. Cette notion sera développée dans la section 1.3.2.2.

Homographies 2D Dans le cas où la scène observée est planaire, la relation qui existe entre deux caméras est définie par une *homographie* H (ou *transformation projective*) 2D qui représente une transformation linéaire inversible de $\mathbb{P}^2$ dans $\mathbb{P}^2$ qui conserve l'alignement. Le théorème suivant permet de caractériser de façon matricielle les homographies 2D :

Théorème 1 *Une fonction $H : \mathbb{P}^2 \to \mathbb{P}^2$ est une homographie si et seulement si il existe une matrice H de taille 3×3 telle que pour tout point $\tilde{\mathbf{q}}$ de $\mathbb{P}^2$, $H(\tilde{\mathbf{q}}) = \mathsf{H}\tilde{\mathbf{q}}$.*

La matrice H est homogène : elle est définie à un facteur près et possède donc 8 degrés de liberté.

Un des cas courants d'utilisation des homographies 2D est celui décrit dans la figure 1.5. Nous nous plaçons ici dans le cas de deux caméras observant un plan $\mathbf{\Pi}$ de l'espace. La projection centrale du plan de l'espace au plan image (et réciproquement) définit une homographie 2D (les coordonnées des points 2D étant exprimées dans le repère 2D relatif à chacun des plans). Un résultat intéressant est alors que, pour tout point 3D $\mathcal{Q}$ appartenant au plan $\mathbf{\Pi}$, la fonction passant des coordonnées de son obervation $\mathbf{q}_1$ dans l'image 1 aux coordonnées de son observation $\mathbf{q}_2$ dans l'image 2 est également une homographie. En effet, la composition de 2 homographies est une homographie. Le lien entre les observations peut donc s'écrire :

$$\begin{aligned} \tilde{\mathbf{q}}_2 &\sim \mathsf{H}_{1\to 2}\tilde{\mathbf{q}}_1 \\ &\sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tilde{\mathbf{q}}_1. \end{aligned} \tag{1.31}$$

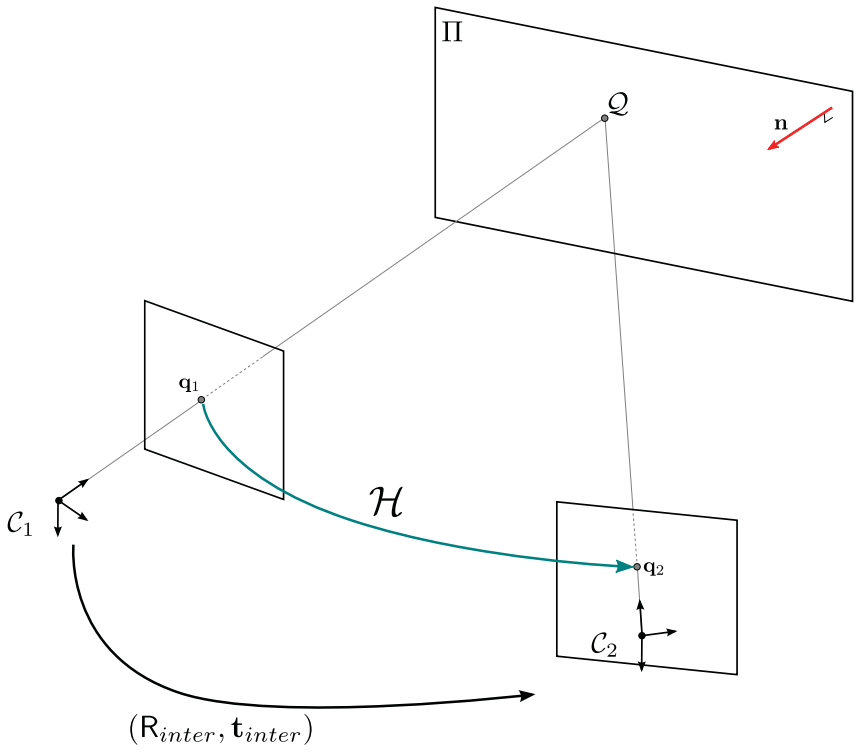


FIGURE 1.5 – **Homographies 2D.** Les coordonnées des observations correspondantes de points 3D situés sur un même plan de l'espace sont reliées par une homographie 2D.

1.3.2 Calcul de la géométrie de l'environnement

Dans cette section, nous allons présenter l'ensemble des outils mathématiques élémentaires qui permettent de calculer la géométrie d'une scène 3D, à savoir la pose des différentes caméras ainsi que le nuage de points 3D associés aux points d'intérêt observés.

1.3.2.1 Poses de caméras et déplacement relatif

Le but de cette section est de formaliser la notion de *déplacement relatif* entre deux caméras ainsi que les notations associées. Comme nous l'avons vu précédemment, la pose des caméras peut être vue comme un changement de repère entre le repère monde et les repères attachés aux caméras :

$$\tilde{\mathcal{Q}}^{\mathcal{C}_1} \sim \begin{pmatrix} {\mathcal{R}_1}^\mathsf{T} & -{\mathcal{R}_1}^\mathsf{T}\mathbf{t_1} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{W}} \tag{1.32}$$

$$\tilde{\mathcal{Q}}^{\mathcal{C}_2} \sim \begin{pmatrix} {\mathcal{R}_2}^\mathsf{T} & -{\mathcal{R}_2}^\mathsf{T}\mathbf{t_2} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{W}} \tag{1.33}$$

avec $\mathcal{Q}^{\mathcal{C}_1}$ et $\mathcal{Q}^{\mathcal{C}_2}$ les coordonnées de $\mathcal{Q}$ respectivement dans les repères liés aux caméras $\mathcal{C}_1$ et $\mathcal{C}_2$ et $\mathcal{Q}^{\mathcal{W}}$ ce même point exprimé dans le repère monde. Afin de fixer les notations, nous appellerons $(\mathcal{R}_{1\to 2}, \mathbf{t}_{1\to 2})$ le déplacement relatif entre les caméras, c'est-à-dire la transformation permettant de passer du repère lié à $\mathcal{C}_1$ à celui lié à $\mathcal{C}_2$:

$$\tilde{\mathcal{Q}}^{\mathcal{C}_2} \sim \begin{pmatrix} \mathcal{R}_{1\to 2} & \mathbf{t}_{1\to 2} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{C}_1}. \tag{1.34}$$

Des équations 1.32 et 1.33, on peut obtenir le système d'équations suivant :

$$\begin{cases} \tilde{\mathcal{Q}}^{\mathcal{W}} \sim \begin{pmatrix} \mathcal{R}_1 & \mathbf{t_1} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{C}_1} \\ \\ \tilde{\mathcal{Q}}^{\mathcal{C}_2} \sim \begin{pmatrix} {\mathcal{R}_2}^\mathsf{T} & -{\mathcal{R}_2}^\mathsf{T}\mathbf{t_2} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{W}} \end{cases} \tag{1.35}$$

d'où

$$\tilde{\mathcal{Q}}^{\mathcal{C}_2} \sim \begin{pmatrix} {\mathcal{R}_2}^\mathsf{T}\mathsf{R}_1 & {\mathcal{R}_2}^\mathsf{T}(\mathbf{t}_1 - \mathbf{t}_2) \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}^{\mathcal{C}_1}. \tag{1.36}$$

Le déplacement relatif entre les caméras est donc défini comme suit :

$$\begin{cases} \mathcal{R}_{1\to 2} = {\mathcal{R}_2}^\mathsf{T}\mathcal{R}_1 \\ \\ \mathbf{t}_{1\to 2} = {\mathcal{R}_2}^\mathsf{T}(\mathbf{t}_1 - \mathbf{t}_2) \end{cases} \tag{1.37}$$

1.3.2.2 Calcul du déplacement relatif par associations 2D/2D

Lorsque la structure de l'environnement est inconnu, il est tout de même possible de calculer le déplacement relatif entre deux caméras. Cela nécessite d'associer les observations des 2 caméras qui correspondent aux mêmes points 3D de l'environnement. Comme nous l'avons vu précédemment (section 1.3.1), ceci permet de calculer la matrice fondamentale (algorithme des 8 points, Hartley (1997)) ou essentielle (algorithme des 5 points, Nister (2004)). Il est alors possible d'extraire d'une de ces matrices le déplacement inter-caméra $(\mathcal{R}_{1\to 2}, \mathbf{t}_{1\to 2})$.

Dans le cas de caméras non-calibrées, $\mathcal{R}_{1\to 2}$ et $\mathbf{t}_{1\to 2}$ sont calculés à partir de la matrice fondamentale (Hartley and Zisserman (2004)). Dans ce cas, le déplacement inter-caméra ne peut être retrouvé qu'à une transformation projective près. En particulier, ceci induit qu'il est impossible de retrouver les rapports de distance et les angles.

Le calibrage des caméras étant connu dans notre étude, il est préférable d'utiliser la matrice essentielle. La décomposition en valeurs singulières SVD (Faugeras (1993)) de celle-ci permet en effet d'en extraire 4 couples solution possibles pour $\mathcal{R}_{1\to2}$ et $\mathbf{t}_{1\to2}$. Parmi ces 4 couples, on retient le couple permettant de reconstruire les 5 points ayant servi au calcul de E devant les 2 caméras . Le détail de cette décomposition peut être trouvé dans l'article de Nister (2004).

Dans le cas calibré, le déplacement relatif entre les 2 caméras (et donc toute la structure 3D sous-jacente) est défini à un facteur près. En effet, dans le cas du calcul du déplacement par associations 2D/2D, le facteur d'échelle de la scène (c'est-à-dire sa métrique) n'est pas observable. En pratique, cette échelle est donc fixée arbitrairement.

Notons également que seul le déplacement relatif est défini mais pas la pose des caméras dans le repère monde. En effet, aucune information de localisation absolue n'est fournie de sorte que les deux caméras obtenues sont positionnées à une rotation et une translation près dans le monde. Ainsi, si le déplacement relatif est défini à un facteur près, la pose absolue des caméras est définie à 7 degrés près. Une transformation 3D possédant ces 7 degrés de liberté est appelée *similitude* et peut être représentée par la matrice homogène suivante :

$$\mathsf{S} \sim \begin{pmatrix} s\mathcal{R} & \mathbf{t} \\ 0_{1\times3} & 1 \end{pmatrix}, \tag{1.38}$$

avec s le facteur d'échelle, $\mathcal{R}$ la rotation et $\mathbf{t}$ la translation.

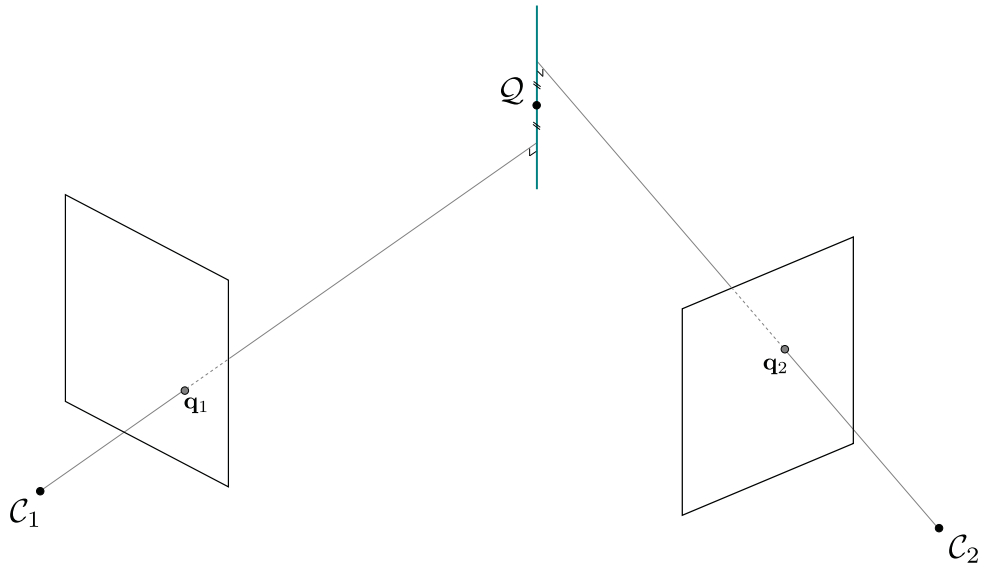


FIGURE 1.6 – **Triangulation de points 3D.** La structure de l'environnement peut être obtenue par triangulation des observations dans les images.

1.3.2.3 Calcul de la structure de l'environnement

Nous avons vu que la rétroprojection d'une observation 2D d'un point de l'espace permet d'obtenir sa position 3D à la profondeur près (section 1.1.2.2). Dès lors qu'au moins 2 caméras dont la pose et le calibrage sont connus observent ce point, la profondeur du point peut être estimée. On parle alors de *triangulation* du point. L'idée de la triangulation est de calculer l'intersection des rayons optiques issus des 2 observations. En pratique, à cause des bruits sur les

différentes données (calibrage, pose des caméras, position des observations, *etc.*), les rayons ne s'intersectent pas. Dans le cas de 2 caméras, le résultat de la triangulation est le point équidistant des deux rayons (figure 1.6).

Dans un but de robustesse et de précision des calculs numériques, la notion de triangulation peut être généralisée à plus de 2 caméras. Par exemple, dans le cas de 3 caméras, il est possible de calculer 3 triangulations différentes à partir des couples de caméras (1,2), (2,3) et (1,3). Le résultat final de la triangulation est alors le barycentre de ces 3 points. Il existe également une approche linéaire permettant de trianguler un point observé par N-vues en utilisant la méthode DLT (Hartley and Zisserman (2004)).

1.3.2.4 Calcul de pose par associations 2D/3D

Une fois la structure de l'environnement partiellement connue, il est possible de calculer la pose d'une caméra tiers à partir d'associations réalisées entre les observations 2D de son image et la position 3D de 3 points de l'environnement. De nombreuses méthodes ont été proposées pour résoudre ce problème. Une comparaison de certaines de ces méthodes peut être trouvée dans l'article de Haralick et al. (1994). Plus récemment, Lepetit et al. (2009) ont proposé une nouvelle approche plus performante (en temps de calcul et en précision) du calcul de pose.

L'utilisation d'associations 2D/3D plutôt que 2D/2D présente plusieurs avantages. Tout d'abord, il est à noter que le calcul de pose 2D/3D est beaucoup plus rapide que le calcul de pose 2D/2D (l'estimation de la matrice essentielle étant une étape coûteuse). De plus, nous avons vu précédemment que l'extraction des paramètres à partir de la matrice essentielle ne permet pas d'estimer le facteur d'échelle et donc en particulier la norme de la translation entre les différentes caméras. Avec l'approche 2D/3D, le facteur d'échelle peut être estimé à partir de l'observation de la distance entre les différents points de l'espace. Enfin, Tardif et al. (2008) ont montré que l'utilisation de l'approche 2D/3D offre un calcul plus précis de la position de la caméra.

1.3.2.5 Erreur de reprojection et ajustement de faisceaux

Lorsqu'un ensemble de points 3D et de caméras sont reconstruits à l'aide des méthodes définies précédemment, il est nécessaire de définir une erreur permettant de mesurer la qualité de cette reconstruction. L'idée principale de cette erreur est de mesurer la distance entre l'endroit où le point est détecté dans l'image et sa position estimée. Si des erreurs 3D ont été proposées (par exemple mesurer la distance entre le rayon optique issu de l'observation et le point 3D), il a été montré qu'il est généralement préférable d'utiliser une erreur 2D (Lu et al. (2000)), en particulier pour éviter que les points 3D au loin aient une erreur plus importante du fait de leur profondeur.

La solution couramment retenue est *l'erreur de reprojection* (figure 1.7). Elle consiste à mesurer la distance 2D entre l'observation du point 3D dans l'image (c'est-à-dire la position 2D du point d'intérêt) et la projection du point 3D reconstruit dans cette même image :

$$\mathbf{f} = ||\mathbf{q} - \pi(\mathsf{P}\tilde{\mathcal{Q}})||. \tag{1.39}$$

Les méthodes de calcul de pose des caméras et de la structure de l'environnement telles qu'elles ont été présentées précédemment ne fournissent pas une solution optimale au problème de reconstruction et localisation simultanées. Pour corriger cela, il est possible de raffiner l'ensemble des paramètres de la scène (à savoir les six paramètres de pose de chaque caméra et

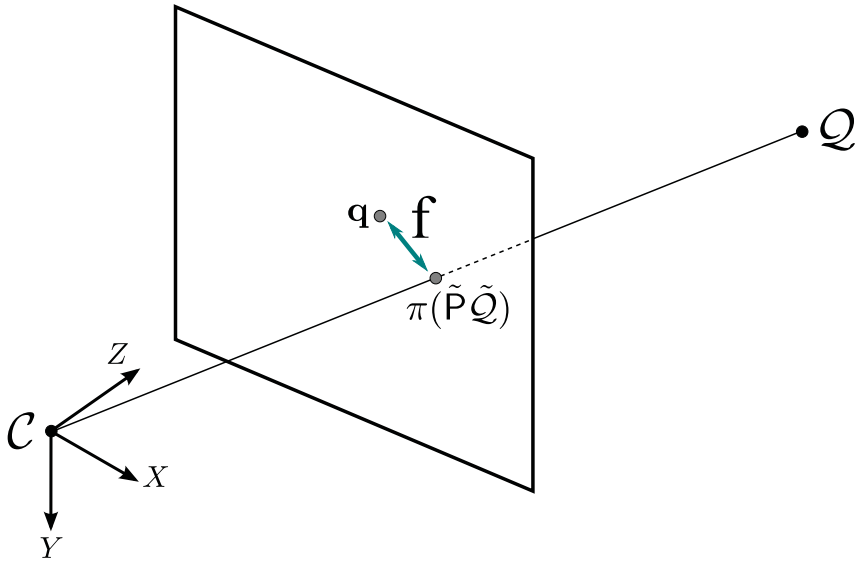


FIGURE 1.7 – **Erreur de reprojection.** L'erreur de reprojection est la distance entre l'observation $\mathbf{q}$ d'un point $\mathcal{Q}$ et sa projection dans l'image $\pi(\mathsf{P}\tilde{\mathcal{Q}})$.

les trois paramètres de la position de chaque point 3D) en cherchant à minimiser l'erreur de reprojection pour chacun des couples caméra-point 3D observé. On parle alors d'*ajustement de faisceaux*. La fonction à minimiser s'écrit donc :

$$f(\{\mathcal{R}_j, \mathbf{t}_j\}_{j=1}^{N}, \{\mathcal{Q}_i\}_{i=1}^{M}) = \sum_{1\leq i\leq M} \sum_{j\in\mathcal{D}_i} ||\mathbf{q}_{i,j} - \pi(\mathsf{P}_j\tilde{\mathcal{Q}}_i)||^2, \tag{1.40}$$

où les $\{\mathcal{R}_j, \mathbf{t}_j\}_{j=1}^{N}$ sont respectivement les orientations et les positions 3D prises par la caméra et $\{\mathsf{P}_j\}_{j=1}^{N}$ représentent ses matrices de projection. $\{\mathcal{Q}_i\}_{i=1}^{M}$ sont les points 3D à optimiser alors que $\mathbf{q}_{i,j}$ est l'observation 2D du $i^{ème}$ point 3D dans la $j^{ème}$ caméra. L'ensemble $\mathcal{D}_i$ quant à lui contient l'ensemble des indices des caméras observant le point $\mathcal{Q}_i$. Afin de minimiser cette fonction de coût, on utilisera l'algorithme du Levenberg Marquard décrit dans la section 1.2.3.

L'ajustement de faisceaux représente une étape fondamentale en reconstruction 3D. La majorité des contributions apportées par nos travaux est axée sur ce processus d'optimisation. C'est pour cette raison que nous détaillons son implémentation ci-après :

Construction des matrices Jacobienne et Hessienne Le principal objectif d'une itération d'ajustement de faisceaux consiste à estimer les incréments $\delta = (\delta_{camera}^T, \delta_{point}^T)^T$ [1] à appliquer aux paramètres à raffiner afin de minimiser la fonction de coût décrite dans l'équation 1.40. Ces incréments sont obtenus par le résolution du système linéaire défini par l'équation :

$$\mathsf{J}^T\mathsf{J} \times \delta = \mathsf{J}^T \times \mathbf{r}, \tag{1.41}$$

où $\mathbf{r}$ est le vecteur concaténant toutes les erreurs de reprojections des points 3D participant à l'ajustement de faisceaux, et J représente la matrice Jacobienne créée à partir des dérivées partielles de l'erreur de reprojection par rapport aux paramètres de la scène à optimiser comme

1. Les indices $_{camera}$ et $_{point}$ représentent respectivement la dépendances aux paramètres des poses de la caméra et des points 3D.

suit :

$$\mathsf{J} = \left(\frac{\partial f}{\partial \mathcal{R}_1}, \frac{\partial f}{\partial \mathbf{t}_1} \ldots, \frac{\partial f}{\partial \mathcal{R}_N}, \frac{\partial f}{\partial \mathbf{t}_N}, \frac{\partial f}{\partial \mathcal{Q}_1}, \ldots, \frac{\partial f}{\partial \mathcal{Q}_M} \right). \tag{1.42}$$

Étant donné que tous les points ne sont pas observés par toutes les caméras et que la matrice J contient plusieurs coefficients nuls, la matrice Hessienne, qui s'écrit selon l'approximation de de Gauss-Newton $\mathsf{J}^T\mathsf{J}$, possède une structure à la fois creuse et par blocs (voir figure 1.8). Cette matrice peut alors être représentée de la manière suivante :

$$\mathsf{J}^T\mathsf{J} = \begin{pmatrix} \mathsf{U} & \mathsf{W} \\ \mathsf{W}^T & \mathsf{V} \end{pmatrix}, \tag{1.43}$$

où :

- ▷ $\mathsf{U} = \mathsf{J}_{camera}^T \mathsf{J}_{camera}$, matrice carrée (de taille $6N \times 6N$) contenant des blocs diagonaux de taille 6×6 ;
- ▷ $\mathsf{V} = \mathsf{J}_{point}^T \mathsf{J}_{point}$, matrice carrée (de taille $3M \times 3M$) diagonale par blocs de taille 3×3. Elle est donc facilement inversible ;
- ▷ $\mathsf{W} = \mathsf{J}_{camera}^T \mathsf{J}_{point}$, matrice (de taille $6N \times 3M$)) exprimant les intercorrélations entre les paramètres des points 3D et les paramètres extrinsèques des caméras. La structure de W est directement liée au fait que les points sont vus ou non dans les images. W a un nombre, non nuls et égal au nombre de reprojections 2D, de blocs de dimension 6×3 .

Résolution éparse des équations normales par le complément de Schur L'approximation de Gauss-Newton de la matrice Hessienne aura la structure représentée dans la figure 1.8. Cette structure particulière rend possible la résolution de l'équation 1.41 de manière très rapide par le complément de Schur (voir l'annexe A pour plus de détails).

$$\begin{pmatrix} \mathsf{U} & \mathsf{W} \\ \mathsf{W}^T & \mathsf{V} \end{pmatrix} \begin{pmatrix} \delta_{camera} \\ \delta_{point} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_{camera} \\ \mathbf{g}_{point} \end{pmatrix}, \tag{1.44}$$

avec $\mathbf{g}_{camera} = \mathsf{J}_{camera}\mathbf{r}$ et $\mathbf{g}_{point} = \mathsf{J}_{point}\mathbf{r}$.
Plus de détails sur la résolution de ce système sont disponible dans l'annexe A.

L'ajustement de faisceaux utilisant le complément de Schur est donc une technique très rapide et efficace. En effet, seule la matrice V, diagonale par bloc 3×3, nécessite d'être inversée. De plus les équations normales résolues à chaque itération ont une structure éparse par blocs. La prise en compte explicite de cette structure permet de résoudre ces équations avec une complexité réduite. Cette solution exploite la structure éparse correspondant aux paramètres des points. La complexité devient ainsi linéaire pour les points, mais reste cubique pour les poses de la caméra, c'est-à-dire qu'elle est réduite de $O(N^3M^3)$ à $O(N^3M)$.

1.4 Algorithmes et données utilisés

Le but de cette section est de présenter les deux entrées principales de notre méthode. Nous présenterons tout d'abord un algorithme de localisation et cartographie simultanées par vision monoculaire avant d'introduire les données utilisées au cours de nos travaux notamment le GPS, les modèles 3D des bâtiments ainsi que le modèle d'élévation de terrain.

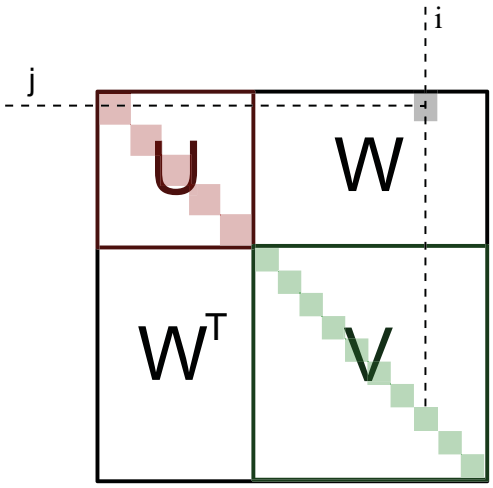


FIGURE 1.8 – **Structure de la matrice Hessienne de l'ajustement de faisceaux approchée $J^T J$.**

1.4.1 Algorithme de localisation et cartographie simultanées

La méthode que nous présentons dans ce mémoire s'appuie fortement sur la méthode de SLAM monoculaire proposée par Mouragnon et al. (2006) (schématisée sur la figure 1.9). Les algorithmes de SLAM monoculaire ont pour but de localiser une caméra dans un environnement inconnu. La résolution de ce problème s'effectue en passant par la construction en ligne d'une carte de l'environnement à partir des observations réalisées par la caméra au cours du temps. Ainsi, à l'instant $t + 1$, la caméra observe des amers déjà présents dans la carte de l'environnement. Ces observations vont permettre de localiser la caméra. La carte pourra alors être enrichie : la position des amers existantes pourra être raffinée et de nouveaux amers seront ajoutés, ce qui permet de cartographier des zones de l'environnement qui n'ont pas encore été explorées.

Dans la suite, nous allons détailler comment sont réalisées ces différentes étapes dans la méthode proposée par Mouragnon et al. (2006). Cela permettra en particulier d'introduire les notions nécessaires à la bonne compréhension de la suite du mémoire.

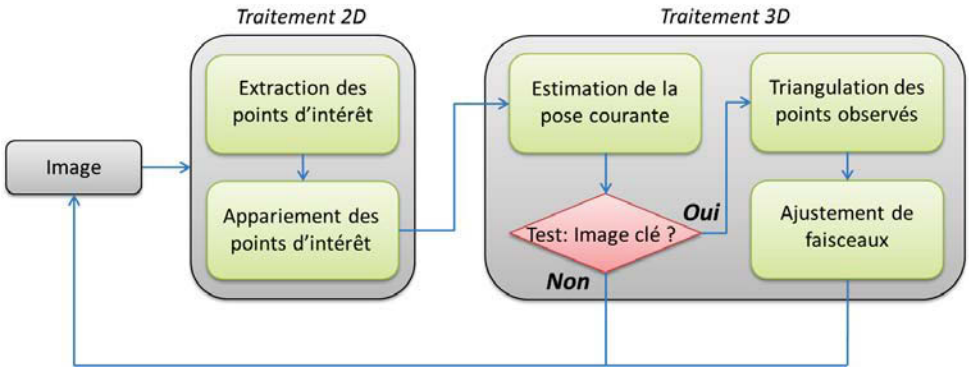


FIGURE 1.9 – **Schéma du fonctionnement du SLAM de Mouragnon et al. (2006).**

1.4.1.1 Traitements 2D

Le but des traitements 2D (c'est-à-dire au niveau des images) de la méthode de SLAM visuel est triple : détecter les points d'intérêt, associer les points d'intérêt correspondant entre les images successives et enfin fournir un critère indiquant si l'image courante est une image clé ou non (cette notion étant définie plus loin).

Points d'intérêt. Les *points d'intérêt* utilisés dans la méthode de Mouragnon sont des points de Harris (Harris and Stephens (1988)) : il a été montré par Schmid et al. (2000) que ces points d'intérêt offrent une bonne répétabilité, ce qui maximise les chances de pouvoir détecter les mêmes points dans les images successives. Il est à noter que la détection des points d'intérêt est faite par baquets, c'est-à-dire que l'image est découpée en sous-zones et que les points d'intérêt sont recherchés dans chacune de ces zones. Ceci permet de mieux répartir les points d'intérêt dans l'image, ce qui est une configuration nécessaire pour maximiser la qualité des résultats des différents processus de reconstruction 3D. Plus de détails sur les détecteurs de points d'intérêt et leur performance respective peuvent être trouvés dans l'article de Mikolajczyk and Schmid (2002).

Descripteurs associés. Le *descripteur* d'un point d'intérêt est la signature permettant de mesurer sa similarité avec tout autre point d'intérêt. L'idée du descripteur est de calculer la signature du voisinage du point d'intérêt considéré. Pour mettre en correspondance les points d'intérêt, on mesure alors la distance entre leurs descripteurs respectifs (cette mesure étant dépendante du type de descripteur utilisé). La taille du voisinage pris en compte pour le calcul du descripteur peut être choisie automatiquement à partir de l'échelle du point d'intérêt (par exemple pour SIFT, Lowe (2004)). Dans notre contexte, le voisinage est une fenêtre de taille constante (20×20).

Le descripteur utilisé dans la méthode d'origine est la corrélation ZNCC (Zero Normalized Cross Correlation). L'idée de la ZNCC est de comparer l'intensité lumineuse des voisinages des points d'intérêt à associer. Cependant, la méthode ZNCC est une méthode peu robuste aux changements d'apparence importants, et donc aux larges déplacements de caméra. Dans notre étude, ce descripteur a été remplacé par un descripteur offrant des performances similaires à SURF (Speeded Up Robust Features, Bay et al. (2006)) qui repose sur la distribution des ondelettes de Haar 2D sur le voisinage des points d'intérêt. D'autres descripteurs ont été présentés et comparés dans l'article de Mikolajczyk and Schmid (2005).

Notions d'image clé. Dans la méthode de SLAM utilisée, toutes les images de la vidéo n'ont pas le même rôle. Certaines images seront uniquement localisées dans l'environnement précédemment reconstruit. Les autres images, appelées *images clés*, ont un rôle particulier. Nous verrons dans la section suivante que ces images sont utilisées par la brique de reconstruction 3D. Il est donc nécessaire de définir un critère permettant de savoir si une image est clé ou non. Ce critère, essentiellement 2D, est notamment basé sur le nombre d'appariements 2D avec l'image clé précédente. Lorsque le nombre d'appariements 2D entre l'image courante et la dernière image clé est inférieur à un seuil, une image clé est détectée [2].

2. C'est l'image précédente qui est sélectionnée car c'est la dernière image ayant eu un nombre de correspondances supérieur au seuil

1.4.1.2 Traitements 3D

Les données créées par la brique 2D, c'est-à-dire les associations de points d'intérêt et la détection d'images clés, sont utilisées par les algorithmes de localisation et de reconstruction 3D. Dans cette section, nous allons brièvement présenter les différents cas de figure rencontrés.

Initialisation. Au début de la reconstruction, seules les informations calculées par le suivi 2D sont disponibles. En particulier, aucune information sur la structure de l'environnement n'est fournie. La première étape de la reconstruction 3D est donc d'initialiser la carte de l'environnement, c'est-à-dire la pose des premières caméras et la position des points 3D observés (figure 1.10(a)). Pour cela, dès que la brique 2D a détecté 3 images clés, la structure peut être retrouvée grâce aux associations 2D/2D fournies et à un calcul de la matrice essentielle (section 1.3.2.2). Une fois la reconstruction initialisée, le processus incrémental est lancé.

Processus incrémental. Dès lors que l'initialisation est réalisée, les appariements 2D fournis par le module de suivi permettent de remonter à des associations 2D/3D entre les points d'intérêt de l'image courante et les points 3D préalablement reconstruits. La pose de la caméra courante (figure 1.10(b)) peut alors être estimée à partir de ces appariements (section 1.3.2.4).

Si la caméra courante est détectée comme étant une caméra clé, elle est alors utilisée pour augmenter la reconstruction de l'environnement (figure 1.10(c)) :

- ▷ Sa pose et ses observations sont utilisées pour trianguler de nouveaux points 3D (section 1.3.2.3).
- ▷ Un ajustement de faisceaux est appliqué pour raffiner la géométrie de la reconstruction.

La particularité des travaux de Mouragnon et al. (2006) est que l'ajustement de faisceaux ne raffine pas l'ensemble de la reconstruction. En effet, afin d'assurer un traitement temps-réel, l'ajustement de faisceaux est uniquement réalisé sur une sous-partie de la reconstruction. Cette sous-partie est constituée des N dernières caméras clés et des points 3D associés. De plus, parmi cette structure, seuls les paramètres des n dernières caméras clés et des points 3D qu'elles observent sont raffinés. Les $N - n$ autres caméras clés sont fixées, ce qui permet d'apporter les contraintes assurant la cohérence géométrique de la reconstruction de proche en proche. On parle dans ce cas d'ajustement de faisceaux local ou glissant.

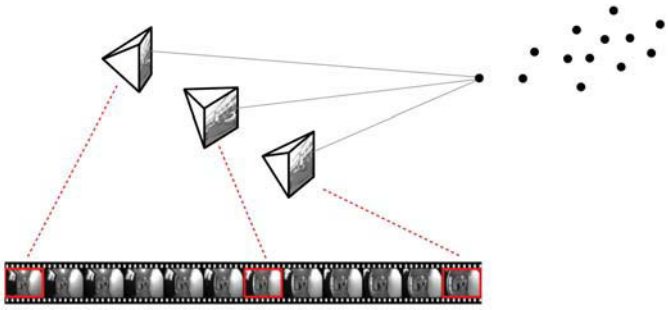
Mouragnon et al. (2006) ont montré que cette méthode permet de réaliser des reconstructions de grande échelle en temps-réel. En particulier, les expériences ont montré que les résultats obtenus sont similaires aux méthodes utilisant un ajustement de faisceaux global (par exemple Royer et al. (2005)). Des exemples de reconstructions obtenues avec cette méthode peuvent être trouvés à la figure 1.11.

Cependant, cet algorithme de SLAM monoculaire est sensible à l'accumulation des erreurs ainsi qu'à la dérive du facteur d'échelle. Nous allons maintenant présenter les données utilisées dans notre méthode afin de corriger ces dérives à savoir le GPS et des modèles SIG (les modèles 3D des bâtiments ainsi que le modèle d'élévation de terrain).

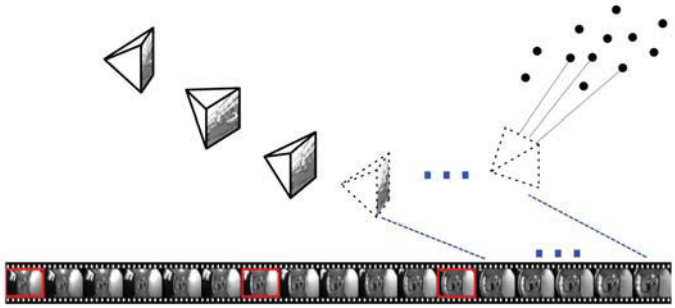
1.4.2 Les données d'entrée

1.4.2.1 Les données GPS

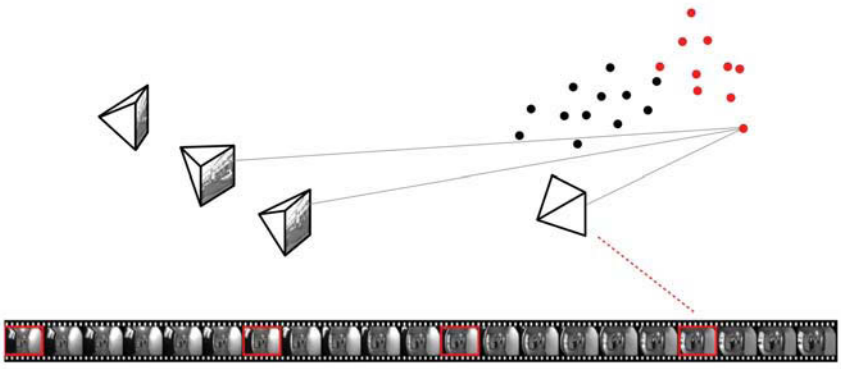
Au jour d'aujourd'hui, les seules solutions commercialisées de localisation de véhicule sont basées sur le GPS. Selon la précision de ce dernier, deux types d'applications sont visés : le



(a) **Initialisation.** L'initialisation de la structure à partir de 3 images clés est réalisée à l'aide de l'algorithme des 5 points.



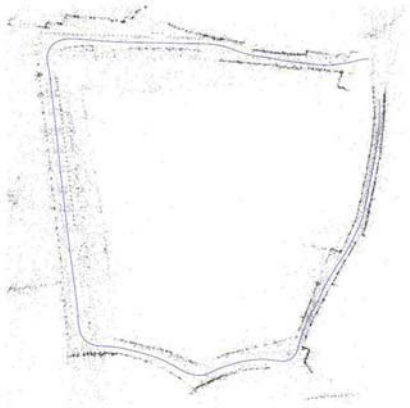
(b) **Localisation de la caméra.** Chaque nouvelle image est localisée à partir des points 3D déjà reconstruits.



(c) **Création d'une nouvelle caméra clé.** Si la caméra courante est clé, elle est alors utilisée pour trianguler de nouveaux points 3D. Les paramètres des 3 dernières caméras clés et des points 3D qu'elles observent sont alors raffinés à l'aide d'un ajustement de faisceaux.

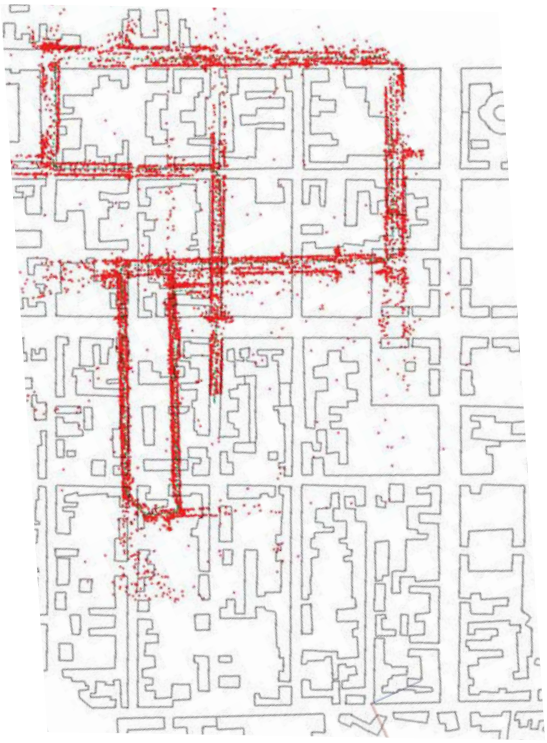
FIGURE 1.10 – **Résumé de la méthode de reconstruction 3D utilisée (Mouragnon et al. (2006)).** Les caméras en pointillés sont des caméras classiques et les caméras en trait plein sont des caméras clés. Les images encadrées en rouge sont les images détectées comme étant des images clés.

(a)



(b)

(c)



(d)

FIGURE 1.11 – **Reconstructions SLAM.** Exemples de reconstructions obtenues avec la méthode de Mouragnon et al. (2006) sur une distance de 400 mètres (a,b) et sur une distance de 1.5 kilomètres (c,d).

SPS (Standard Positioning Service) disponible à tout utilisateur et le PPS (Precision Positioning Service) réservé au gouvernement américain et aux utilisateurs militaires.

Le système GPS permet de fournir à la fois une position 3D ainsi que la correction de l'horloge de son récepteur. Ces informations sont déterminées en se basant sur une constellation de 24 satellites actifs répartis sur six orbites autour de la terre. La répartition des satellites est tel qu'à tout instant de la journée au moins quatre satellites soient visibles pour toute position sur la planète. Chacun de ces satellites transmet à 50 bit/s un signal contenant sa position et son horloge. Ainsi, la position 3D et la correction de l'horloge du récepteur GPS sont estimées à partir de la mesure du temps de vol du signal entre chacun des satellites et le récepteur GPS. Cette mesure est connue sous le nom de "pseudo-distance".

Sources d'erreurs D'une manière générale, l'imprécision de GPS peut être expliquée par les trois sources d'erreurs suivantes relatives au calcul de la pseudo-distance :

- ▷ **Erreurs dues aux satellites** Les erreurs d'horloge, les erreurs d'éphémérides du satellite.
- ▷ **Erreurs dues à la transmission** Les retards troposphériques et ionosphériques, les réflexions du signal GPS sur des surfaces environnantes aux récepteurs GPS, les occultations des satellites.
- ▷ **Erreurs dues au récepteur** Erreurs liées aux composants électroniques internes de celui-ci.

Étant donnée l'importante imprécision sur la mesure d'altitude fournie par un GPS standard, on ne considérera, dans la suite, que la position 2D $\mathbf{v} = (x^{gps}, y^{gps})^T$ du GPS dans le plan de déplacement.

Modèles des erreurs GPS D'après une étude établie dans Chausse et al. (2005), l'incertitude du GPS peut être modélisée par un biais $\mathbf{b} = ((\mathbf{b})_x, (\mathbf{b})_y^T$ auquel est additionné un bruit gaussien de faible amplitude $\mathbf{v}_g = ((\mathbf{v}_g)_x, (\mathbf{v}_g)_y)^T$. Ainsi, pour une mesure GPS donnée $\mathbf{v}^{gps}$, la position 2D réelle $\mathbf{v}^{réel}$ peut être déterminée comme suit :

$$\mathbf{v}^{réel} = \mathbf{v} + \mathbf{b} + \mathbf{v}_g. \tag{1.45}$$

Le biais $\mathbf{b}$ est supposé fixe entre deux constellations (voir figure 1.12). Les variations éven-

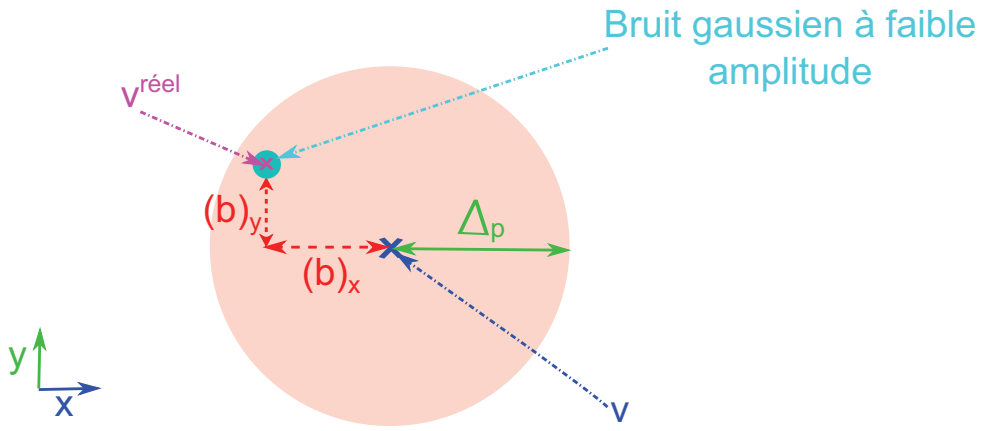


FIGURE 1.12 – **Modélisation du bruit du GPS.**

tuelles du biais sont bornées par le seuil Δp défini par la relation suivante :

$$\Delta p = \text{HDOP} \times \text{UERE}, \tag{1.46}$$

où :

- ▷ HDOP (Horizontal Dilution of Precision) : précision sur le positionnement des satellites par rapports au récepteur. Étant donné que cette information n'est pas disponible, elle peut être remplacée par GDOP (Geometric Dilution of Precision) dont la valeur est disponible à la sortie du GPS.
- ▷ UERE (User Equivalent Ranging Error) : précision de la mesure de la distance entre l'utilisateur et chaque satellite. Sa valeur est fournie par le fabriquant du GPS.

Caractéristiques du GPS utilisés. Dans le cadre de nos travaux, le GPS utilisé est de type "uBlox EVK-6PPP-0" caractérisé par :

- ▷ Une fréquence de localisation de 1 Hz.
- ▷ Une précision sur la position horizontale de 2.5m.
- ▷ Une précision sur la vitesse de 0.1m/s.
- ▷ Une précision sur le cap de 0.5 degré.
- ▷ Une altitude limite opérationnelle de 50km.
- ▷ Une vitesse limite opérationnelle de 500m/s.

1.4.2.2 Les modèles 3D urbains

Dans cette section, nous détaillerons les caractéristiques des modèles 3D disponibles à grande échelle puis nous présenterons les modèles utilisés dans nos travaux.

Système d'Information Géographique Un *Système d'Information Géographique* (*SIG*) est un système d'information permettant de représenter un ensemble de données géoréférencées. La plupart du temps, ces données sont représentées sous formes de différentes couches apportant chacune leurs informations : cadastre, modèle d'élévation de terrain, image satellite, bâtiments 3D, *etc.* Ces bases de données sont de plus en plus présentes dans notre quotidien à travers, par exemple, les systèmes d'assistance à la navigation, les visites virtuelles, les projets architecturaux, les cartes du monde interactives, *etc.* De plus, si elles étaient auparavant principalement destinées aux professionnels dans le cadre de leurs activités, elles sont désormais de plus en plus utilisées par le grand public. A ce titre, la présentation de ces bases a évolué et elles sont désormais disponibles à travers de nombreuses applications web (voir figure 1.13).

Caractéristiques des modèles 3D utilisés. Les modèles 3D disponibles dans les SIG ont différentes provenances. Ils sont majoritairement issus d'instituts nationaux (par exemple l'IGN [3] pour le Géoportail [4] en France), de collectivités locales ou d'entreprises spécialisées. Dernièrement, des communautés se sont formées autour de la création de modèles 3D. Par exemple, le groupe Google propose le logiciel Google SketchUp [5] qui permet de créer aisément des modèles 3D et de les incorporer dans Google Earth. Tout cela permet l'apparition rapide de données 3D pour des zones de plus en plus larges.

Il est néanmoins important de noter que les modèles 3D disponibles à grande échelle dans les SIG sont des modèles approximatifs (voir figure 1.14). Notons que la faible précision de ces modèles constituera un point crucial de nos travaux. En particulier les modèles 3D diffèrent souvent de la réalité sur ces points :

3. Institut Géographique National - www.ign.fr
4. www.geoportail.fr
5. Site de la communauté : sketchup.google.com/intl/fr/community

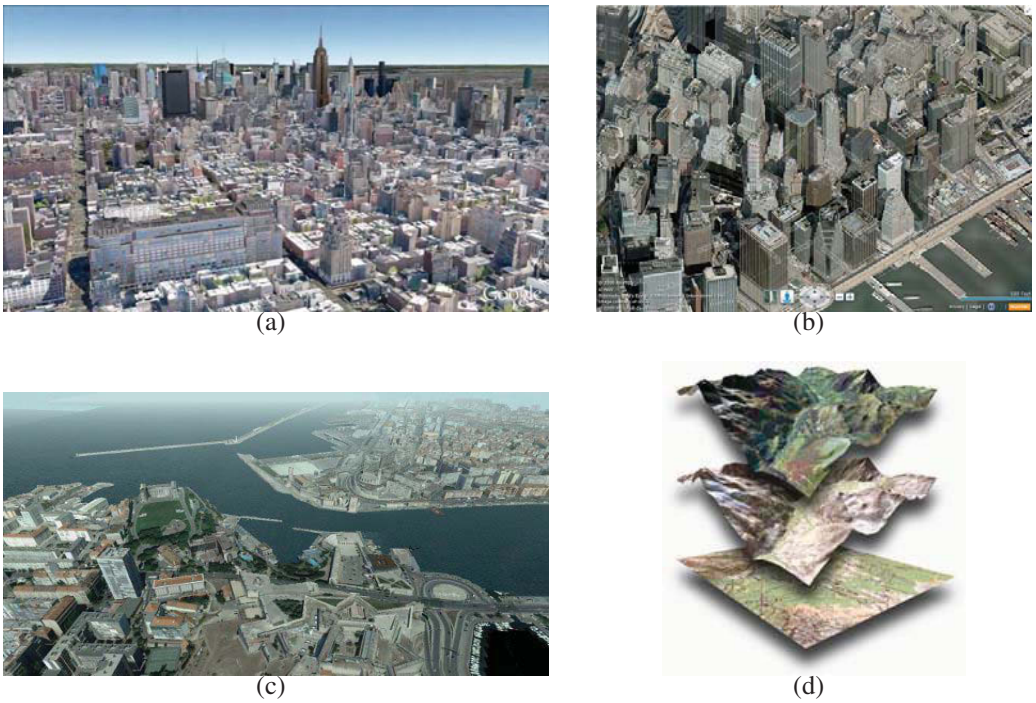


FIGURE 1.13 – **Exemples de SIG.** (a) Google Earth, (b) Microsoft Bing Map 3D, (c) Modèles 3D des bâtiments proposé par l'IGN et (d) différentes couches de modèle d'élévation de terrain fournies également par l'IGN.

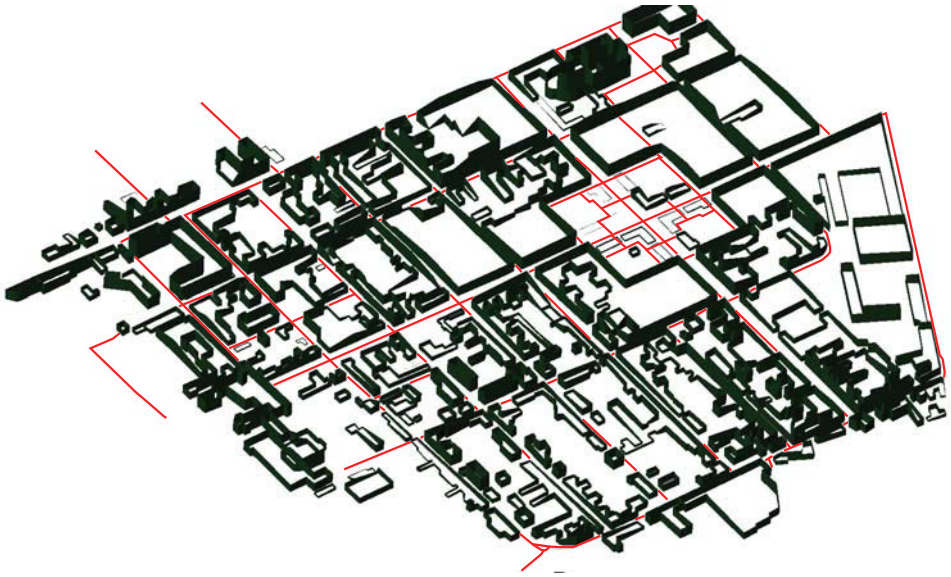


FIGURE 1.14 – **Modèles SIG utilisés.** En vert les modèles 3D des bâtiments, en rouge le modèle d'élévation de terrain.

▷ Pour les modèles 3D des bâtiments :

- **Simplification de la géométrie.** La géométrie des modèles 3D ne détaille que la structure globale des bâtiments. En particulier, les portes, les fenêtres et les colonnades sont absentes en 3D et n'apparaissent que sur la texture des modèles. Ceci implique donc que l'information géométrique fournie par ces modèles est limitée.

- **Géométrie imprécise.** En plus d'être simplifiée, la géométrie obtenue est imprécise. En effet, pour permettre de créer des modèles à grande échelle, des processus automatiques ont été déployés. Ces processus reposent en général sur des mesures d'images satellitaires (Elaksher et al. (2002)), ce qui limite la précision obtenue et peut engendrer des erreurs. Ainsi, il sera nécessaire de prendre en compte cette incertitude au cours de nos différents processus.

- **Modèles sans texture.** Actuellement, les modèles 3D des bâtiments disponibles gratuitement sur tout le territoire Français ne sont pas texturés, d'où l'absence de l'exploitation de l'information liée à la texture dans nos travaux.

▷ Pour le Modèle d'Elévation de Terrain (MET) :

- **Représentation simplifiée.** Comme les modèles 3D des bâtiments, ce modèle est caractérisé par une représentation minimaliste où chaque route est schématisée par un segment 3D représentant son axe (voir figure 1.14). Même si cette représentation nous informe sur l'inclinaison de la route latéralement (*i.e* montée ou descente), l'inclinaison de la route le long de la largeur reste inconnue. Par conséquent, dans la suite, on supposera que cette inclinaison est nulle. D'autre part, ces modèles ne représentent pas les imperfections de la route notamment les courbures ou les dos-d'âne. Pour cette raison, dans nos travaux, chaque segment de route sera modélisé par un plan fini et parfait. On détaillera ultérieurement l'impact de l'absence de ces informations sur la précision de nos résultats.
- **Imprécision du modèle.** D'une façon similaire, le MET est reconstruit à partir d'images satellitaires. Ceci explique l'imprécision du modèle qui est au maximum de 2 mètres.

Il est important de noter que les modèles décrits ci-dessus correspondent à l'état des modèles au début de nos travaux, en 2010. Au jour d'aujourd'hui, la géométrie tend à s'améliorer. On peut raisonnablement imaginer que la précision des modèles va continuer de s'affiner tandis que leur disponibilité va croître de façon importante.

CHAPITRE 2

Etat de l'art

Ce premier chapitre a pour objectif de faire le tour d'horizon des approches existantes pour la localisation d'une caméra mobile dans un milieu urbain. Nous commencerons par les méthodes de localisation sans aucune connaissance a priori. Ensuite, des méthodes de localisation par vision intégrant des informations a priori sur la pose de la camera issues de capteurs additionnels ou des informations a priori sur l'environnement traversé seront présentées.

2.1 Localisation basée vision sans a priori

L'objectif de cette section consiste à présenter les approches classiques basées vision permettant la localisation d'une caméra mobile (embarquée sur un véhicule) dans un environnement urbain. D'une manière générale, le principe de la localisation par vision monoculaire est identique pour la grande majorité de ces approches. La première position de la caméra est inconnue et est donc fixée arbitrairement. Ensuite, le déplacement 2D au niveau image nous informe sur le déplacement 3D de la caméra ainsi que la géométrie de la scène. Pour estimer le déplacement 2D, il est possible de se limiter au déplacement d'éléments spécifiques et faciles à identifier dans l'image, nommés *éléments d'intérêt*. Ces éléments sont généralement des zones ou plus couramment un ensemble spécifique de segments (*e.g.* contours) ou des points (*e.g.* coins) de l'image. A partir de l'observation du déplacement 2D, il est possible d'inférer le mouvement 3D de la caméra. Ceci peut être réalisé directement à partir de l'information 2D : on parlera alors d'odométrie visuelle. Une autre approche qui s'est fortement développée ces dernières années consiste à passer par l'intermédiaire de la création d'une carte 3D de l'environnement : nous parlerons alors de SLAM visuel. Dans ce qui suit, nous allons commencer par détailler les spécifications de chacune de ces familles d'approches [1]. Ensuite, nous énumérerons leurs limitations.

1. La distinction SLAM visuel/Odométrie visuelle varie souvent d'un auteur à l'autre

2.1.1 Odométrie visuelle

Comme nous l'avons précisé précédemment, l'idée de l'odométrie visuelle est d'inférer directement le déplacement 3D de la caméra à partir du déplacement 2D des éléments d'intérêt dans les images. En particulier, ceci implique qu'aucune reconstruction 3D de l'environnement n'est nécessaire pour localiser la caméra au fil du temps. Pour ceci, il est possible, par exemple, de s'appuyer sur la géométrie épipolaire liant les images successives (comme le font Tardif et al. (2008)) ou d'aligner photométriquement ces images (comme c'est le cas pour Comport et al. (2007)). Néanmoins, l'estimation du déplacement 3D à partir de la transformation 2D observée au niveau des images est un processus généralement coûteux et peu robuste. Pour faire face à ces limitations, il est souvent nécessaire de simplifier la transformation 2D recherchée entre les images en exploitant des hypothèses simples sur l'environnement observé. En particulier, il est possible d'exploiter le mouvement de certains plans de la scène (Simond and Rives (2004), Silveira et al. (2008)) ou plus couramment, dans le cadre du déplacement d'un véhicule terrestre, le mouvement particulier du sol qui est alors supposé plan (Scaramuzza and Siegwart (2008), Wang et al. (2005), Ke and Kanade (2003), Liang and Pears (2002)). Dans ce cas, le déplacement, entre plusieurs images, de toutes les observations d'un même plan décrit une homographie. Une fois cette homographie calculée, le déplacement de la caméra peut être déduit (Triggs (1998)). Par conséquent, la trajectoire du véhicule peut être reconstruite sur l'ensemble de la séquence vidéo.

2.1.2 SLAM visuel

La "localisation et cartographie simultanées" (SLAM), a été initialement développée pour promouvoir la création d'un robot autonome qui crée une carte 3D cohérente de son environnement tout en se localisant à l'intérieur de celle-ci. Cet algorithme a été largement étudié par la communauté de robotique et celle de la vision par ordinateur au cours des deux dernières décennies. En conséquence, de nombreux algorithmes SLAM ont surgi. En effet, initialement les approches de type SLAM utilisaient plusieurs capteurs, tels que le laser (Castellanos et al. (1999)), le radar (Dissanayake et al. (2001)), le sonar (Tardos et al. (2002)), ou encore l'odométrie fournie par le mouvement des roues. La fusion de ces données était généralement l'axe principal de ces recherches (Castellanos et al. (2001)). Comme, parallèlement à cela, les recherches en vision par ordinateur se sont largement développées, la vision a pris une part prépondérante dans ces schémas de fusion de données. Elle fut même utilisée comme l'unique capteur d'entrée pour le SLAM, nous parlons alors de SLAM visuel. Particulièrement, le SLAM monoculaire réfère à l'utilisation d'une seule caméra comme principal ou unique capteur pour réaliser la localisation. Dans cette branche du SLAM, la cartographie peut être de différentes natures :

- ▷ Métrique (*e.g.* Nister et al. (2004),Klein and Murray (2007)) : dans ce cas les éléments de la scène sont reconstruits en 3D dans un même repère et la notion de distance est respectée.
- ▷ Topologique (*e.g.* Cummins and Newman (2009)) : les éléments de la scène sont simplement reliés entre eux par des informations de voisinage.
- ▷ Une combinaison des deux (*e.g.* Schleicher et al. (2009), Angeli et al. (2009), Lim et al. (2012b)) : des informations de voisinage reliant différentes reconstructions métriques, chacune ayant son propre repère.

Alors que le SLAM métrique est plus adapté aux applications liées à une localisation locale et précise, le SLAM topologique s'adresse principalement au problème de planification globale (*e.g.* calcul du chemin le plus court/ sécurisé). Étant donné que nous visons à travers nos travaux

des applications de Réalité Augmentée où la précision de la localisation (en termes d'erreur de re-projection) est le critère le plus important à respecter, nous allons dans la suite nous intéresser uniquement aux approches de SLAM monoculaire métrique. Nous rappelons que l'idée sous-jacente à cette approche est que l'observation d'un même point d'intérêt par des points de vue différents permet d'estimer sa position 3D. Il est ainsi possible de reconstruire une carte de l'environnement au fil du temps sous la forme d'un nuage de points 3D. Une fois cette carte créée à l'instant t, il est alors possible de localiser la caméra à l'instant $t+1$ à partir de celle-ci. Dès lors que la caméra est localisée, la carte peut être mise à jour grâce aux observations obtenues à l'instant $t+1$: de nouveaux amers sont alors créés et les positions des amers déjà existants sont raffinées. Afin d'assurer ce processus, deux approches différentes peuvent être adoptées :

- ▷ Les méthodes du SLAM par filtrage dont le processus d'optimisation repose sur les techniques d'estimation par filtrage (*e.g.* filtre de Kalman, filtre à particule, *etc.*).
- ▷ Les méthodes du SLAM basé sur la notion d'image clé reposant sur une optimisation avec un ajustement de faisceaux.

2.1.2.1 SLAM par filtrage

Pour répondre au problème de la localisation et la cartographie simultanées, la communauté de robotique s'appuie sur des outils statistiques tels que le filtre à particules (Eade and Drummond (2006)) ou le filtre de Kalman (Davison et al. (2007); Lemaire et al. (2007)). Ce dernier, dans sa version étendue (Kalman (1960)), reste néanmoins actuellement le plus répandu. Dans les systèmes du SLAM par vision basés sur le filtre de Kalman, l'objectif est d'estimer, à chaque image, un vecteur d'état composé à la fois par la pose courante de la caméra et par les positions de tous les amers qui constituent la carte. A chaque nouvelle image, les amers existants dans la carte et identifiés dans l'image courante sont utilisés pour localiser la caméra. Par la suite, les positions des amers existants sont raffinées et les nouveaux amers sont ajoutés au vecteur d'état. Par ailleurs, en plus de l'estimation de la pose de la caméra, une matrice de covariance est estimée à tout instant et associée au vecteur d'état. Celle-ci permet de quantifier l'incertitude sur chacune des données de l'état courant.

Néanmoins, la limite la plus importante de l'approche SLAM basé sur les techniques de filtrage est le temps de traitement nécessaire à son fonctionnement. En effet, l'étape de mise à jour (c'est-à-dire l'étape permettant de calculer la pose courante de la caméra et de raffiner la carte) a une complexité en V^2, où V est la taille du vecteur d'état. Dès lors, dès que la taille de la carte reconstruite est importante (*i.e.* plus d'une centaine de points), un traitement en temps réel n'est plus envisageable avec la méthode en l'état. Pour faire face à ce problème, des nouvelles versions plus rapides, qui seront évoquées dans la suite de ce chapitre, ont été développées.

2.1.2.2 SLAM basé image clé

Cette approche de SLAM connue également sous le nom de SfM incrémental (Incremental Structure from Motion) a été initialement introduite par la communauté de vision par ordinateur. Les outils utilisés pour assurer la localisation et la reconstruction 3D sont principalement la triangulation, calcul de pose, ajustement de faisceaux *etc.* Comme c'est le cas pour le SLAM par filtrage, la reconstruction 3D ainsi que la localisation de la caméra se font au fur et à mesure que de nouvelles images arrivent. Toutefois contrairement au SLAM par filtrage où le vecteur d'état est ré-estimé à chaque image, le SLAM basé image clé cherche à extraire un sous-ensemble d'images offrant suffisamment de parallaxe pour réaliser cette estimation d'où

la notion d'*image clé*. Le raffinement est assuré grâce au processus d'ajustement de faisceaux dont la complexité algorithmique est cubique en nombre M d'images clés à optimiser. Pour obtenir des performances temporelles compatibles avec un traitement temps réel, il est important de réduire le nombre de variables à optimiser. Pour cela, l'optimisation s'applique généralement à un sous ensemble d'images clés contenant, entre autres, la dernière image clé créée, nous parlons alors d'ajustement faisceaux local. Cet ensemble peut être choisi sur une fenêtre temporelle glissante autour de l'image courante (Mouragnon et al. (2006)) ou encore selon un critère spatial comme dans Klein and Murray (2007) [2]. La résolution du problème de l'ajustement de faisceaux local, à chaque nouvelle image clé, est effectuée par le même procédé que pour l'ajustement de faisceaux global (décrit par l'équation 1.40). Optimiser uniquement un nombre restreint de paramètres permet ainsi de réaliser cette tâche en temps réel. Les nombreux progrès réalisés en vision font qu'à l'heure actuelle, ces méthodes permettent d'obtenir en temps réel une localisation de la caméra sur de longues distances (plusieurs kilomètres Civera et al. (2009), Konolige et al. (2007)).

2.1.2.3 Comparaison entre SLAM par filtrage et SLAM basé image clé

Le SLAM par filtrage et celui basé sur la notion d'image clé sont équivalents dans le principe. Les deux approches estiment le déplacement de la caméra tout en construisant l'environnement qu'elle observe. Leurs processus d'optimisation partagent également la même fonction de coût à savoir celle qui minimise les erreurs de re-projection liant l'ensemble des points reconstruits et leurs observations 2D dans les images. La principale différence entre ces approches réside dans la formalisation du problème du SLAM. En effet, initialement, cet algorithme peut être schématisé sous forme d'inférences sur un graphe dont les éléments sont représentés dans la figure 2.1(a). Dans ce graphe, l'ensemble $\left\{\mathcal{C}_j\right\}_{j=1}^{5}$ représente les différentes poses de la caméra tandis que les points 3D reconstruits sont représentés par $\{\mathcal{Q}_i\}_{i=1}^{5}$. Ces éléments sont reliés entre eux à travers les observations 2D $\{\mathbf{q}_{i,j}\}_{j=1}^{5}$, observations du point $\mathcal{Q}_i$ dans les caméras $\left\{\mathcal{C}_j\right\}_{j=1}^{5}$. Ces liens sont représentés par les segments du graphe. Au cours du processus SLAM, la taille de ce graphe va continuellement augmenter étant donné qu'à chaque image une nouvelle pose de caméra ainsi qu'un nouvel ensemble de points 3D et les observations 2D correspondantes sont ajoutés. Plus la taille du graphe est importante plus le temps d'exécution est élevé et plus l'espace mémoire nécessaire est important. Afin d'éviter ce problème, il semble indispensable de remplacer ce graphe par une représentation plus réduite.

Dans le cas du SLAM par filtrage, la réduction de taille de graphe est assurée en marginalisant toutes les poses de la caméra autres que la pose actuelle. Les points 3D observés par les caméras marginalisées sont, quant à eux, retenus. Il en résulte ainsi un graphe relativement compact comme le montre la figure 2.1(b). Sa taille reste alors constante et n'augmente que si des nouvelles zones sont explorées. Toutefois le principal problème de cette représentation réside dans le fait que le graphe devient rapidement complètement inter-connecté à cause de la marginalisation des précédentes poses de la caméra. En effet, la suppression d'une pose de caméra du graphe implique l'ajout d'un grand nombre de liens entre les points 3D observés par cette caméra et les observations 2D correspondantes. Ceci impacte directement la complexité de calcul ce qui représente une réelle limitation de cette approche.

Une deuxième option pour réduire la taille du graphe du SLAM est apportée par le SLAM

2. Ils proposent en plus de traiter sur deux *threads* différents la reconstruction et l'optimisation. Cela permet de réaliser, en parallèle de la localisation, un raffinement global de la carte 3D.

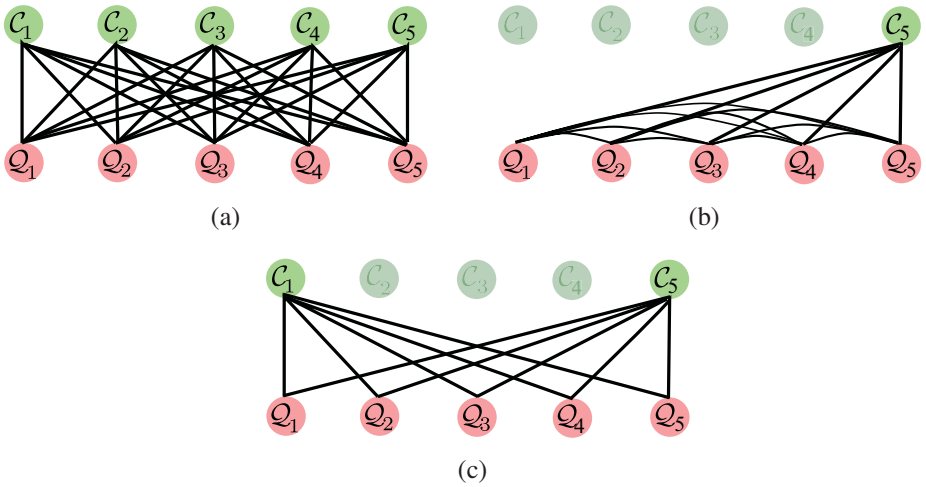


FIGURE 2.1 – **Différents graphes schématisant le problème du SLAM**. L'ensemble $\left\{\mathcal{C}_j\right\}_{j=1}^{5}$ représente les poses de la caméra. Les points 3D reconstruits sont représentés par $\{\mathcal{Q}_i\}_{i=1}^{5}$. Les observations 2D des points $\{\mathcal{Q}_i\}_{i=1}^{5}$ dans la caméra $\mathcal{C}_j$ sont schématisées par les liens du graphe. (a) Graphe complet du SLAM. (b) Graphe associé au SLAM par filtrage. (c) Graphe associé au SLAM basé image clé.

basé image clé et son processus d'optimisation basé sur l'ajustement de faisceaux. En exploitant la notion d'image clé sur une fenêtre glissante, le graphe associé à cette approche reste éparse comme le montre la figure 2.1(c). En effet, contrairement au SLAM par filtrage, l'ensemble de poses antérieures de la caméra ne sont pas marginalisées. Ceci implique un graphe ayant plus d'éléments par rapport à l'approche par filtrage. Toutefois l'absence de la marginalisation permet de réduire considérablement les inter-connexions. Ainsi le graphe résultant demeure relativement efficace, même si le nombre de ses éléments est plus élevé.

Certaines études, notamment celle de Strasdat et al. (2012), comparant ces deux approches en termes de la qualité de la localisation obtenue, ont conclu qu'elles ont toutes les deux leurs avantages et inconvénients. Tandis que le SLAM basé image clé peut être plus précis, le SLAM par filtrage est plus adapté pour la fusion de données. Il est donc en pratique très difficile de dire qu'une méthode est meilleure que l'autre car cela est très dépendant de l'application visée. Toutefois, la même étude présentée par Strasdat et al. (2012) a mis en avant que l'approche de type image clé avec ajustement de faisceaux est meilleure en termes de compromis entre précision et coût de calcul. Notons cependant que des travaux récents cherchent à réduire les dimensions du filtre Kalman pour traiter le problème de complexité SLAM par filtrage Gamage and Drummond (2013), Civera et al. (2009), Leonard and Newman (2003). Par ailleurs, d'autres récents travaux sur le SLAM basé image clé permettent de combler certaines lacunes qui existaient par rapport au SLAM par filtrage. A savoir, la propagation des covariances à travers l'ajustement de faisceaux (Eudes and Lhuillier (2009)), la fusion de capteur dans l'ajustement de faisceaux (Eudes et al. (2010a), Michot et al. (2010), Lhuillier (2012)) et l'exploitation de primitives déjà reconstruites pour gérer la tâche de fermeture de boucles (Klein and Murray (2007)).

Remarque importante : Comme nous venons de le constater, quelle que soit l'approche

adoptée pour réaliser l'algorithme SLAM, l'objectif reste toujours le même : localiser en temps réel une caméra dans un milieu inconnu en passant par l'intermédiaire de la création en ligne d'une carte de l'environnement. De son côté, même si la majorité des méthodes d'odométrie visuelle ne crée pas la carte 3D de l'environnement, les primitives 3D sont implicitement paramétrées à travers les poses de la caméra et les observations 2D. Il est donc tout à fait possible de construire ces primitives au fur et à mesure que la trajectoire est reconstruite (*e.g.* Scaramuzza et al. (2009b)). En ce sens, les deux méthodes du SLAM visuel et d'odométrie visuelle permettent bien de réaliser simultanément la localisation d'une caméra et la cartographie de l'environnement parcouru. Voilà pourquoi nous ne distinguerons généralement plus ces notions dans la suite de ce mémoire. Nous parlerons le plus souvent de SLAM puisque selon nous, c'est ce terme qui traduit le mieux l'idée générale sur laquelle s'appuient ces méthodes.

2.1.3 Limites de ce type de méthodes

Si les approches de localisation monoculaire de type SLAM permettent d'estimer la trajectoire d'une caméra dans un environnement inconnu, elles présentent néanmoins des limites qui empêchent leur utilisation pour un grand nombre d'applications. Ces limites sont principalement le non géo-référencement et les dérives notables au niveau de la reconstruction obtenue.

2.1.3.1 Localisation non géo-référencée

Dans l'approche SLAM, aucune information a priori n'est disponible sur l'environnement parcouru et sur la pose initiale de la caméra. Le SLAM fournit le déplacement relatif de la caméra au fil du temps par rapport au repère initial de la caméra, généralement choisi arbitrairement. Donc la caméra n'est pas localisée par rapport à un repère absolu. On parle alors d'une localisation non géo-référencée. L'absence de géo-localisation n'est pas nécessairement problématique dans certaines applications (reconstruction de trajectoire, suivi de convois, *etc.*). Néanmoins, cela devient une limite bloquante pour proposer une alternative au système GPS classique par exemple.

2.1.3.2 Dérives sur les longues trajectoires

La reconstruction obtenue par le processus SLAM monoculaire est sujette à différentes sources de dérives. La cause principale de ces dérives est l'erreur issue des mesures effectuées dans les images. En effet, l'ensemble du système SLAM s'appuie sur des points d'intérêt détectés dans les images successives. Cependant, l'étape de détection de ces primitives n'est pas parfaite. Les coordonnées de ces dernières présentent un bruit, généralement supposé gaussien. Naturellement, les erreurs effectuées sur les mesures 2D ont un impact direct sur la reconstruction 3D du nuage de points et sur la localisation de la caméra. A ce type d'erreurs s'ajoutent également les erreurs de calibrage, les erreurs d'arrondis des calculs et les erreurs de mise en correspondance des primitives. Il en résulte deux types de dérives :

- ▷ **Accumulation des erreurs.** Le SLAM est un processus incrémental, c'est-à-dire qu'à chaque image, l'algorithme s'appuie sur la reconstruction de la scène obtenue à l'image précédente pour estimer les nouveaux points 3D et nouvelles poses de caméra. Ceci implique que l'erreur commise sur un paramètre à un instant $t-1$ se propage aux paramètres estimés à l'instant t. Si l'étape de raffinement est censée réduire l'influence d'une telle erreur en ré-estimant continuellement l'ensemble des paramètres, celle-ci maintient une interdépendance des paramètres et donc une propagation des erreurs. La

prise en compte de la fermeture de boucle peut aider à limiter ce phénomène. Toutefois cette approche est peu utile dans le cas d'un scénario d'aide à la navigation d'un véhicule puisque l'objectif de ce type d'application est de réaliser la trajectoire la plus rapide et donc généralement dépourvue de boucles.

▷ **Dérive du facteur d'échelle.** Dans le cas du SLAM monoculaire, la reconstruction est réalisée à un facteur d'échelle près. Ce facteur est fixé arbitrairement au début du processus et doit être propagé tout au long de la séquence. Dans le cas de la localisation monoculaire, il est difficile de propager ce facteur car il n'est pas observable au cours de la séquence. Il subit directement l'accumulation des erreurs et dérive au cours du temps, en particulier lorsque le mouvement de la camera est tel que le suivi d'un grand nombre de primitives est perdu entre deux images (mouvement brusque de la caméra, virage, *etc.*).

2.2 Localisation basée vision avec ajout d'informations additionnelles

Afin de pallier les limites du SLAM monoculaire énumérées dans la section précédente, des nouvelles approches ont été proposées intégrant des informations supplémentaires aux approches classiques présentées en section 2.1. Selon la nature de ces informations, nous distinguons deux grandes familles d'approches. La première s'intéresse à l'exploitation de capteurs supplémentaires renseignant directement sur la trajectoire de la caméra tels que le GPS, la centrale inertielle, l'odomètre *etc.*. La deuxième, quant à elle, se base sur la connaissance a priori de l'environnement parcouru (données SIG, base d'amers géo-référencés). Dans ce qui suit, nous allons détailler ces deux familles d'approches.

2.2.1 Intégration des capteurs supplémentaires

Comme il l'a été mentionné ci-dessus, les principales limitations des algorithmes de localisation monoculaire classiques sont le non géo-référencement et la dérive de la localisation. Pour faire face à ces problèmes, la solution la plus intuitive consiste à fusionner le SLAM visuel avec des capteurs supplémentaires (GPS, centrale inertielle, odomètre, *etc.*). Ces derniers fournissent soit des informations absolues sur la pose de la caméra (*e.g.* GPS) ce qui permet de résoudre le problème de géo-référencement soit des informations sur le mouvement relatif de la caméra (*e.g.* la norme de déplacement via un odomètre, l'orientation via l'angle de braquage ou la centrale inertielle, *etc.*) ce qui limite la dérive du facteur d'échelle et l'accumulation des erreurs. Quel que soit le type de capteur utilisé, nous distinguons deux approches différentes pour réussir la fusion des données :

Fusion par filtrage. L'approche la plus commune pour réaliser la fusion des données fournies par des capteurs demeure les techniques de filtrage. Dans le contexte de la localisation en milieu urbain, l'approche la plus répandue consiste à fusionner les données visuelles extraites du flux vidéo avec les données GPS. Par exemple, afin d'assurer une localisation à grande échelle, Schleicher et al. (2009) proposent d'introduire les mesures GPS dans les deux niveaux d'un SLAM hiérarchique (*i.e.* topologique/métrique) : un SLAM métrique bas niveau et un SLAM topologique haut niveau. Dans cette approche, les données GPS sont introduites dans le SLAM métrique à travers un filtre Kalman étendu (EKF) afin d'obtenir des sous reconstructions SLAM

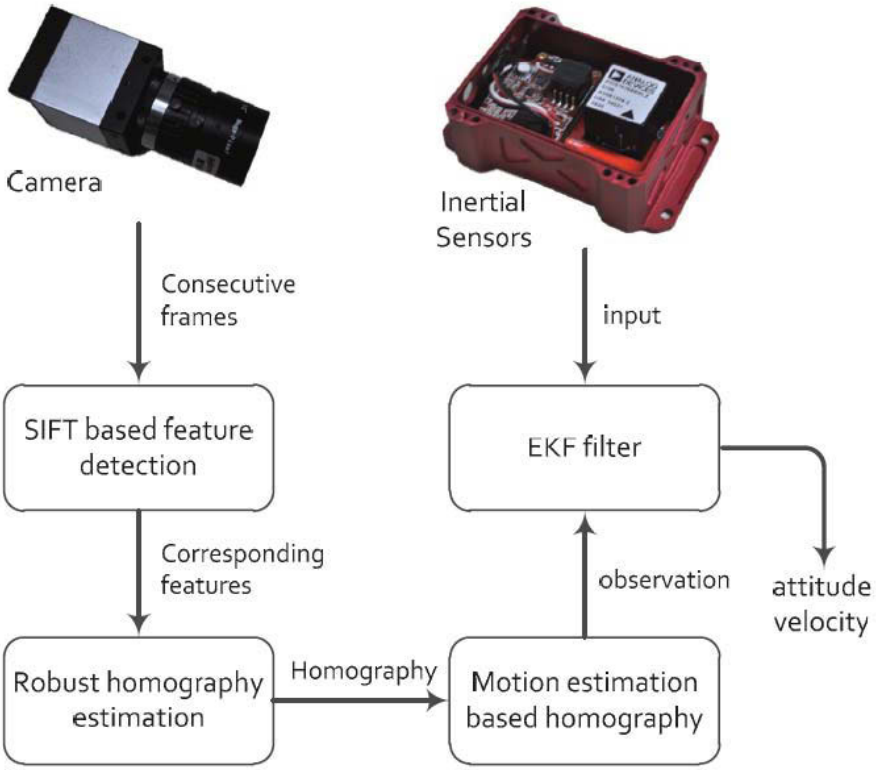


FIGURE 2.2 – **Schéma de la fusion proposée par Wang et al. (2012) (image extraite de Wang et al. (2012)).**

géo-localisées. Par la suite, les mesures GPS sont utilisées dans le SLAM haut niveau uniquement sous forme de contraintes absolues pour assurer la cohérence globale de la carte 3D en fusionnant des sous reconstructions SLAM obtenues par le SLAM bas niveau. Toutefois, en milieu urbain les données GPS peuvent être imprécises ou totalement inexistantes (*e.g.* tunnels, perte de signal...). Pour faire face au problème d'imprécision des données, Dawood et al. (2011) exploitent en plus des données GPS, les données fournies par un odomètre et un gyroscope. Plus précisément, ils proposent de fusionner les informations visuelles avec le GPS si les mesures fournies par ce dernier sont précises sinon ils fusionnent les informations visuelles avec les données du gyroscope et d'odomètre. Pour gérer les deux modèles d'évolution utilisés, Dawood et al. (2011) utilisent le IMM-UKF *Interacting Multiple Model-Unscented Kalman Filter* qui permet de gérer l'utilisation de plusieurs modèles. Par ailleurs pour assurer la localisation en l'absence totale des données GPS dans certaines zones, d'autres approches (*e.g.* Wang et al. (2012) et Fraundorfer et al. (2012)) proposent d'exploiter les mesures de la centrale inertielle afin de limiter les dérives du SLAM en cas de perte de signal GPS. Pour ceci, Wang et al. (2012) propose, par exemple, de fusionner l'odométrie visuelle avec les mesures de la centrale inertielle à travers un filtre Kalman étendu afin de déduire la vélocité du déplacement de la caméra (voir figure 2.2). Cette vélocité est par la suite utilisée lors d'un SLAM basé également sur un filtre de Kalman étendu afin d'estimer la position actuelle de la caméra.

Fusion à travers un ajustement de faisceaux. Récemment, de plus en plus de travaux se sont concentrés sur l'intégration des données issues de capteurs dans le processus de l'ajustement de faisceaux. Notons par exemple les travaux de Eudes et al. (2010a) qui proposent

d'exploiter les données fournies par un odomètre afin de mieux contraindre le facteur d'échelle. Dans le cadre de ces travaux, ces informations supplémentaires sont utilisées uniquement durant l'étape d'initialisation du processus d'optimisation. Même si une bonne initialisation implique une meilleure convergence, rien ne peut garantir le respect de la contrainte pendant et à l'issue de l'ajustement de faisceaux. Pour résoudre ce problème, d'autres méthodes proposent d'intégrer directement la contrainte supplémentaire dans la fonction de coût à minimiser. Nous parlerons alors d'ajustement de faisceaux contraint. Dans ce cas, l'approche la plus classique consiste à ajouter à l'erreur de re-projection standard un terme d'accroche aux données pondéré par un poids contrôlant l'influence de la contrainte (*e.g.* Eudes et al. (2010b) Michot et al. (2010), Hideyuki et al. (2010)). Pour apporter plus de robustesse face aux éventuelles données aberrantes du capteur, Lhuillier (2012) propose une nouvelle formalisation de la fonction de coût en intégrant une contrainte d'inégalité (cette approche sera détaillée dans la section 3.3) afin de ne prendre en compte la fusion que si les données du capteur ne dégradent pas significativement la géométrie multi-vues.

2.2.2 Exploitation de la connaissance a priori de l'environnement

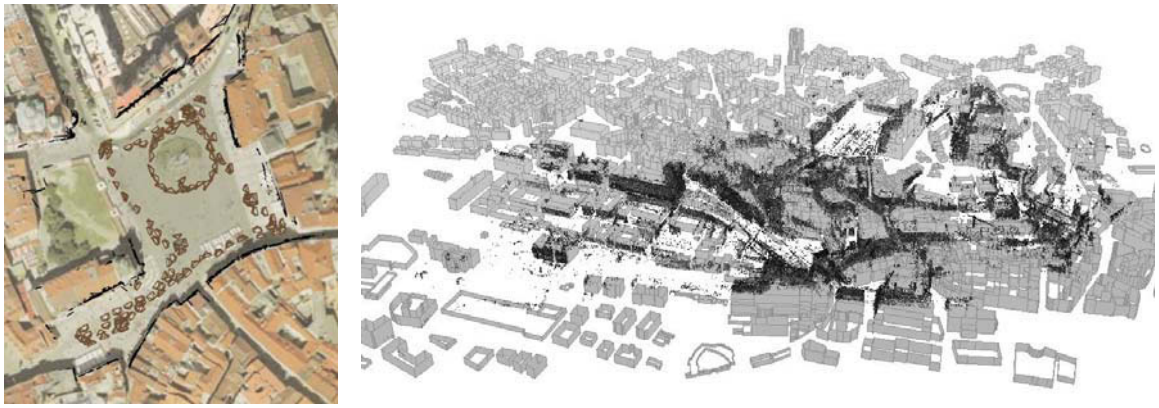
La connaissance a priori de l'environnement peut être apportée soit par des modèles géométriques fournis par exemple par des SIG (Système d'Information Géographique) tels que les modèles 3D des bâtiments, le modèle d'élévation de terrain *etc.* soit en exploitant des modèles photo-géométriques tels que des modèles texturés ou des bases d'amers géo-référencés préalablement reconstruites. Certaines approches exploitent directement cette connaissance pour estimer la pose de la camera à chaque instant (*i.e. model-based tracking*) alors que d'autres solutions les exploitent afin de contraindre une localisation de type SLAM pour en réduire la dérive. Selon le type d'information utilisée, nous distinguons ainsi deux familles approches.

2.2.2.1 Utilisation des modèles géométriques

Les Systèmes d'Information Géographique fournissent plusieurs couches de données purement géométriques qui peuvent être exploitées dans la localisation de véhicule. Les données géométriques utilisées sont souvent des modèles 2D (empreinte au sol) ou des modèles 3D des bâtiments de la zone explorée. L'augmentation récente du nombre de travaux exploitant ces modèles peut s'expliquer par différentes raisons. Tout d'abord, ces modèles sont exempts de dérive et présentent peu de données aberrantes. De plus, ils sont largement disponibles, en particulier depuis l'apparition des interfaces web grand public. A cela s'ajoute le fait que certains organismes et certaines communautés les distribuent désormais librement (*e.g.* openStreetMap, l'IGN pour la recherche). Par ailleurs, une information purement géométrique est durable dans le temps et est indépendante des conditions d'illumination comme du point de vue. Les travaux existants aujourd'hui peuvent être différenciés par le fait que la localisation de la caméra soit réalisée hors ligne ou en ligne.

Localisation hors ligne. Comme indiqué dans la section 2.1, les méthodes de localisation monoculaire permettent de construire un nuage de points 3D décrivant la géométrie de la scène ainsi que les différentes poses décrivant la trajectoire de la caméra. L'exploitation a posteriori des données SIG peut permettre de géo-localiser avec précision cette reconstruction. Une première approche consiste à estimer la transformation rigide qui recale au mieux la reconstruction en question avec les empreintes de bâtiments obtenues soit à partir d'images satellites dans

Kaminsky et al. (2009) ou directement à partir d'une carte 2D des bâtiments Strecha et al. (2010) (voir figure 2.3). Dans le cas où la dérive de la reconstruction est importante (*e.g.* trajectoire d'un véhicule sur une distance importante), la transformation nécessaire pour aligner la reconstruction au SIG n'est plus uniquement une similitude. Un modèle de transformation plus complexe doit être utilisé. Pour ceci, Saurer et al. (2010) ont proposé d'exploiter un ensemble de points de contrôle, ceux-ci étant fixés par l'utilisateur sur le plan de la zone parcourue. Dans le même esprit, Wang et al. (2013) et Lothe et al. (2009) proposent de corriger et géo-référencer la reconstruction initiale en la contraignant avec les modèles 3D des bâtiments à travers des transformations non rigides plus adaptées à la complexité des dérives du SLAM. Par exemple, Lothe et al. (2009) proposent un nouvel ajustement de faisceaux intégrant les contraintes géométriques apportées par les modèles 3D des bâtiments. Il en résulte une géo-localisation hors ligne précise. Plus de détails sur l'approche introduite par Lothe et al. (2009) seront donnés dans chapitre suivant.



(a) (b)

FIGURE 2.3 – **Recalage d'une reconstruction SfM sur un SIG.** Le recalage peut s'effectuer par exemple sur (a) une image satellite (*extrait de Kaminsky et al. (2009)*) ou (b) un modèle 3D des bâtiments (*extrait de Strecha et al. (2010)*).

Localisation en ligne. Concernant les méthodes permettant une localisation en ligne, peu de travaux existaient au début de notre étude (en 2010). L'activité dans ce domaine s'est développée au cours de ces trois dernières années. Nous distinguons alors, dans un premier temps, les méthodes qui exploitent les données SIG pour estimer la position absolue d'une unique image. Notamment, Bioret et al. (2009), Cham et al. (2010) utilisent l'empreinte au sol des bâtiments. L'idée sous-jacente à cette approche est d'extraire la forme des bâtiments observés à partir de l'image. Une fois cette étape réalisée, la structure extraite est alors recherchée au sein de la carte des empreintes au sol. Dans le même esprit, Taneja et al. (2012) propose de corriger la position de la caméra en utilisant les contraintes géométriques fournies par les modèles 3D des bâtiments. La géo-localisation de la caméra fournie par le GPS permet d'identifier l'ensemble de bâtiments observés. La pose de la caméra est par la suite raffinée afin de recaler au mieux la re-projection des modèles 3D des bâtiments sur les façades dans l'image. Toutefois, cette solution exige des modèles 3D avec une précision élevée (*i.e.* une erreur maximale de 0.5m). De tels modèles ne sont pas largement disponibles ce qui limite le déploiement de cette approche. De plus, étant donné que chaque image est traitée séparément, cette dernière manque de robustesse quand les contraintes apportées par les modèles des bâtiments sont insuffisantes.

Une autre famille de méthodes tente d'inclure cette information dans un processus de SLAM en ligne. Par exemple, Sourimant et al. (2007) ont proposé de remplacer la géométrie multi-vue classique par la rétroprojection de leurs observations sur le modèle 3D. De la même manière que pour les processus de SLAM, les points 3D obtenus sont utilisés pour localiser la caméra suivante et la carte est alors enrichie. Cependant, une fois calculée, la position des points 3D n'est jamais remise en cause. Ainsi, la précision de cette position est sensiblement liée à la précision des modèles 3D de l'environnement. La pose des caméras suivantes étant calculée à partir de ces points, la localisation à chaque instant de la caméra mobile varie donc avec la précision du SIG utilisé. Une approche différente a été proposée par Lothe et al. (2010). Dans le cadre de ces travaux, le déplacement de la caméra est estimé par un SLAM basé image clé dont le processus d'optimisation est assuré par un ajustement de faisceaux local. Au niveau des virages, un algorithme de type ICP (Iterative Closest Point, Rusinkiewicz and Levoy (2001)) est utilisé pour aligner localement les points 3D reconstruits avec les modèles 3D des bâtiments. Étant donné que la convergence de l'ICP n'est garantie que si la scène fournit suffisamment de contraintes géométriques, cette correction n'est pas appliquée dans les longues lignes droites. Ceci implique que la localisation demeure peu précise dans ces zones. Par ailleurs, dans leurs travaux de suivi d'objets, Tamaazousti et al. (2011) ont proposé une solution temps réel intégrant les contraintes géométriques fournies par un modèle géométrique de la scène directement dans l'ajustement de faisceaux. Même si cette solution est appliquée avec succès d'une façon en ligne pour assurer le suivi d'objets, son utilisation dans le cadre d'une localisation à grande échelle dans un milieu urbain est limitée à une exploitation hors ligne à cause du manque de robustesse face aux incertitudes des modèles SIG utilisés.

2.2.2.2 Utilisation des modèles photo-géométriques

Pour apporter plus de précision à la localisation basée vision, certaines approches dans l'état de l'art se basent sur l'exploitation des informations photo-géométriques. Ces informations sont fournies soit via des modèles texturés soit à partir d'une base d'amers géo-référencée.

Modèle texturé Dans des environnements restreints (*i.e.* à l'échelle d'une place) ou spécifiques (*e.g.* bâtiments architecturaux, *etc.*), il est parfois possible de se procurer un modèle 3D de l'environnement avec une texture de haute qualité. Dès lors, il est envisageable d'exploiter l'information photométrique associée à ce modèle 3D (*e.g.* Simon (2011) et Dawood et al. (2011)). L'image courante de la caméra peut être recalée à chaque instant à la texture de ce modèle. La caméra est alors localisée par rapport au modèle 3D. Si celui est géolocalisé, la caméra l'est donc également. Par exemple, Dawood et al. (2011) proposent de raffiner la pose de la caméra en utilisant la texture du modèle 3D de l'environnement. Pour cela, l'idée qu'ils proposent consiste à synthétiser des images virtuelles à proximité de la pose estimée (figure 2.4). La pose correspondante à l'image virtuelle visuellement la plus proche de l'image réelle courante est alors retenue comme une nouvelle observation de la pose courante.

Les approches présentées ci-avant permettent d'obtenir une localisation précise de la caméra mobile. Néanmoins, ces méthodes restent peu adaptées aux grands environnements du fait de la difficulté et du coût liés à l'obtention de ce type de modèles texturés. De plus, l'association entre les observations courantes de la caméra et les points du modèle est uniquement basée sur de l'information photométrique (généralement par une mise en correspondance de descripteurs). Par conséquent, cette approche est peu robuste aux conditions d'illumination (position du soleil, heure de la journée, *etc.*).

FIGURE 2.4 – **Localisation basée vision exploitant des modèles 3D texturés.** Il est possible de mettre en correspondance l'image courante (à gauche) avec une image de synthèse générée à partir du SIG (à droite) *(extrait de Cappelle et al. (2010)).*

Utilisation d'une base d'amers géo-référencée Afin d'étendre les approches de localisation par vision basées sur l'utilisation des modèles photo-géométriques à une exploitation à grande échelle, il est important de remplacer les modèles texturés difficiles à obtenir et à exploiter par des bases d'amers géo-référencées qui sont plus légères et faciles à déployer. Ces bases représentent généralement une reconstruction SLAM géo-référencée obtenue en intégrant au processus SLAM des informations issues soit d'un capteur supplémentaire soit en exploitant des modèles purement géométriques, comme nous l'avons expliqué précédemment. Ces bases peuvent être exploitées via un processus de re-localisation en ligne. Il est donc évident que plus la base est précise, plus la localisation l'est aussi. Pour assurer, une telle localisation, la majorité des solutions existantes dans l'état de l'art se base sur des algorithmes de reconnaissance de points de vue. Dans leurs travaux, Dong et al. (2009) introduisent une méthode de re-localisation temps réel qui se base sur une correspondance 2D-3D. La première étape de cette solution se déroule d'une façon hors ligne. Elle consiste à sélectionner un ensemble de points 3D optimal nommé ensemble points clés qui sera stocké dans un arbre de vocabulaire afin de simplifier et d'accélérer la localisation en ligne. Cette dernière est assurée grâce à une mise en correspondance 2D-3D entre les points d'intérêt obtenus par le détecteur SIFT et les points 3D existant dans l'arbre préalablement reconstruit. à partir de cette mise en correspondance, la pose de la caméra peut être estimée. Toutefois, cette méthode reste sensible à l'exhaustivité de points de vue qui ont permis de construire la base initiale. Pour faire face à ce problème, Irschara et al. (2009) proposent d'enrichir la base d'amers par l'intermédiaire d'un ensemble d'images de synthèse représentant des points de vue qui ne sont pas initialement représentés. D'autres méthodes telle que celle proposée par Tong et al. (2012), utilise une caméra panoramique lors de la localisation en ligne. En effet, un champ de vision plus large permet d'augmenter les chances de réussir le processus de reconnaissance de point de vue. Malgré l'amélioration des résultats de la localisation, ces méthodes restent sensibles aux changements d'illumination. Afin d'assurer, une géo-localisation permanente, une autre famille d'approches propose de fusionner le processus de reconnaissance de point de vue avec un algorithme de suivi de caméra Lim et al. (2012a), Gay-Bellile et al. (2010). Dans la solution proposée par Gay-Bellile et al. (2010), le suivi de la caméra est assuré par un SLAM basé image clé. A chaque image clé, l'algorithme de reconnaissance de point de vue essaie de mettre en correspondance les observations 2D de l'image courante avec les amers 3D de la base. Quand la mise en correspondance réussit, les dérives du SLAM sont corrigées en calculant la similitude entre les deux dernières poses obtenues par le SLAM et celles retournées par l'algorithme de reconnaissance de point de vue. Il en résulte une

localisation précise permettant des applications de Réalité Augmentée.

2.3 Bilan

Dans la première partie de ce chapitre nous avons présenté les principales approches de localisation d'une caméra mobile sans aucun a priori. Au jour d'aujourd'hui, la majorité des méthodes existantes cherche à localiser la caméra et cartographier l'environnement parcouru. Nous avons également montré que parmi ces méthodes, le SLAM basé image clé semble le plus adapté à notre objectif à savoir une localisation temps réel pour assurer des applications de Réalité Augmentée. Toutefois, la localisation fournie par le SLAM n'est pas géo-localisée et est souvent sujet de dérives dues à l'accumulation des erreurs et la mauvaise estimation du facteur d'échelle. Pour éviter ces limitations, certaines approches de localisation par vision intègrent des informations absolues issues d'un capteur additionnel ou de la connaissance a priori de l'environnement observé. Plusieurs techniques ont été adoptées pour prendre en compte ces informations. Des récents travaux tendent à les introduire directement dans le processus d'optimisation (*i.e.* l'ajustement de faisceaux dans le cas de SLAM basé image clé). Au vu de l'application visée, à savoir des applications de Réalité Augmentée, nous nous intéresserons plus particulièrement dans le chapitre suivant aux méthodes existantes qui intègrent des contraintes supplémentaires dans l'ajustement de faisceaux. Nous parlerons donc de l'ajustement de faisceaux contraint.

CHAPITRE 3

État de l'art : Ajustement de faisceaux contraint

Dans ce second chapitre dédié aux méthodes de l'état de l'art, nous nous intéresserons particulièrement aux algorithmes de localisation par vision d'un véhicule basés sur un SLAM avec un ajustement de faisceaux contraint. En effet, pour garantir une bonne localisation, l'intégration des contraintes supplémentaires se fait directement dans le processus d'optimisation à savoir l'ajustement de faisceaux, d'où l'appellation d'ajustement de faisceaux contraint. Dans le cadre ce chapitre, nous présenterons deux familles d'approches : l'ajustement de faisceaux contraint aux données SIG et l'ajustement de faisceaux contraint aux données GPS.

3.1 Introduction

Le SLAM est un processus incrémental qui alterne "l'estimation de la géométrie de l'environnement à partir du mouvement de la caméra" et "l'estimation du mouvement de la caméra à partir de la géométrie de l'environnement". Ceci implique que contraindre une des deux étapes explicitement revient à contraindre l'autre implicitement. C'est pourquoi deux approches de SLAM contraint ont été proposées dans la littérature : soit *contraindre la trajectoire* soit *contraindre la reconstruction*. Ces approches ont également donné naissance à différentes techniques pour intégrer une contrainte dans un ajustement de faisceaux : soit à travers *une contrainte dure* soit via *une contrainte douce*.

Quel que soit l'approche adoptée introduire une contrainte supplémentaire au niveau du processus d'optimisation du SLAM exige les trois étapes suivantes :

1. Établissement de la contrainte : identifier la propriété liée à la contrainte dont le respect ou le non respect peut être observé sur la reconstruction SLAM (nuage de points ou poses de caméra).
2. Formalisation de la contrainte : formaliser la propriété identifiée dans l'étape précédente sous forme d'une énergie à partir des éléments de la reconstruction SLAM. En général, ceci revient à minimiser une fonction de coût permettant de mesurer le respect de l'ensemble des contraintes (*e.g.* multi-vues et contrainte supplémentaire) tout en assurant la robustesse aux données aberrantes (*e.g.* M-estimateurs, contrainte inégalité *etc.*).

3. Optimisation de la scène vis à vis de la contrainte : le processus complet d'optimisation adopté pour garantir une bonne convergence de l'ajustement de faisceaux contraint.

Dans ce chapitre, nous allons uniquement détailler les méthodes de SLAM contraint les plus pertinentes dans le cadre de cette thèse et qui seront utilisées par la suite dans le mémoire. Nous commencerons par présenter une approche de SLAM contraint en reconstruction se basant sur l'intégration d'*une contrainte dure* dans l'ajustement de faisceaux (section 3.2). Puis, nous présenterons une approche de SLAM contraint en trajectoire se basant sur l'introduction d'*une contrainte douce* dans l'ajustement de faisceaux (section 3.3).

3.2 SLAM contraint à un modèle géométrique de la ville : Contraindre la reconstruction

Une manière de contraindre la trajectoire estimée par le SLAM consiste à contraindre la reconstruction des primitives 3D de cet algorithme, en d'autres termes le nuage de points reconstruit. En effet, les poses de la caméra sont déterminées à partir des primitives 3D. Ainsi contraindre ces derniers permettent de contraindre implicitement la trajectoire SLAM. Dans le cas d'une application de localisation de véhicule, il est préférable que la contrainte introduite au SLAM vérifie un certain nombre de propriétés :

- ▷ Une contrainte absolue sur la position des primitives, c'est à dire qu'il est préférable que la position d'une primitive 3D soit contrainte dans le repère global.
- ▷ Les primitives contraintes doivent être nombreuses et fréquemment observées.
- ▷ Une contrainte disponible à grande échelle et avec un coût d'accès aux données peu élevé.
- ▷ Une contrainte avec peu de données aberrantes.

Afin de satisfaire ces différents critères, certaines approches (*e.g.* Lothe et al. (2009) et Tamaazousti et al. (2011)) ont eu recours aux modèles 3D des bâtiments. En effet, dans les milieux urbains, les bâtiments représentent les structures géométriques les plus observées. Ayant des façades orientées dans différentes directions orthogonales aux plans des routes, ils fournissent autant d'informations géométriques permettant de contraindre principalement les degrés de liberté *dans le plan* de la reconstruction SLAM. S'agissant du coût d'accès à ce type de données, nous notons l'existence d'offre commerciale (*e.g.* IGN) mais aussi l'apparition de SIG communautaires (*e.g.* OpenStreetMap) offrant un accès gratuit. Par ailleurs, ces modèles ont également l'avantage d'être géo-référencés et contiennent peu de données aberrantes.

Dans la suite nous détaillerons le principe de la contrainte en reconstruction à travers deux exemples d'approches. La première approche est introduite par Lothe et al. (2009) dans le but de raffiner d'une façon hors ligne une reconstruction SLAM. La deuxième a été proposée par Tamaazousti et al. (2011). Bien que cette dernière a été introduite principalement pour le suivi en ligne des objets, elle peut être appliquée à notre cadre d'études a posteriori afin d'assurer une localisation hors ligne précise.

3.2.1 Établissement de la contrainte aux bâtiments : Segmentation du nuage de points et association point/plan

Le principe de la contrainte aux modèles 3D des bâtiments repose sur l'hypothèse qu'une reconstruction SLAM parfaite (*i.e.* sans dérive de l'échelle, sans accumulation d'erreurs, *etc.*)

devrait aboutir à un nuage de points 3D s'alignant avec les façades des modèles SIG utilisés. Ainsi, il est possible d'évaluer le respect ou le non respect de la contrainte en mesurant l'écart entre les positions des points 3D reconstruits et les façades 3D des bâtiments sur lesquelles ils sont supposées s'aligner. Cependant, tous les points obtenus par le SLAM ne correspondent pas forcément à une façade. En effet, certains points 3D peuvent modéliser des éléments appartenant au reste de l'environnement notamment les voitures garées, des panneaux routiers, des arbres *etc.* Cet ensemble de points $\mathcal{E}$ n'est pas alors concerné par la contrainte apportée par les modèles 3D des bâtiments. Ceci implique qu'une première étape d'établissement de contrainte est exigée. Au cours de cette étape, il est nécessaire d'identifier l'ensemble de points 3D $\mathcal{M}$ appartenant aux façades des bâtiments (*i.e.* les points concernés par la contrainte en reconstruction) et associer chacun de ces points à la façade qu'il modélise.

Pour les deux solutions proposées par Lothe et al. (2009) et Tamaazousti et al. (2011), l'étape de segmentation du nuage de points est assurée grâce à un simple lancé de rayon reliant le centre optique de la caméra à chaque point 3D. Ainsi, ce dernier est classifié comme appartenant aux modèles 3D des bâtiments si le rayon en question intersecte un plan des modèles 3D. Toutefois, cette solution ne gère pas le problème d'occultation qui est assez courant dans le contexte urbain. Par exemple tous les points modélisant un arbre occultant une façade d'un bâtiment sont considérés comme appartenant aux modèles (voir figure 3.1). Ceci peut nuire à la convergence du processus d'optimisation.

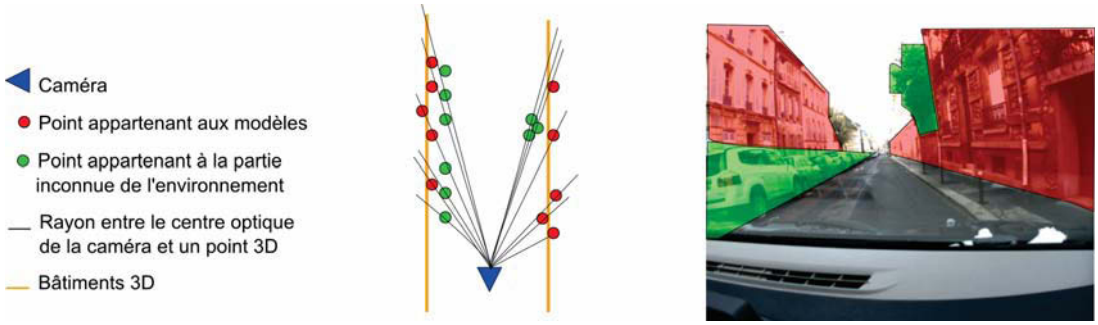


FIGURE 3.1 – **Illustrations des limitations de la méthode de segmentation.** Tout les points 3D sur la figure, même ceux modélisant les voitures garées et les arbres, sont classés comme appartenant aux modèles.

Concernant l'étape d'association point/plan des modèles, celle-ci est assurée en faisant correspondre chaque point $\mathcal{Q}_i \in \mathcal{M}$ associé aux modèles des bâtiments au plan le plus proche en terme de distance euclidienne (notons ainsi que le plan identifié au niveau de l'étape de segmentation peut ne pas correspondre au plan considéré au niveau de l'étape d'association point/plan). A cause de l'accumulation des erreurs de localisation, cette association peut être erronée comme le montre la figure 3.2. Ceci peut également entrainer des problèmes de convergence.

A la fin de l'étape d'établissement de contrainte, le nuage de points reconstruit est segmenté en deux ensembles : un ensemble de points 3D $\mathcal{E}$ associé à la partie inconnue de l'environnement et un ensemble de points 3D $\mathcal{M}$ associé aux modèles 3D des bâtiments. Chaque point 3D $\mathcal{Q}_i$ de ce deuxième ensemble est associé à une façade.

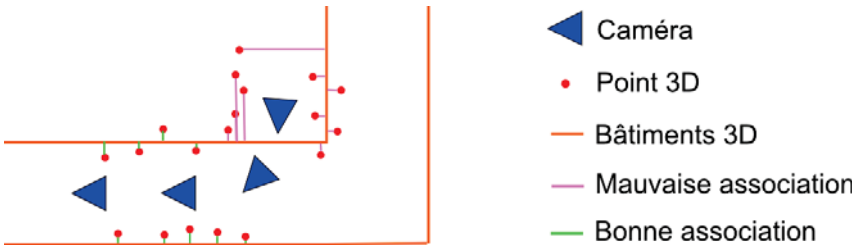


FIGURE 3.2 – **Illustrations des limitations de la méthode d'association point/plan.** Faire correspondre un point 3D au plan le plus proche entraine des fausses association (liens roses).

3.2.2 Formalisation de la contrainte apportée par les modèles 3D des bâtiments

Une fois l'étape d'établissement de contrainte effectuée, la contrainte en reconstruction peut être introduite dans l'ajustement de faisceaux. Pour ceci, une nouvelle fonction de coût permettant de mesurer le respect des différentes contraintes est utilisée. Dans la suite nous détaillerons les deux formalisations introduites par Lothe et al. (2009) et Tamaazousti et al. (2011).

Convergence des faisceaux en un point 3D appartenant à une façade de bâtiment : Formalisation proposée par Lothe et al. (2009). La méthode proposée par Lothe et al. (2009) est une des premières approches à intégrer la contrainte géométrique apportée par des modèles de bâtiments dans l'ajustement de faisceaux. Le principe général de cette approche consiste à encourager chaque point $\mathcal{Q}_i$ associé aux modèles à appartenir à sa façade correspondante $\mathbf{\Pi}^{h_i}$. Pour ceci, Lothe et al. (2009) proposent d'encourager, pour chacun des points $\mathcal{Q}_i \in \mathcal{M}$, la convergence des faisceaux issus de ses observations 2D en un point appartenant à sa façade comme le montre la figure 3.3. Ainsi, l'erreur de re-projection standard utilisée dans l'ajustement de faisceaux est remplacée par l'erreur de re-projection du barycentre $\mathcal{Q}'_i$ des points d'intersections des faisceaux avec la façade en question. Si $\mathbf{q}_{i,j}$ est l'observation 2D du point $\mathcal{Q}_i$ dans la $j^{ème}$ image clé alors l'erreur de re-projection considérée est donnée par :

$$\mathbf{g}'_{i,j} = \mathbf{q}_{i,j} - \pi(\mathsf{K}\mathcal{R}_j^T \left[\mathsf{I}_{3\times 3}\right| - \mathbf{t}_j] \, \mathcal{Q}'_i. \tag{3.1}$$

Nous rappelons que $\mathcal{R}_j$ et $\mathbf{t}_j$ sont respectivement l'orientation et la position de la $j^{ème}$ pose de la caméra et K est la matrice de ses paramètres intrinsèques. Minimiser cette erreur de re-projection est équivalent à faire converger $\mathcal{Q}_i$ vers le point $\mathcal{Q}'_i$. Si $\mathcal{D}_i$ est l'ensemble des indices des caméras clés qui observent le point $\mathcal{Q}_i$, alors la fonction de coût intégrant la contrainte des bâtiments s'écrit sous cette forme :

$$g'\left(\{\mathcal{R}_j, \mathbf{t}_j\}_{j=1}^N\right) = \sum_{i\in\mathcal{M}} \sum_{j\in\mathcal{D}_i} \rho\left(\|\mathbf{g}'_{i,j}\|, c\right). \tag{3.2}$$

La fonction $\rho(r, c)$ est un M-estimateur qui sera détaillée ultérieurement (voir section 3.2.3).

Nouvelle paramétrisation des points 3D associés aux modèles des bâtiments : Formalisation proposée par Tamaazousti et al. (2011) Plutôt que d'utiliser une fonction de coût favorisant le rapprochement d'un point 3D associé aux modèles vers son plan 3D associé, Tamaazousti et al. (2011) proposent de forcer ce point à appartenir à sa façade au cours de l'ajustement de faisceaux. Pour cela, les points associés aux modèles sont projetés préalablement

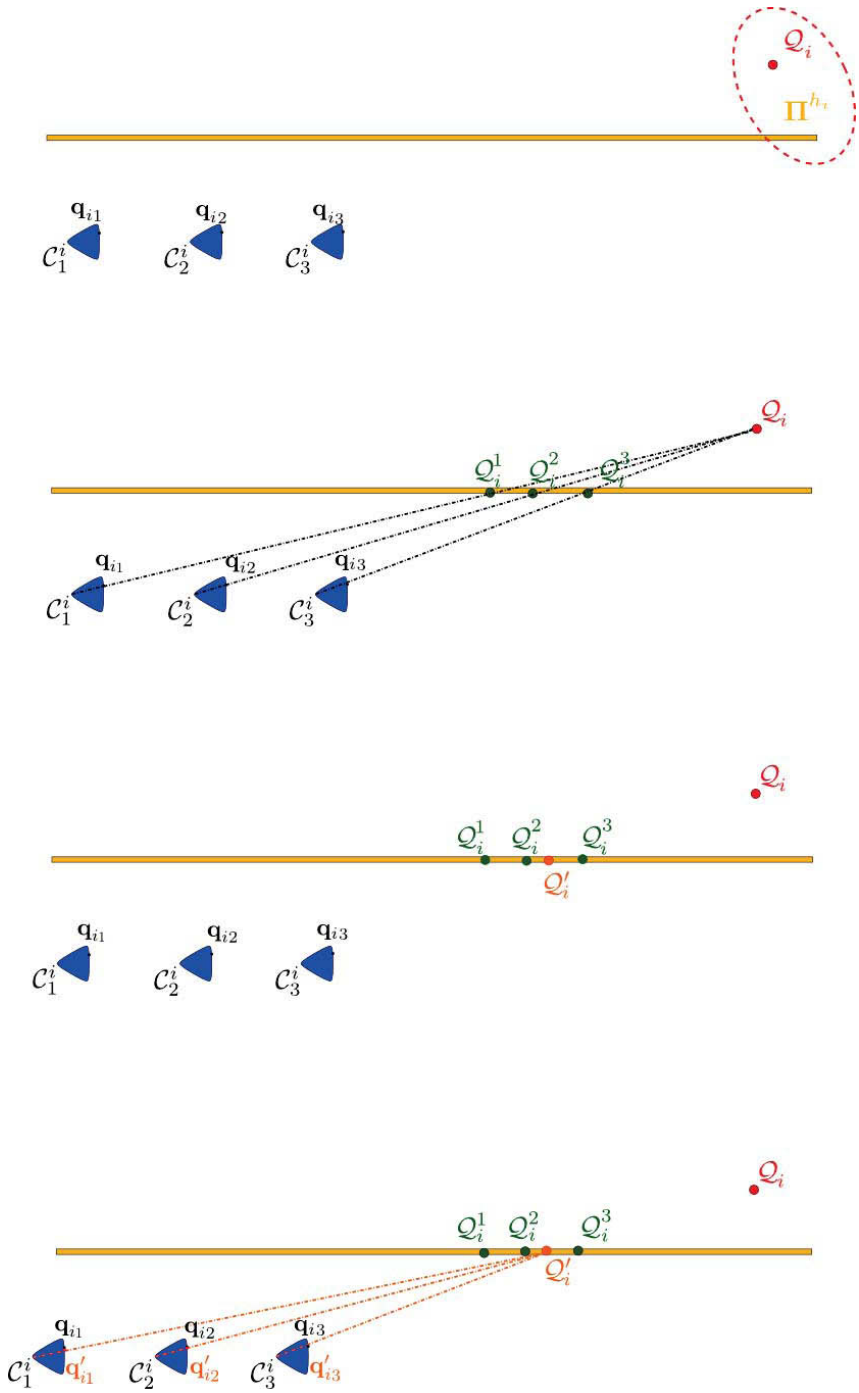


FIGURE 3.3 – **Fonction de coût proposée par Lothe et al. (2009).** Exemple d'un point 3D $\mathcal{Q}_i$ observé par 3 caméras. Les différentes sous-figures représentent les étapes successives. L'erreur de re-projection considérée est l'erreur entre les observations 2D $\left\{\mathbf{q}_{i,j}\right\}_{j\in\mathcal{D}_i}$, les re-projections $\left\{\mathbf{q}'_{i,j}\right\}_{j\in\mathcal{D}_i}$ du point 3D $\mathcal{Q}'_i$ dans les trois caméras. (Cette figure est extraite de Lothe et al. (2009))

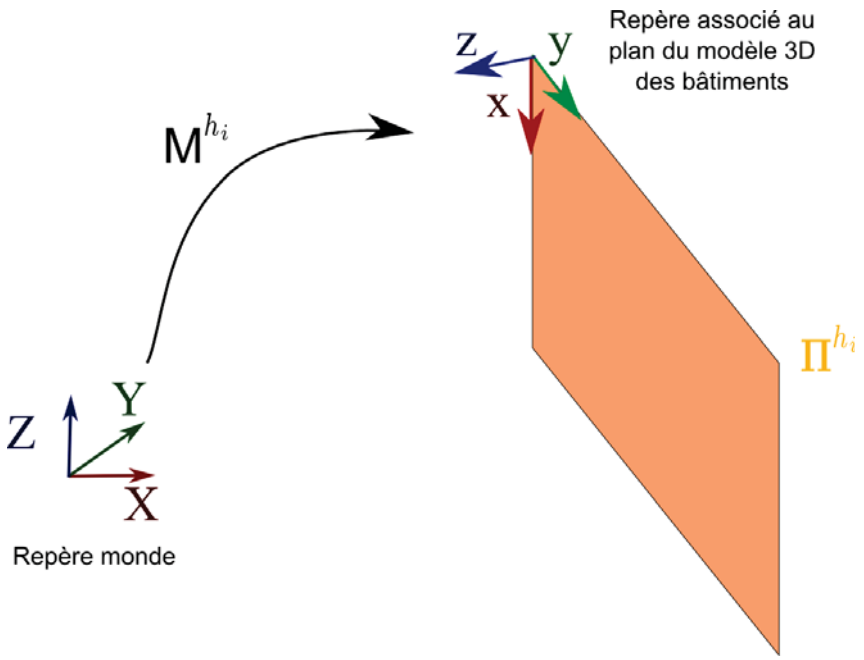


FIGURE 3.4 – **Matrice de changement de repère** M^{h_i}**.**

sur leurs façades correspondantes. Pour maintenir cette contrainte au cours de l'optimisation, Tamaazousti et al. (2011) proposent de passer par une nouvelle paramétrisation de ces points. Cette paramétrisation consiste à exprimer les coordonnées de chacun des points $\mathcal{Q}_i \in \mathcal{M}$ dans le repère du plan correspondant $\mathbf{\Pi}^{h_i}$ comme le montre la figure 3.4. Ainsi, en introduisant $\tilde{\mathsf{M}}^{h_i}$ la matrice de passage de taille (4×4) permettant de passer du repère monde au repère du plan $\mathbf{\Pi}^{h_i}$, la nouvelle paramétrisation du point $\mathcal{Q}_i$ est donnée par :

$$\tilde{\mathcal{Q}}_i^{h_i} = \begin{pmatrix} X_i^{h_i} \\ Y_i^{h_i} \\ Z_i^{h_i} \\ 1 \end{pmatrix} = \tilde{\mathsf{M}}^{h_i} \tilde{\mathcal{Q}}_i. \tag{3.3}$$

Appartenant au plan $\mathbf{\Pi}^{h_i}$, la coordonnée $Z_i^{h_i}$ du point $\tilde{\mathcal{Q}}_i^{h_i}$ le long de la normale du plan en question est donc supposée nulle. Tamaazousti et al. (2011) intègrent cette propriété dans l'ajustement de faisceaux en réduisant le nombre de paramètres à optimiser pour chaque point 3D associé aux modèles. Ainsi pour chaque point $\tilde{\mathcal{Q}}_i^{h_i}$ seuls deux paramètres au lieu de trois sont raffinés. Nous parlons alors d'*une contrainte dure*. La fonction de coût résultante est donc donnée par :

$$g\left(\{\mathcal{R}_j, \mathbf{t}_j\}_{j=1}^{N}, \left\{\mathcal{Q}_i^{h_i}\right\}_{i \in \mathcal{M}}\right) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{D}_i} \rho\left(\|\mathbf{g}_{i,j}\|, c\right), \tag{3.4}$$

avec

$$\mathbf{g}_{i,j} = \mathbf{q}_{i,j} - \pi(\mathsf{K}\mathcal{R}_j^T \left[\mathsf{I}_{3\times 3}\right| - \mathbf{t}_j] \left(\tilde{\mathsf{M}}^{h_i}\right)^{-1} \tilde{\mathcal{Q}}_i^{h_i}) \tag{3.5}$$

est l'erreur de re-projection d'un point $\mathcal{Q}_i$ appartenant aux modèles des bâtiments et observé par la $j^{ème}$ caméra clé.

Afin d'assurer plus de robustesse quand peu de bâtiments sont observés, Tamaazousti et al. (2011) proposent de prendre en compte à la fois les contraintes multi-vues sur l'ensemble des

points reconstruit et les contraintes géométriques apportées par les modèles 3D des bâtiments. La fonction de coût bi-objective suivante a été utilisée :

$$g\left(\{\mathcal{R}_j, \mathbf{t}_j\}_{j=1}^{N}, \{\mathcal{Q}_i\}_{i\in\mathcal{E}} \left\{\mathcal{Q}_i^{h_i}\right\}_{i\in\mathcal{M}}\right) = \sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{D}_i} \rho\left(\|\mathbf{f}_{i,j}\|, c_{\mathcal{E}}\right) + \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{D}_i} \rho\left(\|\mathbf{g}_{i,j}\|, c_{\mathcal{M}}\right). \quad (3.6)$$

Nous rappelons que $\mathbf{f}_{i,j}$ est l'erreur de re-projection standard du $i^{ème}$ point 3D observé par la $j^{ème}$ caméra :

$$\mathbf{f}_{i,j} = \mathbf{q}_{i,j} - \pi(\mathsf{K}\mathcal{R}_j^T \left[\mathsf{I}_{3\times 3}\right| - \mathbf{t}_j] \, \tilde{\mathcal{Q}}_i) \quad (3.7)$$

Toutefois l'utilisation d'une fonction bi-objective implique un problème de pondération des deux termes de la fonction de coût. Pour pallier ce problème, Tamaazousti et al. (2011) proposent d'exploiter le seuil du M-estimateur utilisé. Ce point sera détaillé dans la section suivante.

3.2.3 Optimisation robuste vis à vis des contraintes utilisées

Pour améliorer le respect des contraintes formulées dans la section précédente, une étape d'optimisation des paramètres de la scène est nécessaire. Généralement celle-ci repose sur un ajustement de faisceaux minimisant une des fonctions de coût détaillées ci-dessus. Pour respecter le critère temps réel, les systèmes d'équations peuvent être résolus en exploitant la structure creuse des matrices Hessienne associées aux systèmes en question. Plus de détails sur la méthode de résolution sont présentés en Annexe (section A.2.1).

Par ailleurs, afin de limiter les problèmes de convergence en cas de mauvaises identifications de contraintes (*e.g.* segmentation erronée, mauvaise association point/plan), Lothe et al. (2009) et Tamaazousti et al. (2011) proposent d'utiliser, d'une part, un M-estimateur et, d'autre part, de mettre en place un processus itératif d'optimisation. Dans la suite nous détaillerons ces deux points.

Utilisation des M-estimateurs Une mauvaise segmentation du nuage de points et une association point/plan erronée représentent deux sources de données aberrantes. Pour assurer plus de robustesse face à ces données et améliorer la convergence de l'ajustement de faisceaux, Lothe et al. (2009) et Tamaazousti et al. (2011) ont eu recours à l'utilisation d'un estimateur robuste, plus préciseraient celui de Geman-McClure. Ainsi chaque erreur de re-projection r est remplacée par la valeur $\rho(r, c)$ définie par :

$$\begin{aligned} \rho(r,c) : \mathbb{R} &\longrightarrow [0..1] \\ r &\longrightarrow \frac{r^2}{(r^2+c^2)}, \end{aligned} \quad (3.8)$$

où c est le seuil de rejet du M-estimateur.

Pour la solution introduite par Lothe et al. (2009), le seuil c est estimé directement comme suit :

$$c = \text{médiane}(\mathbf{r}) + 5.2 \times \text{MAD}(\mathbf{r}), \quad (3.9)$$

où $\mathbf{r} = (r_1 \dots r_m)^T$ est le vecteur des résidus de taille m concaténant les erreurs de re-projection de tous les points 3D pris en compte dans l'optimisation (*i.e.* dans le cas de Lothe et al. (2009) seuls les points qui ont un plan associé) alors que la fonction $\text{MAD}(\mathbf{r})$ est définie par :

$$\text{MAD}(\mathbf{r}) = \text{médiane} \begin{pmatrix} |r_0 - \text{médiane}(\mathbf{r})| \\ \cdot \\ \cdot \\ |r_m - \text{médiane}(\mathbf{r})| \end{pmatrix}. \tag{3.10}$$

En ce qui concerne la solution introduite par Tamaazousti et al. (2011), en analysant les distributions des erreurs de re-projections des points contraints et non contraints par les bâtiments, ils concluent que celles-ci sont généralement différentes. En particulier, si le modèle 3D est peu précis, l'erreur de re-projection des points contraints risque d'être plus importante. Dans un tel cas de figure, si les points non contraints sont plus nombreux que les points contraints, un seuil estimé à partir de la distribution globale risquerait de classer l'ensemble des points contraints comme *outliers*. C'est pourquoi Tamaazousti et al. (2011) proposent d'estimer deux seuils et de conserver le plus grand :

$$c = \max(c_{\mathcal{E}}, c_{\mathcal{M}}), \tag{3.11}$$

tel que

- ▷ $c_{\mathcal{E}} = \text{médiane}(\mathbf{r}_{\mathcal{E}}) + 5.2 \times \text{MAD}(\mathbf{r}_{\mathcal{E}})$ le seuil de rejet estimé sur les résidus de la partie inconnue de l'environnement.
- ▷ $c_{\mathcal{M}} = \text{médiane}(\mathbf{r}_{\mathcal{M}}) + 5.2 \times \text{MAD}(\mathbf{r}_{\mathcal{M}})$ le seuil de rejet estimé sur les résidus des points associés aux modèles.

En général, ce seuil c correspond à $c_{\mathcal{M}}$. Plus de détails sur le choix de ce seuil sont présentés dans Tamaazousti et al. (2011).

Processus d'optimisation itératif Le processus de segmentation du nuage de points et d'association point/plan exploitent principalement la pose estimée de la caméra. La qualité de ces deux étapes dépend donc de la qualité de la pose estimée. Par conséquent, si la pose est peu précise, des erreurs de segmentation et d'associations risquent de survenir et ainsi de limiter la convergence de l'ajustement de faisceaux à suivre. Par contre, à l'issue de l'ajustement de faisceaux, une meilleure pose est disponible et permet donc d'effectuer une meilleure segmentation. C'est pourquoi un processus itératif d'optimisation où la minimisation de la fonction de coût et la segmentation du nuage de point 3D ainsi que l'association des point/plan sont alternées. Ce processus est répété jusqu'à la réalisation du critère d'arrêt. Ce dernier correspond à une décroissance du MAD, calculé à partir des erreurs de re-projection des points associés aux modèles des bâtiments, inférieure à un certain seuil. Le processus d'optimisation proposé par Tamaazousti et al. (2011) est synthétisé dans l'algorithme 1.

3.2.4 Limitations

Afin d'évaluer les méthodes proposées par Lothe et al. (2009) et Tamaazousti et al. (2011), nous avons testé ces deux approches d'une manière hors ligne sur une séquence réelle (2400m) enregistrée dans la ville de Versailles (milieu urbain dense). Pour ceci, une première reconstruction (voir figure 3.11) est obtenue à travers la fusion du SLAM avec le GPS introduite par Lhuillier (2012) et qui sera détaillée dans la deuxième partie de ce chapitre (voir section 3.3). Une fois la reconstruction initiale obtenue, un ajustement de faisceaux global contraint aux modèles des bâtiments est appliqué. La figure 3.5(a) représente le résultat final obtenue avec l'ajustement de faisceaux contraint proposé par Lothe et al. (2009). La figure 3.5(b) représente,

$\texttt{nouveauMAD} = 0$;
$\texttt{ancienMAD} = 0$;
repeat
  Segmenter le nuage de points en $\{\mathcal{Q}_i\}_{i\in\mathcal{E}}$ et $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  foreach $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ **do**
    Associer le point $\mathcal{Q}_i$ au plan le plus proche $\mathbf{\Pi}^{h_i}$;
    Projeter orthogonalement le point $\mathcal{Q}_i$ sur sa façade $\mathbf{\Pi}^{h_i}$;
  end
  $\texttt{ancienMAD} = \texttt{MAD}$ calculé sur les erreurs de re-projection de points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Calculer le seuil de rejet du M-estimateur c ;
  Minimiser la fonction de coût considérée en utilisant l'algorithme de Levenberg Marquardt ;
  $\texttt{nouveauMAD} = \texttt{MAD}$ calculé sur les erreurs de re-projection de points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Triangulation des points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ en prenant en compte les nouvelles poses de la caméra ;
until $(\texttt{nouveauMAD} - \texttt{ancienMAD}) < 0.1$;

Algorithme 1 : Différentes étapes du processus d'optimisation du SLAM contraint aux modèles des bâtiments proposé par Tamaazousti et al. (2011).

quant à elle, la reconstruction SLAM obtenue après l'ajustement de faisceaux contraint proposé par Tamaazousti et al. (2011). Dans cette section, l'évaluation sera limitée à l'analyse qualitative des localisations obtenues et des re-projections des modèles sur un ensemble d'images extraites de la séquence utilisée.

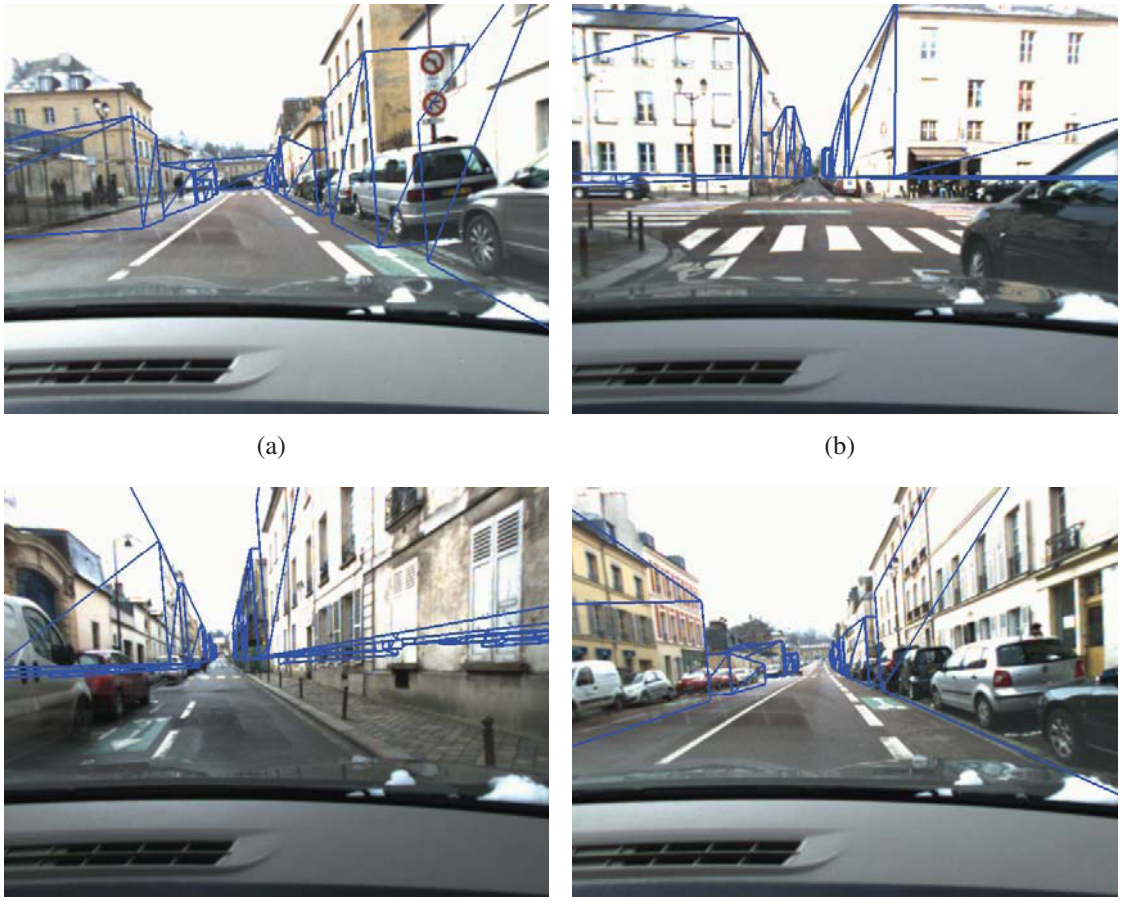
Comme le montre la figure 3.5(a), exploiter uniquement des points 3D associés aux modèles des bâtiments rend la solution proposée par Lothe et al. (2009) sensible au manque de contraintes fournies par les bâtiments. En effet, les agrandissements de la figure 3.5(a) mettent en évidence la localisation erronée dans les zones où peu de bâtiments sont visibles. Ceci implique que cette solution est restrictive aux zones urbaines denses. Grâce à l'exploitation des contraintes multi-vues fournies par la partie inconnue de l'environnement, la solution proposée par Tamaazousti et al. (2011) offre plus de robustesse face au problème mentionné ce-dessus (voir figure 3.5(b)). Malgré cette amélioration, nous notons une importante limitation commune aux deux approches. En effet, les bâtiments permettent de contraindre principalement la position *dans le plan* de la caméra et son angle lacet. Ainsi, optimiser les six degrés de liberté de la caméra peut entrainer des problèmes de convergence visibles sur les degrés de liberté restants surtout si la reconstruction initiale est éloignée de la solution optimale. Ces dérives sont mises en évidence dans la figure 3.6 où les modèles des bâtiments sont re-projetés sur des images extraites de la séquence utilisée.

En évaluant l'algorithme de Tamaazousti et al. (2011) dans le cadre d'une localisation en ligne dans un processus SLAM avec ajustement de faisceaux local, nous constatons que cette solution échoue rapidement comme le montre la figure 3.7. Une mauvaise segmentation du nuage de points, une association point/plan erronée ainsi que l'imprécision des modèles utilisés représentent sans doute les principales causes de cet échec. En effet, étant donné que la contrainte introduite par Tamaazousti et al. (2011) sur la reconstruction est une *contrainte dure* (*i.e.* réduction de nombre de degrés de liberté), cette solution demeure sensible aux imprécisions des modèles 3D des bâtiments utilisées ce qui peut entrainer une mauvaise estimation du facteur d'échelle. Par ailleurs, contrairement à un ajustement de faisceaux global où chaque pose de ca-

(a)

(b)

FIGURE 3.5 – **Comparaison entre les résultats obtenus, hors ligne, avec l'ajustement de faisceaux global proposé par Tamaazousti et al. (2011) et celui introduit par Lothe et al. (2009).** En rouge : le nuage de point 3D. En vert : les poses de la caméra. (a) Résultat du raffinement global d'une reconstruction SLAM avec l'approche de Lothe et al. (2009). (b) Résultat du raffinement global d'une reconstruction SLAM avec l'approche de Tamaazousti et al. (2011). La reconstruction SLAM initiale utilisée est obtenue avec la méthode de Lhuillier (2012) qui sera détaillée dans la section 3.3



(a) (b)

(c) (d)

FIGURE 3.6 – **Re-projection des modèles 3D des bâtiments sur un ensembles d'images extraite de la séquence de Versailles.** Ces résultats mettent en évidence les imprécisions des degrés de liberté *hors plan* à l'issue d'un ajustement de faisceaux contraint aux modèles 3D des bâtiments introduit par Lothe et al. (2009). Notons que des résultats équivalents sont obtenus à l'issue de l'ajustement de faisceaux contraint aux modèles 3D des bâtiments introduit par Tamaazousti et al. (2011).

méra bénéficie des contraintes liées aux poses antérieures et suivantes, lors d'un ajustement de faisceaux local, chaque pose de caméra bénéficie uniquement des contraintes antérieures. Ces contraintes disponibles peuvent alors être insuffisantes pour corriger la reconstruction SLAM à l'instant présent (*e.g.* quand peu de bâtiments sont visibles *etc.*), d'où les dérives au niveau des degrés de liberté *dans le plan*. Les degrés de liberté *hors plan* étant non contraints par les bâtiments, ils peuvent subir alors une importante dérive (principalement l'altitude). Ces imprécisions sur la pose de la caméra perturbent l'étape de segmentation et l'association point/plan et donc la convergence de l'ajustement de faisceaux contraint. Tous ces facteurs réunis entrainent l'échec de l'approche de Tamaazousti et al. (2011) en ligne dans le contexte urbain.

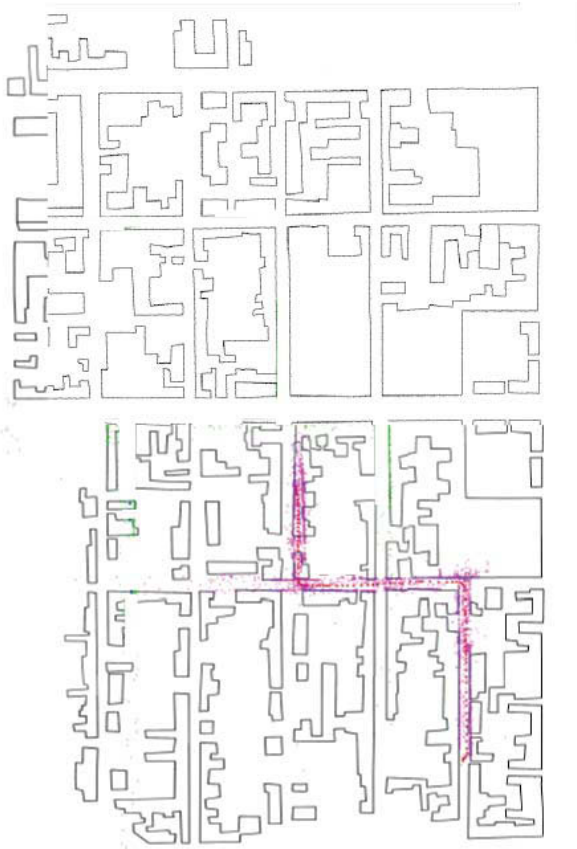


FIGURE 3.7 – **Résultat de la localisation en ligne obtenue avec l'approche de Tamaazousti et al. (2011).** Pour la séquence utilisée, la solution échoue.

Ces deux différentes expérimentations montrent clairement qu'intégrer les contraintes géométriques apportées par les modèles 3D des bâtiments dans l'ajustement de faisceaux permet d'améliorer la précision d'une reconstruction SLAM. Cependant, ces solutions restent limitées à une exploitation hors ligne en milieux urbains denses. En effet, leurs précisions se dégradent considérablement dans les milieux péri-urbains où peu de bâtiments sont observés. Par ailleurs, nous avons remarqué qu'à cause du manque de contraintes disponibles localement, il est difficile d'assurer une localisation en ligne et précise en se basant uniquement sur les modèles 3D grossiers des bâtiments.

3.3 SLAM contraint aux données GPS : Contraindre la trajectoire

Dans cette section, nous nous intéressons aux approches contraignant explicitement la trajectoire fournie par le SLAM à l'aide des données issues d'un capteur embarqué : le GPS. Comme c'est le cas pour la contrainte sur la reconstruction, pour contraindre la trajectoire de la caméra trois étapes sont nécessaires : établissement de contrainte, formalisation de la contrainte et optimisation vis à vis de la contrainte.

Pour la clarté des équations, nous noterons, dans cette section, $\kappa = (\left\{\alpha_j, \beta_j, \gamma_j, \mathbf{t}_j^T\right\}_{j=1}^{N}, \{X_i, Y_i, Z_i\}_{i=1}^{M})$ le vecteur de paramètres à optimiser. $(\alpha_j, \beta_j, \gamma_j)$ sont les trois angles euler correspondant à l'orientation $\mathcal{R}_j$ de la $j^{ème}$ caméra, $\mathbf{t}_j$ est la position 3D et (X_i, Y_i, Z_i) sont les coordonnées du point $\mathcal{Q}_i$.

3.3.1 Établissement de la contrainte aux données GPS : association image clé/mesure GPS

Le principe de la contrainte apportée par le GPS consiste tout simplement à considérer que la trajectoire, dans le plan de la route, estimée par le SLAM doit suivre au plus proche la trajectoire fournie par le GPS. En effet, si nous considérons que la caméra est rigidement liée au GPS et que la translation entre ces deux capteurs est négligeable, alors le respect de cette propriété peut être observé en étudiant le décalage entre les données GPS et les positions *dans le plan* de la caméra.

Dans ces conditions, la contrainte au GPS implique qu'à chaque pose de la caméra nous associons une mesure GPS. Ceci suppose donc que les données issues des deux capteurs sont synchronisées. Or, le GPS a une fréquence de fonctionnement nettement inférieure à celle d'une caméra 30Hz et ces deux capteurs fournissent généralement des données de manière asynchrone. Pour pouvoir néanmoins associer une donnée GPS à une image clé, nous proposons d'exploiter un système de datation des données. Dans la suite, nous considérerons que cette datation est réalisée par un processus externe dont le fonctionnement sort du cadre de cette thèse. La donnée GPS associée à une image clé est alors sélectionnée selon un critère de proximité temporelle. Dans le cas où aucune donnée GPS proche de la date de l'image clé n'est disponible (*e.g.* zone où trop peu de satellites sont visibles), l'image clé en question ne sera associée à aucune donnée GPS et ne sera donc pas contrainte dans l'ajustement de faisceaux.

3.3.2 Formalisation de la contrainte apportée par les données GPS

Dans la littérature, contraindre la trajectoire SLAM en exploitant les mesures GPS revient généralement à introduire dans la fonction de coût de l'ajustement de faisceaux un nouveau terme d'accroche aux données mesurant l'écart entre la position *dans le plan* de chaque image clé et la donnée GPS associée, un terme dont l'influence peut être ajustée au travers d'un poids ω. Dans ce cas nous parlons d'*une contrainte douce*. La fonction de coût résultante a donc la forme suivante :

$$f_{GPS}(\kappa) = f(\kappa) + \omega \times \|\mathsf{M}\kappa - \mathbf{v}\|^2, \tag{3.12}$$

avec $\mathbf{v} = (\mathbf{v}_1^T, .., \mathbf{v}_N^T)^T$ le vecteur contenant toutes les données GPS, $\mathbf{v}_j = (x_j^{gps}, y_j^{gps})^T$, utilisées. La matrice M permet de récupérer les positions *dans le plan* de la caméra. Elle est définie par $\mathsf{M} = (\mathsf{D}_{2N\times 6N}|0_{2N\times 3M})$ tel que $\mathsf{D}_{2N\times 6N}$ est une matrice diagonale par bloc où chaque bloc D_j de taille (2×6) est donné par

$$\mathsf{D}_j = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.13}$$

$f(\kappa)$ est la fonction de coût standard basée sur l'erreur de re-projection détaillée dans la section 1.3.2.5.

Pour assurer une bonne précision de localisation, le poids de la contrainte GPS ω doit prendre en compte l'incertitude de ce capteur ($\sim$ 10m pour un GPS grand publique) et la variation de celle-ci au cours du temps (Hideyuki et al. (2010)).

Toutefois, la présence de données aberrantes ($\gg$ 10m), difficilement identifiables, peut rendre l'estimation de ce poids incertain. Ceci risque de dégrader la contrainte multi-vues au point d'entrainer l'échec de l'algorithme SLAM. Pour faire face à ce problème, Lhuillier (2012) propose une formulation différente du problème : minimiser l'erreur euclidienne entre la position *dans le plan* de la caméra estimée par le SLAM et la donnée GPS tout en vérifiant que cette contrainte ne dégrade pas l'erreur de re-projection standard $f(\kappa)$ au delà d'un certain seuil e_t. En d'autres termes, Lhuillier (2012) cherche le vecteur de paramètres κ tel que :

$$\kappa = \begin{cases} \underset{\kappa}{argmin} \, \|\mathsf{M}\kappa - \mathbf{v}\|^2 \\ f(\kappa) < e_t \end{cases} \tag{3.14}$$

Pour estimer le vecteur κ Lhuillier (2012) propose de reformuler l'équation 3.14 à travers une unique fonction de coût qui ajoute au terme d'accroche aux données un terme de régularisation interdisant toute dégradation de l'erreur de re-projection au-delà du seuil prédéfini e_t. Ce terme de régularisation, calculé en se basant sur l'erreur de la re-projection standard $f(\kappa)$, a une valeur négligeable lorsque la contrainte est respectée et tend vers l'infini lorsque la contrainte est proche d'être brisée. La fonction de coût résultante est alors donnée par :

$$f_I(\kappa) = \frac{\omega}{e_t - f(\kappa)} + \|\mathsf{M}\kappa - \mathbf{v}\|^2, \tag{3.15}$$

ω est une constante fixée de manière à attribuer un poids négligeable au terme de régularisation face au terme d'accroche aux données lorsque la dégradation de l'erreur de re-projection est faible. L'ajustement de faisceaux intégrant une telle fonction de coût sera désigné par *ajustement de faisceaux avec contrainte d'inégalité*.

3.3.3 Optimisation vis à vis des contraintes utilisées

Pour réaliser l'ajustement de faisceaux avec contrainte d'inégalité, deux étapes sont nécessaires. La première étape se résume à un ajustement de faisceaux classique où l'erreur de re-projection standard $f(\kappa)$ est minimisée. Ensuite, une seconde optimisation non linéaire de $f_I(\kappa)$ est effectuée. La première étape permet d'estimer le poids de la contrainte GPS ω ainsi que la dégradation maximale tolérée sur l'erreur de re-projection e_t.

En effet, Lhuillier (2012) propose de déterminer ω de la manière suivante :

$$\omega = \frac{e_t - f(\kappa^*)}{10} \times \|\mathsf{M}\kappa^* - \mathbf{v}\|^2, \tag{3.16}$$

où κ^* représente le vecteur des paramètres à l'issue de la première étape d'optimisation en utilisant l'ajustement de faisceaux classique. Le seuil e_t est estimé quant à lui de la manière suivante :

$$e_t = 1.05 \times f(\kappa^*). \tag{3.17}$$

Pour mieux comprendre le principe de l'ajustement de faisceaux avec la contrainte d'inégalité, les différentes étapes de l'optimisation sont résumées dans l'algorithme 2.

Réaliser un premier ajustement de faisceaux en minimisant la fonction de coût standard ;
Calculer la dégradation maximale tolérée $e_t = 1.05 \times f(\kappa^*)$;
Calculer le poids ω ;
En utilisant l'algorithme de Levenberg Marquardt, minimiser la fonction de coût $f_I(\kappa)$ définie par l'équation 3.15 **sous la contrainte** que $(f(\kappa) < e_t)$;

Algorithme 2 : Ajustement de faisceaux avec contrainte d'inégalité.

Afin de minimiser la fonction de coût avec la contrainte d'inégalité introduite par Lhuillier (2012), il est indispensable de déterminer la nouvelle matrice Hessienne H_I et le vecteur gradient $\mathbf{g}_I$ associés. Les différents calculs donnent :

$$\mathbf{g}_I = \frac{2\omega}{(e_t - f)^2}\mathbf{g} + 2\mathsf{M}^T(\mathsf{M}\kappa - \mathbf{v}) \tag{3.18}$$

et

$$\mathsf{H}_I = \frac{2\omega}{(e_t - f)^3}\left[4\mathbf{g}\mathbf{g}^T + (e_t - f)\,\mathsf{J}^T\mathsf{J}\right] + 2\mathsf{M}^T\mathsf{M}, \tag{3.19}$$

avec $\mathbf{g}$ et J sont respectivement le vecteur gradient et la Jacobienne associés à la fonction de coût standard f. Étant donné que le terme $\mathbf{g}\mathbf{g}^T$ n'est pas éparse, la Hessienne H_I a une structure dense. Afin de contourner ce problème, Lhuillier (2012) propose une nouvelle écriture du système à résoudre qui permet de bénéficier d'une structure creuse et donc d'une résolution temps réel. Les détails des calculs des dérivées ainsi que la résolution du système en question sont présentés dans l'annexe (section A.3.1).

3.3.4 Limitations

Afin d'évaluer la contrainte sur la trajectoire proposée par Lhuillier (2012), nous testons cette approche sur deux séquences.

La première est une séquence de synthèse illustrée dans les figures 3.8(b) et 3.8(c). Pour simuler les données GPS, les positions *dans le plan* de la caméra, fournies par la vérité terrain, sont perturbées dans un premier temps (zone 1 de la figure 3.8(a)) par un bruit Gaussien d'amplitude 0.5m. Ensuite, dans la zone 2 de la figure 3.8(a), nous ajoutons au bruit Gaussien un biais constant par morceaux d'amplitude de 3m.

La deuxième séquence utilisée est une séquence réelle (4000m) enregistrée dans des conditions de conduite normale dans la ville de Saint Quentin en Yvelines (milieu péri-urbain).

L'évaluation sur la séquence de synthèse est réalisée en analysant l'évolution de l'erreur de la localisation *dans le plan*. Ceci permettra de conclure quant à la robustesse de l'approche face aux incertitudes du GPS. L'évaluation sur la séquence réelle sera qualitative en se basant sur la localisation obtenue.

Intégrant la contrainte GPS directement dans l'ajustement de faisceaux avec la contrainte d'inégalité permet de géo-localiser en ligne la reconstruction SLAM tout en limitant les dérives

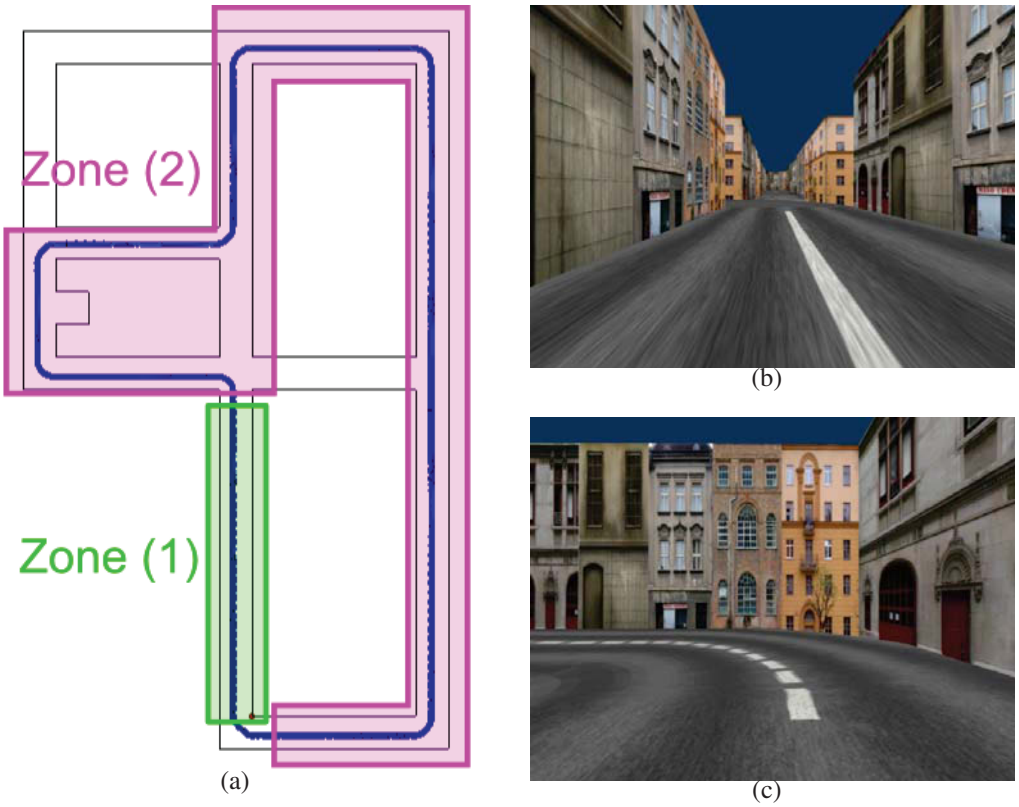


FIGURE 3.8 – **Illustrations de la séquence de synthèse utilisée.**(a) En noir vue de dessus du modèles 3D des bâtiments, en bleu la trajectoire réelle de la caméra. (b), (c), Illustrations de la séquence utilisée.

dues à la mauvaise estimation du facteur d'échelle et l'accumulation des erreurs comme le montre la figure 3.10(a). Toutefois, bien que l'ajout de cette information permet de contraindre efficacement la position *dans le plan* de la caméra et son angle lacet, nous remarquons des dérives importantes au niveau des degrés de liberté restants de la caméra à savoir son altitude ainsi que ses angles roulis et tangage (voir figure 3.10(b)). Par ailleurs, l'évolution de l'erreur de la localisation *dans le plan* représentée dans la figure 3.9 montre que la précision *dans le plan* reste sensiblement liée à celle du GPS. En effet, quand le biais sur les données GPS passe de 0m à 3m, l'erreur moyenne de la localisation *dans le plan* passe de 0.5m dans la zone (1) à 3.2m dans la zone (2). Par conséquent, cette approche n'est pas adaptée aux milieux urbains denses où la précision du GPS se dégrade considérablement alors que les besoins en précision s'accroissent (voir figure 3.11).

3.4 Bilan

Dans ce chapitre, nous avons détaillé deux familles de SLAM contraint à travers les méthodes proposées par Lothe et al. (2009), Tamaazousti et al. (2011) et Lhuillier (2012). La première intègre une contrainte sur la reconstruction fournie par les modèles 3D des bâtiments. La deuxième se base sur une contrainte sur la trajectoire en exploitant les données GPS. Nous avons également présenté deux façons pour intégrer une contrainte supplémentaire à un ajustement de faisceaux : *une contrainte dure* en réduisant le nombre de degrés de liberté à optimiser ou *une*

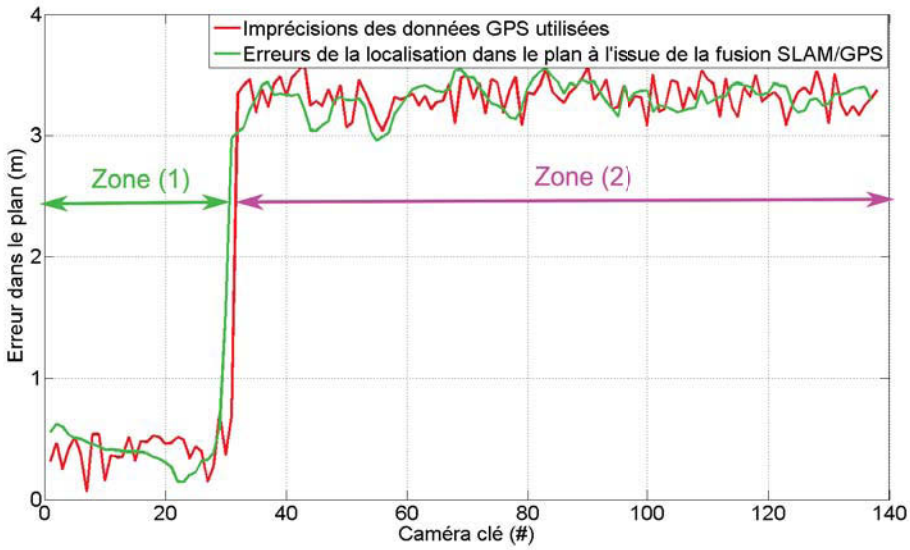
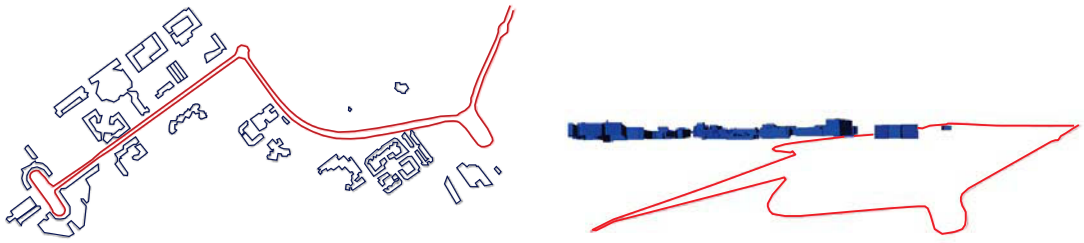


FIGURE 3.9 – Évaluation de l'algorithme de Lhuillier (2012) sur la séquence de synthèse : évolution de l'erreur de la localisation *dans le plan*.



(a) (b)

FIGURE 3.10 – **Résultat de la localisation en ligne du SLAM avec l'ajustement de faisceaux avec la contrainte d'inégalité.** En rouge : la localisation obtenue. En bleu : les modèles 3D des bâtiments. (a) Vue de dessus de la localisation : une bonne précision dans le plan. (b) Vue de côté de la localisation : une dérive importante au niveau de l'altitude.

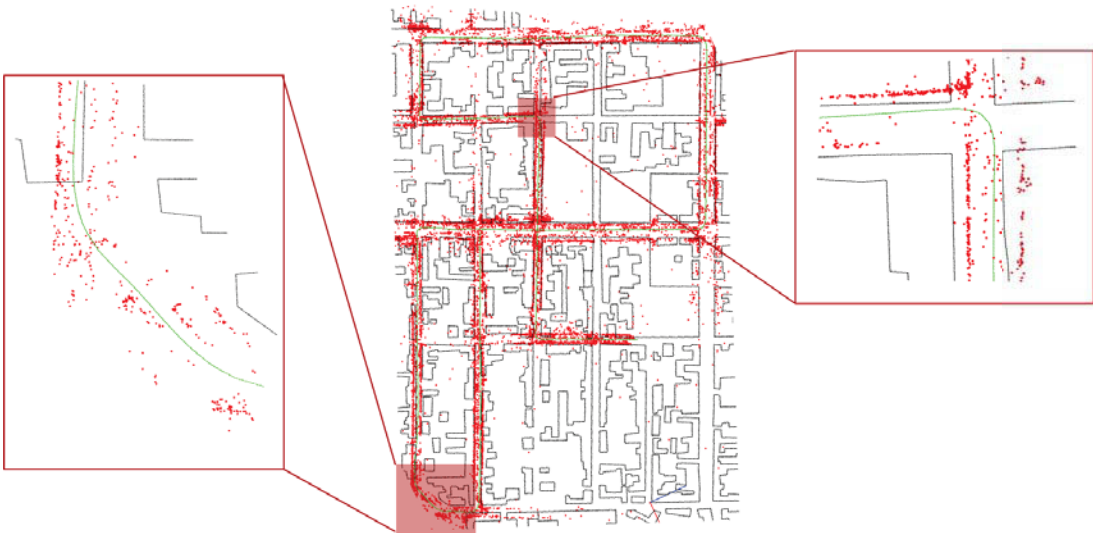


FIGURE 3.11 – **Résultat de la localisation en ligne du SLAM avec l'ajustement de faisceaux avec la contrainte d'inégalité dans un milieu urbain dense.** En vert : la localisation obtenue. Des erreurs de localisation sont notables à cause de l'imprécision du GPS.

contrainte douce via un terme d'accroche aux données. Ces différentes contraintes permettent de limiter les dérives du SLAM principalement sur les degrés de liberté *dans le plan* et fournissent une reconstruction géo-référencée. Toutefois, les expérimentations sur des séquences réelles ont mis en évidence les limitations de chacune de ces approches. En effet, tandis que le SLAM contraint aux modèles 3D des bâtiments est limité à une utilisation dans les milieux urbains denses, le SLAM contraint aux données GPS est limité, quant à lui, aux milieux péri-urbains et ruraux. De plus, nous avons remarqué que les deux approches dérivent au niveau des degrés de liberté mal contraint principalement l'altitude.

Dans la partie suivante nous allons proposer diverses améliorations à chacune des deux approches, notamment en intégrant une contrainte supplémentaire apportée par les Modèles d'Elévation de Terrain (MET). Ensuite, nous allons nous intéresser, dans la deuxième partie de ce mémoire, à combiner les différentes contraintes fournies par le GPS, les MET et les modèles 3D des bâtiments dans un même processus d'optimisation pour obtenir une solution plus précise et fonctionnant dans un plus vaste domaine d'applications (urbain dense et péri urbain, *i.e.* les lieux où la précision est nécessaire).

Première partie

Intégration des contraintes fournies par le MET dans un SLAM contraint pour une géo-localisation plus précise sur les 6DoF de la caméra

Contenu de la partie

Présentation des méthodes proposées **69**

4 SLAM contraint au MET et aux modèles 3D des bâtiments pour une localisation en ligne **71**

4.1 SLAM contraint au MET 71
4.2 SLAM contraint au MET et aux modèles 3D des bâtiments 75
4.3 Segmentation du nuage de points 78
4.4 Évaluation expérimentale 82
4.5 Conclusion et perspectives 95

5 SLAM contraint aux données GPS et au MET **97**

5.1 Introduction 97
5.2 Fusion de la contrainte aux données GPS avec *une contrainte dure en altitude* . 98
5.3 Fusion de la contrainte aux données GPS avec *une contrainte douce en altitude* 100
5.4 Évaluation expérimentale 102
5.5 Conclusion et perspectives 108

Bilan **109**

Présentation des méthodes proposées

Dans la première partie de ce mémoire, nous allons étudier deux différentes approches pour estimer en ligne les six degrés de liberté d'une caméra embarquée sur le véhicule. Elles s'appuient sur les méthodes d'ajustement de faisceaux contraint détaillées dans le chapitre précédent (voir chapitre 3). La première solution consiste à intégrer une contrainte supplémentaire fournie par le MET au SLAM contraint aux modèles 3D des bâtiments. La deuxième fusionne les contraintes fournies par le MET avec celles apportées par le GPS.

Objectif détaillé de l'étude réalisée

Comme nous l'avons montré dans le chapitre 3, intégrer des informations absolues telles que les mesures GPS ou les modèles 3D des bâtiments dans le processus d'optimisation du SLAM, permet d'avoir une reconstruction géo-référencée tout en limitant les dérives du SLAM dues aux accumulations des erreurs et à la mauvaise estimation du facteur d'échelle. Toutefois, ces méthodes restent insuffisantes pour assurer une bonne précision sur les six degrés de liberté de la caméra. En effet, ces données permettent de contraindre principalement les degrés de liberté *dans le plan* à savoir la position 2D de la caméra projetée sur le plan de la route et son angle lacet. Ainsi, optimiser les six degrés de liberté de la caméra en exploitant uniquement ces données ne permet pas de corriger les dérives des degrés de liberté *hors plan*, c'est-à-dire l'altitude de la caméra et ses angles roulis et tangage, voire même perturbe leur estimation. Pour faire face à ce problème, nous proposons d'étendre les solutions de Tamaazousti et al. (2011) et Lhuillier (2012) en intégrant une contrainte supplémentaire fournie par le MET. Le choix du MET se justifie premièrement par sa disponibilité à la fois dans les milieux urbains denses et les milieux péri-urbains. De plus, le MET a également l'avantage de nous informer sur les degrés de liberté *hors plan* de la caméra. En effet, étant donné que la caméra est rigidement liée dans le véhicule, son altitude et ses angles roulis et tangage peuvent être considérés constants par rapport au plan de la route.

Dans ce qui suit, nous allons présenter le principe général de l'intégration de la contrainte du MET dans un ajustement de faisceaux contraint afin d'assurer une estimation en ligne et plus précise des six degrés de liberté de la caméra.

Vue d'ensemble : Intégration de la contrainte du MET

Prendre en compte la contrainte fournie par le MET (dont les propriétés sont détaillées dans la section 1.4.2.2) dans l'ajustement de faisceaux contraint aux bâtiments ou celui contraint aux données GPS implique l'intégration de certaines modifications que nous pouvons classer dans les trois points suivants :

- ▷ **Association caméra/route.** Afin d'établir la contrainte apportée par le MET pour chaque pose de caméra, une étape supplémentaire est exigée. Celle-ci consiste à associer, à chaque image clé, la caméra au plan de route auquel elle est supposée appartenir ;
- ▷ **Nouvelles fonctions de coût.** Une fois la contrainte MET établie à travers les associations caméra/route, sa prise en compte dans le processus d'optimisation du SLAM contraint peut être réalisée à travers la minimisation d'une nouvelle fonction de coût. Dans la suite, deux approches ont été adoptées pour introduire l'information d'altitude dans la fonction de coût : soit sous forme d'*une contrainte dure* soit sous forme d'*une contrainte douce* ;
- ▷ **Processus itératif.** Pour une meilleure convergence, nous proposons un processus d'optimisation itératif où les associations caméra/route sont remises en cause après l'étape de la minimisation de la fonction de coût.

CHAPITRE 4

SLAM contraint au MET et aux modèles 3D des bâtiments pour une localisation en ligne

Dans ce chapitre, nous proposons une première solution en ligne pour estimer avec précision les six degrés de liberté d'une caméra mobile dans un milieu urbain. Cette solution étend l'approche proposée par Tamaazousti et al. (2011) qui exploite les modèles 3D des bâtiments en lui intégrant une contrainte supplémentaire fournie par le Modèle d'Elévation de Terrain (MET). Tandis que la géométrie de la scène est contrainte par les modèles des bâtiments, le MET permet de contraindre la trajectoire de la caméra et principalement ses paramètres hors plan. La fusion des contraintes se fait au niveau de l'ajustement de faisceaux afin d'assurer une meilleure estimation des six degrés de liberté de la caméra. Des améliorations au niveau de la méthode de segmentation du nuage de points utilisée par Tamaazousti et al. (2011) sont également proposées pour mieux l'adapter au contexte urbain.

Nous commencerons ce chapitre par introduire la contrainte fournie par le MET (voir la section 4.1). Dans la section 4.2, cette nouvelle contrainte est intégrée dans l'ajustement de faisceaux contraint aux modèles 3D des bâtiments proposé par Tamaazousti et al. (2011). Pour apporter plus de robustesse à notre approche, nous présenterons, dans la section 4.3, des améliorations à la méthode de segmentation du nuage de points et d'association point/plan utilisé par Tamaazousti et al. (2011).

Ces travaux ont donné lieu à une publication internationale Larnaout et al. (2012).

4.1 SLAM contraint au MET

Afin d'assurer une localisation basée vision dans un milieu urbain, nous exploitons dans le cadre de nos travaux les acquisitions fournies par une caméra rigidement embarquée dans un véhicule. Cette assertion implique, qu'au cours du déplacement du véhicule, la caméra est supposée garder non seulement la même altitude mais aussi les mêmes angles roulis et tangage par rapport au plan de la route. Même si les amortisseurs du véhicule peuvent perturber ces paramètres, nous supposerons dans la suite que ces variations sont négligeables.

Adopter cette hypothèse permet de bénéficier de deux avantages. Premièrement, celle-ci ne

fait pas de restriction sur la nature du véhicule (*e.g.* une hypothèse de non holonomie sur le déplacement Scaramuzza et al. (2009a)). Elle peut donc être appliquée sur un véhicule quel que soit son type. Deuxièmement, elle fournit des nouvelles contraintes sur la trajectoire de la caméra et plus précisément sur ses degrés de liberté *hors plan*.

Afin de prendre en compte ces contraintes supplémentaires, nous exploitons dans la suite les MET (Modèles d'Elévation de Terrain) dont les spécifications sont détaillées dans la section 1.4.2.2. Dans ces modèles géométriques, chaque route est représentée simplement par un segment 3D schématisant son axe. Cette représentation simplifiée (*i.e.* la normale au plan n'est pas fournie) rend l'estimation des angles roulis et tangage approximative. Or, une erreur minime au niveau de l'orientation de la caméra peut augmenter d'une manière significative l'erreur de re-projection (*i.e.* un effet de bras de levier). Contrairement à l'orientation, une petite erreur en altitude entraine généralement des erreurs de re-projection moins importantes. Par conséquent, nous nous intéresserons, dans la suite, uniquement à l'intégration de la contrainte en altitude dans l'ajustement de faisceaux. En plus de la contrainte explicite de l'altitude de la caméra, l'intégration de cette information dans le processus d'optimisation permet également de contraindre implicitement les angles roulis et tangage (principalement au niveau des virages). Nous montrerons ceci, expérimentalement, dans le chapitre suivant. Le principe de la contrainte explicite en orientation sera, quant à lui, proposée dans le chapitre dédié aux perspectives à la fin de ce mémoire.

Dans la suite de cette section, nous allons détailler les différentes étapes nécessaires à l'intégration de la contrainte d'altitude fournie par le MET dans le processus d'optimisation du SLAM. Nous commencerons alors par expliquer comment cette contrainte est établie à partir des routes du MET (voir section 4.1.1). Ensuite, nous passerons dans la section 4.1.2 à la formulation de la contrainte en question en proposant une nouvelle fonction de coût. Enfin, le processus d'optimisation complet sera décrit dans la section 4.1.3.

4.1.1 Établissement de la contrainte en altitude

Avant toute chose, pour pouvoir introduire une contrainte en altitude à la trajectoire de la caméra, nous devons identifier les manifestations observables de cette contrainte. La véritable altitude dans le repère monde n'étant pas connue, nous proposons d'exploiter le fait que la caméra est rigidement liée au véhicule ainsi sa distance orthogonale h au sol est constante (nous faisons l'hypothèse que les variations de cette distance liées aux amortisseurs sont négligeables). Sous ces hypothèses, la caméra est alors supposée se déplacer sur la surface définie par l'ensemble des points de l'espace situés à une distance orthogonale h par rapport à la route où elle évolue (voir figure 4.1). On peut alors observer si la contrainte d'altitude est respectée en mesurant la distance orthogonale entre la position de la caméra estimée par le SLAM et la surface en question.

Pour estimer cette surface, il est donc nécessaire d'identifier, à chaque image clé, le plan fini de la route, dans le MET, où la caméra se situe. Cette association caméra/route est donc une étape critique. Elle n'est pas aussi évidente qu'elle puisse paraitre étant donnée que la position de la caméra est incertaine à cause des dérives du SLAM et que le réseau routier peut être dense et complexe. Dans nos travaux, nous établissons cette association en se basant sur un critère de proximité. En d'autre termes, chaque image clé est associée à la route la plus proche dans le MET en terme de distance orthogonale.

Dans la section suivante, nous introduisons une nouvelle formalisation de l'erreur de re-projection afin d'assurer le respect de cette contrainte en altitude au cours de l'optimisation.

4.1.2 Formalisation de la contrainte en altitude apportée par le MET

A l'issue de l'étape précédente, chaque image clé de la reconstruction SLAM se trouve associée à la route où elle se situe. Pour introduire la contrainte en altitude dans le processus d'optimisation, nous proposons dans ce qui suit une formulation *dure* qui assure un respect strict de la contrainte en question. Nous parlerons alors de la *contrainte dure en altitude*. Pour ceci, nous commençons par projeter orthogonalement la caméra sur le plan parallèle à sa route associée et situé à une distance h de celle-ci. Pour maintenir cette contrainte au cours de l'ajustement de faisceaux, nous proposons de passer par une nouvelle paramétrisation de la caméra. Cette paramétrisation revient à exprimer la pose de l'image clé dans un nouveau référentiel qui correspond au repère de la route sélectionnée (voir la figure 4.1) et où son altitude est connue.

Dans la suite, nous noterons L^{k_j} la matrice de passage (3×4) du repère monde au repère de plan de la route Λ^{k_j} associée à la $j^{ème}$ caméra clé. Le MET étant fixe, le calcul des matrices de passage $\left\{\mathsf{L}^{k_j}\right\}_{k_j=1}^{K}$ peut être effectué au préalable.

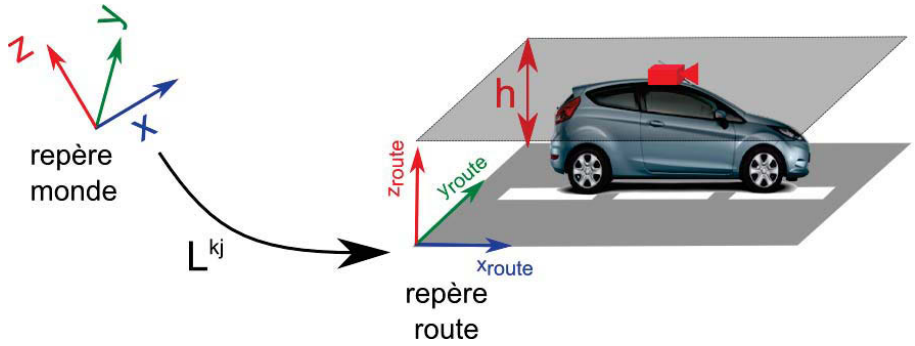


FIGURE 4.1 – **Contrainte en altitude fournie par le MET**. L^{k_j} est la matrice de changement de repère du repère monde au repère de la route associé à la $j^{ème}$ caméra clé.

La nouvelle pose de la $j^{ème}$ caméra clé dans le repère de sa route est donc donnée par :

$$\begin{pmatrix} \mathcal{R}_j^{k_j} & \mathbf{t}_j^{k_j} \\ 0 & 1 \end{pmatrix} = \tilde{\mathsf{L}}^{k_j} \times \begin{pmatrix} \mathcal{R}_j & \mathbf{t}_j \\ 0 & 1 \end{pmatrix}, \tag{4.1}$$

Avec $\tilde{\mathsf{L}}^{k_j}$ est la matrice de passage (4×4) du repère monde au repère de plan de la route Λ^{k_j}.

La caméra est supposée garder une altitude constante et connue dans le repère de la route ce qui implique que $\left(\mathbf{t}_j^{k_j}\right)_z = h$. Ainsi en optimisant la pose de la caméra dans ce repère, seuls cinq degrés de liberté au lieu de six sont raffinés. La fonction de coût résultante pour l'ajustement de faisceaux avec une *contrainte dure en altitude* est donc donnée par :

$$l\left(\left\{\mathcal{R}_j^{k_j}, \mathbf{t}_j^{k_j}\right\}_{j=1}^{N}, \{\mathcal{Q}_i\}_{i=1}^{M}\right) = \sum_{i=1}^{M} \sum_{j \in \mathcal{D}_i} \rho\left(\|\mathbf{l}_{i,j}\|, c\right), \tag{4.2}$$

où

$$\mathbf{l}_{i,j} = \mathbf{q}_{i,j} - \pi\left(\mathsf{K}(\mathcal{R}_j^k)^T \left[\mathsf{I}_{3\times 3}| - \mathbf{t}_j^k\right] \tilde{\mathsf{L}}^{k_j} \tilde{\mathcal{Q}}_i\right). \tag{4.3}$$

est l'erreur de re-projection du $i^{ème}$ point 3D observé par la $j^{ème}$ caméra. Le M-estimateur de Geman-McClure $\rho(r, c) = \frac{r^2}{r^2+c^2}$ est utilisé pour apporter plus de robustesse face aux données aberrantes et aux associations caméra/route erronées. Durant l'optimisation, le nombre de paramètres à raffiner est réduit à $(5 \times N + 3 \times M)$, soit une réduction de N paramètres par

rapport un processus d'ajustement de faisceaux non contraint (Rappelons que N et M sont respectivement le nombre de poses de caméra et le nombre de points 3D optimisés au cours de l'ajustement de faisceaux). Plus de détails sur la minimisation de cette fonction de coût sont disponibles dans l'annexe, section A.2.2. En plus de la réduction de nombre de paramètres à optimiser, cette approche à l'avantage de ne pas nécessiter l'ajout d'un terme de pénalité à la fonction de coût standard de l'ajustement de faisceaux, évitant ainsi tout problème de pondération. De plus, comme nous l'avons précisé, cette approche permet de garantir un respect strict de la contrainte.

4.1.3 Optimisation robuste vis-à-vis des contraintes utilisées

Comme nous l'avons décrit précédemment, initialement, l'altitude de la caméra courante estimée par l'algorithme SLAM ne correspond pas forcément à l'altitude souhaitée. Dans le but de corriger ce paramètre, une étape d'association caméra/route est réalisée avant la minimisation de la fonction de cout 4.2 introduite dans la section précédente. Toutefois, à cause des incertitudes de la localisation, des associations caméra/route erronées peuvent se produire et donc perturber la convergence de l'ajustement du faisceaux contraint. Pour faire face à ce problème, nous proposons de réévaluer cette association après la minimisation de la fonction de coût 4.2. Un processus itératif est donc mis en place où l'association et la minimisation de la fonction de coût sont alternées jusqu'à ce que l'association se stabilise. Les différentes étapes de l'ajustement de faisceaux avec *une contrainte dure en altitude* sont résumées dans l'algorithme 3.

repeat
  foreach ($\mathcal{C}_j$ *participant à l'optimisation*) **do**
    Chercher dans le MET le plan le plus proche $\mathbf{\Lambda}^{k_j}$;
    Corriger l'altitude de la caméra $\mathcal{C}_j$ en fixant $\left(\mathbf{t}_j^{k_j}\right)_z = h$;
    Calculer la nouvelle pose $\mathsf{P}_j^{k_j}$ de la caméra $\mathcal{C}_j$ dans le repère associé au plan de la route $\mathbf{\Lambda}^{k_j}$;
  end
  Calculer le seuil de rejet du M-estimateur c ;
  Minimiser la fonction de coût (Eq. 4.2) en utilisant l'algorithme du Levenberg Marquardt ;
until (*les associations caméra/route soient stabilisées*);

Algorithme 3 : Différentes étapes de l'ajustement de faisceaux *avec une contrainte dure en altitude*.

Nous rappelons que *la contrainte dure en altitude* est appliquée séparément sur chaque caméra clé. Il est donc possible de mélanger les poses de caméra contraintes et d'autres non contraintes dans un même ajustement de faisceaux. Ceci assure plus de flexibilité à la solution proposée vu qu'il est possible de désactiver *la contrainte dure en altitude* dans certaines zones où le MET n'est pas disponible (*e.g.* parking) ou dans le cas d'un déplacement incohérent (*e.g.* dos-d'ânes).

Le processus itératif adopté permet d'assurer une certaine robustesse face aux mauvaises associations caméra/route. Toutefois, ces dernières peuvent être très importantes à cause, notamment, des dérives en facteur d'échelle du SLAM. Dans ce cas, le processus itératif mis en

place n'est plus suffisant. Pour pallier ce problème, nous allons étudier, dans la section suivante, la possibilité de fusionner *la contrainte dure en altitude* avec les contraintes des bâtiments dans l'ajustement de faisceaux. En effet, ces contraintes permettent de contraindre à la fois le facteur d'échelle et la localisation *dans le plan* du SLAM et ainsi limiter les mauvaises associations caméra/route.

4.2 SLAM contraint au MET et aux modèles 3D des bâtiments

L'utilisation du MET dans le processus d'optimisation à travers *la contrainte dure en altitude* introduite ci-dessus permet généralement d'améliorer la précision des poses des images clés, en particulier au niveau des degrés de liberté *hors plan*. Toutefois, cette précision n'est garantie que si l'association caméra/route est précise. Cette dernière reste sensible à l'état initial de la reconstruction SLAM qui est souvent perturbé par une mauvaise estimation du facteur d'échelle. C'est pour cette raison que nous proposons dans cette section d'intégrer, en plus de la contrainte MET, la contrainte apportée par les modèles 3D des bâtiments. En effet, comme nous l'avons montré dans la section 3.2, appliquées au nuage de points reconstruit, les contraintes géométriques fournies par les modèles des bâtiments contraignent efficacement le facteur d'échelle et la position *dans le plan* de la caméra.

Dans la suite, nous nous intéresserons, dans un premier temps, à l'étape de l'établissement de contraintes nécessaires pour l'ajustement de faisceaux contraint en question. Ensuite, nous introduisons une nouvelle fonction de coût tenant en compte à la fois *la contrainte dure en altitude* et la contrainte apportée par les modèles 3D des bâtiments. Le processus complet d'optimisation est par la suite présenté.

Établissement des contraintes. Dans le cas présent, deux types de contraintes sont exploités. La première, apportée par le MET, concerne les poses de la caméra et plus précisément son altitude. La deuxième, fournie par les modèles 3D des bâtiments, concerne la reconstruction de la scène, en d'autres termes le nuage de points 3D.

La contrainte d'altitude (*resp.* la contrainte aux modèles des bâtiments) ne s'appliquant pas nécessairement à l'ensemble des caméras (*resp.* l'ensemble des points 3D), une première étape consiste à identifier les caméras (*resp.* points 3D) pour lesquelles la contrainte s'applique, ainsi que d'identifier pour chaque caméra contrainte (*resp.* point contraint) le plan de la route (*resp.* le plan du modèle de bâtiment) auquel elle (*resp.* il) est supposée appartenir.

Ces étapes de segmentation et d'association correspondent à celles décrites dans la section 3.2.1 pour les points 3D et la section 4.1.1 pour la caméra.

Nous rappelons qu'à l'issue du processus de segmentation du nuage de points, ce dernier sera classé en deux ensembles :

- ▷ Un ensemble de points 3D $\mathcal{M}$ associé au modèle et représentant les points correspondants aux façades des bâtiments.
- ▷ Un ensemble de points 3D $\mathcal{E}$ associé au reste de l'environnement et représentant la partie inconnue de la scène.

Chaque point 3D $\mathcal{Q}_i$ appartenant à l'ensemble $\mathcal{M}$ est donc associé au plan $\mathbf{\Pi}^{h_i}$ (du modèle 3D des bâtiments) qui correspond à la façade qu'il représente. Une fois l'association point/plan réalisée, les points 3D de l'ensemble $\mathcal{M}$ sont corrigés en les projetant sur leurs façades corres-

pondantes. Nous rappelons que les coordonnées du point $\mathcal{Q}_i$ après correction dans le repère du plan $\mathbf{\Pi}^{h_i}$ sont représentées par $\mathcal{Q}_i^h$.

Formulation des contraintes Bien que les contraintes apportées par MET et par les bâtiments reposent toutes les deux sur une réduction du nombre de degrés de libertés, cette réduction s'applique sur des paramètres différents puisque l'un affecte les paramètres de pose des images clés alors que le second s'applique à la position des points 3D. Ces contraintes n'affectant pas les mêmes entités, celles-ci peuvent alors être combinées au sein d'une même fonction de coût en modifiant à la fois la paramétrisation des points et les caméras clé contraintes. La fonction de coût résultante est donnée par :

$$lg\left(\left\{\mathcal{R}_j^{k_j},\mathbf{t}_j^{k_j}\right\}_{j=1}^{N},\{\mathcal{Q}_i\}_{i\in\mathcal{E}}\left\{\mathcal{Q}_i^{h_i}\right\}_{i\in\mathcal{M}}\right)=\sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{D}_i}\rho\left(\|\mathbf{l}_{i,j}\|,c\right)+\sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{D}_i}\rho\left(\|\mathbf{lg}_{i,j}\|,c\right), \quad (4.4)$$

où

$$\mathbf{l}_{i,j}=\mathbf{q}_{i,j}-\pi\left(\mathsf{K}(\mathcal{R}_j^{k_j})^T\left[\mathsf{I}_{3\times3}|-\mathbf{t}_j^{k_j}\right]\tilde{\mathsf{L}}^{k_j}\tilde{\mathcal{Q}}_i\right). \quad (4.5)$$

et

$$\mathbf{lg}_{i,j}=\mathbf{q}_{i,j}-\pi\left(\mathsf{K}(\mathcal{R}_j^{k_j})^T\left[\mathsf{I}_{3\times3}|-\mathbf{t}_j^{k_j}\right]\tilde{\mathsf{L}}^{k_j}(\tilde{\mathsf{M}}^{h_i})^{-1}\tilde{\mathcal{Q}}_i^h\right), \quad (4.6)$$

nous rappelons que $\tilde{\mathsf{M}}^{h_i}$ est la matrice de passage (4×4) du repère du plan $\mathbf{\Pi}^{h_i}$ au repère monde (voir les détails de la définition du repère en question dans la figure 3.4). Plus de détails sur la minimisation de cette fonction de coût sont disponibles dans l'annexe, section A.2.3.

Intégrer ces deux contraintes simultanément dans la fonction de coût permet à la fois d'améliorer l'estimation du facteur d'échelle de la reconstruction SLAM et celle de l'altitude de la caméra. En effet, l'amélioration de l'estimation du facteur d'échelle favorise une meilleure association caméra/route et donc plus de précision à *la contrainte dure en altitude*. De la même façon, une meilleure précision au niveau de l'altitude de la caméra permet d'améliorer la segmentation point/plan et donc d'accroitre la précision dans l'estimation du facteur d'échelle.

Intégration dans le processus d'optimisation. Comme nous l'avons montré précédemment, les deux contraintes fournies par le MET et les modèles 3D des bâtiments sont complémentaires. Elles permettent de contraindre la reconstruction SLAM afin d'assurer une meilleure localisation en ligne. Toutefois la précision de la localisation à l'issue de l'optimisation reste principalement dépendante de la qualité de l'étape préliminaire à savoir la précision des différentes associations établies. Afin d'assurer une convergence optimale, un processus itératif d'optimisation est alors adopté. En d'autres termes, les associations point/plan ainsi que les associations caméra/route sont ré-établies après chaque minimisation de la fonction du coût 4.4. Ce processus est répété jusqu'à la réalisation du critère d'arrêt. Ce dernier correspond à la stabilisation des associations caméra/route et quand la décroissance du MAD calculé à partir des erreurs de re-projection des points associés aux modèles des bâtiments est inférieure à un certain seuil. L'algorithme 4 résume les différentes étapes de l'ajustement de faisceaux contraint proposé.

Discussion Dans cette section, nous avons mis en évidence le double bénéfice en intégrant simultanément les contraintes fournies par les modèles 3D des bâtiments et le MET dans l'ajustement de faisceaux (*i.e.* contraindre le facteur d'échelle et l'altitude de la caméra). Nous avons

$\texttt{nouveauMAD} = 0$;
$\texttt{ancienMAD} = 0$;
repeat
  foreach ($\mathcal{C}_j$ *participant à l'optimisation*) **do**
    Chercher dans le MET le plan le plus proche $\mathbf{\Lambda}^{k_j}$;
    Corriger l'altitude de la caméra $\mathcal{C}_j$;
    Calculer la nouvelle pose de la caméra $\mathcal{C}_j$, $\mathsf{P}_j^{k_j}$ dans le repère associé au plan de la route $\mathbf{\Lambda}^{k_j}$;
  end
  Segmenter le nuage de points en $\{\mathcal{Q}_i\}_{i\in\mathcal{E}}$ et $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  foreach $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ **do**
    Associer le point $\mathcal{Q}_i$ au plan le plus proche $\mathbf{\Pi}^{h_i}$;
    Projeter $\mathcal{Q}_i$ sur le plan correspondant $\mathbf{\Pi}^{h_i}$ des modèles 3D des bâtiments ;
  end
  $\texttt{ancienMAD} = \texttt{MAD}$ calculé sur les erreurs de re-projection de points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Calculer le seuil de rejet du M-estimateur c ;
  Minimiser la fonction de coût 4.4 en utilisant l'algorithme de Levenberg Marquardt ;
  $\texttt{nouveauMAD} = \texttt{MAD}$ calculé sur les erreurs de re-projection de points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Triangulation des points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ en prenant en compte les nouvelles poses de la caméra ;
until $(\texttt{nouveauMAD} - \texttt{ancienMAD}) < 0.1$ *et*
(*les associations caméra/plan de routes soient stabilisées*);

Algorithme 4 : Différentes étapes de l'ajustement de faisceaux contraint au MET et aux modèles 3D des bâtiments.

également mis l'accent sur l'inter-connexion des différentes étapes du processus d'optimisation proposé. En effet, plus l'estimation du facteur d'échelle et de l'altitude de la caméra est bonne, moins les associations point/plan et caméra/route sont erronées et plus la localisation à l'issue du processus d'optimisation est précise. Le processus d'optimisation itératif adopté permet d'assurer une certaine robustesse face aux éventuelles associations caméra/route et point/plan erronées. Toutefois, comme nous l'avons déjà mentionné, cette approche itérative reste insuffisante pour pallier les erreurs significatives des associations principalement les associations point/plan. En effet, ces associations ont un impact considérable sur tout le processus d'optimisation, contrairement aux associations caméra/route donc l'impact est limitée (*i.e.* les variations d'altitude dans un même quartier sont souvent peu importantes). Par conséquent, se baser sur la méthode de segmentation utilisée par Tamaazousti et al. (2011) (voir section 3.2.1) pour réaliser l'association point/plan semble peu judicieux. En effet, cette approche n'est pas adaptée au contexte urbain où les modèles 3D des bâtiments sont peu précis et souvent occultés par des arbres, des voitures garées *etc.*. Dans la section suivante, nous présenterons une nouvelle méthode de segmentation et d'association point/plan permettant plus de précision et de robustesse à notre algorithme de localisation en ligne.

4.3 Segmentation du nuage de points

Pour garantir plus de robustesse et précision à notre algorithme de SLAM contraint au MET et aux modèles 3D des bâtiments, une segmentation du nuage de points ainsi qu'une association point/plan précises sont exigées. Comme nous l'avons détaillé dans la section 3.2.1, un simple lancé de rayon liant le centre optique de la caméra et chaque point 3D est utilisé par Tamaazousti et al. (2011) et Lothe et al. (2009) pour segmenter le nuage de points. Si le rayon intersecte un plan des modèles 3D des bâtiments alors le point est considéré comme point du modèle et il est associé au plan le plus proche. Cependant, cette solution s'avère ne pas être assez robuste dans le contexte urbain comme le montre la figure 3.2. Ces erreurs proviennent soit de l'étape de segmentation soit de l'étape de l'association point/plan. Par exemple, en utilisant cette méthode de segmentation, des points correspondants à des voitures garées devant des bâtiments sont fréquemment considérés comme appartenant aux modèles. De plus, à cause des dérives du SLAM, le plan le plus proche du point ne correspond pas forcément au plan auquel il doit appartenir, comme le montre l'exemple de la figure 3.2. Pour améliorer, la précision de l'association point/plan, Lothe et al. (2010) propose d'exploiter la normale au point 3D. Cette information permet d'associer au point 3D le plan le plus probable au lieu du plan le plus proche. Toutefois, elle ne permet pas de limiter les erreurs liées à l'étape de segmentation. Pour améliorer la précision de la segmentation, d'autres solutions ont été proposées par exemple par Brostow et al. (2008b) et Recky et al. (2011). Ces méthodes se basent sur la segmentation des façades des bâtiments à chaque image. Cependant, ces approches ont aussi un temps d'exécution trop important pour une localisation temps réel. Pour cette raison, nous proposons dans cette section de guider la segmentation en exploitant les modèles 3D des bâtiments.

L'idée principale de notre approche consiste à établir une segmentation et une association point/plan en se basant sur l'analyse de la distance signée d_i séparant chaque point 3D $\mathcal{Q}_i$ au plan fini le plus proche dans les modèles 3D des bâtiments. Nous introduisons alors les ensembles $\{\mathcal{H}_i\}_{i=1}^{M}$ qui contiennent pour un point 3D $\mathcal{Q}_i$ l'ensemble des indices des plans du modèle 3D tel que le projeté orthogonal $\mathcal{Q}'_i$ de $\mathcal{Q}_i$ sur le plan infini associé à $\mathbf{\Pi}^h$ appartient au plan fini $\mathbf{\Pi}^h$.

Ainsi, la distance d_i est définie comme suit :

$$d_i = \begin{cases} +\min\limits_{h \in \mathcal{H}_i} d\left(\mathcal{Q}_i, \mathbf{\Pi}^h\right) & \text{Si } (\mathcal{H}_i \neq \varnothing) \text{ et (} \mathcal{Q}_i \text{ est situé devant } \mathbf{\Pi}^h) \\ -\min\limits_{h \in \mathcal{H}_i} d\left(\mathcal{Q}_i, \mathbf{\Pi}^h\right) & \text{Si } (\mathcal{H}_i \neq \varnothing) \text{ et (} \mathcal{Q}_i \text{ est situé derrière } \mathbf{\Pi}^h) \\ +\infty & \text{Sinon,} \end{cases} \tag{4.7}$$

En se basant sur cette distance, il est possible de segmenter et établir l'association point/plan en une seule étape. En effet, le point $\mathcal{Q}_i$ ne sera considéré comme point du modèle que si la valeur absolue de la distance mesurée d_i est inférieure à un certain seuil à déterminer. Nous expliquerons dans la suite notre approche pour déterminer le seuil en question.

Seuil adapté aux dérives du SLAM et aux incertitudes des modèles 3D des bâtiments Comme nous l'avons précisé ci-dessus, notre méthode de segmentation est basée sur un critère de proximité. Ce critère n'est pertinent que si la localisation initiale est assez précise et si les modèles 3D ne sont pas erronés. Toutefois, ceci n'est pas garanti à cause des dérives du SLAM d'une part et les imprécisions des modèles utilisés d'autre part. Pour faire face à ce problème, le seuil utilisé doit donc compenser ces deux types d'imprécisions. Nous proposons alors de déterminer le seuil automatiquement à chaque image clé à partir de la distribution des distances $\{d_i\}_{i=1..M}$.

La procédure d'estimation du seuil précédemment présentée nécessite qu'une segmentation de la scène soit établie. Or, à cet instant, cette segmentation n'est pas encore disponible pour l'image courante étant donnée que l'étape en cours a pour but d'établir cette segmentation. Néanmoins, étant donnée que le processus de SLAM est sujet d'accumulation d'erreurs généralement lente et progressive, il est raisonnable de considérer que la distribution des distances des points 3D aux plans calculée pour les n images clés précédentes offre une approximation acceptable de la distribution correspondante pour l'image courante. De plus, cette approximation offre l'avantage de pouvoir calculer la distribution sur un plus grand nombre de points 3D. Pour limiter l'influence des points 3D associés à une façade par erreur, nous proposons de pondérer chaque point $\mathcal{Q}_i$ par le nombre b de fois que celui-ci a été observé comme *inlier* dans les n dernières images clés. En d'autres termes, la distance d_i est considérée b fois dans la distribution. Ainsi, un point 3D ayant plusieurs observations mais dont la plupart sont *outliers* aura moins de poids qu'un point 3D ayant moins d'observations mais plus d'observations *inliers*.

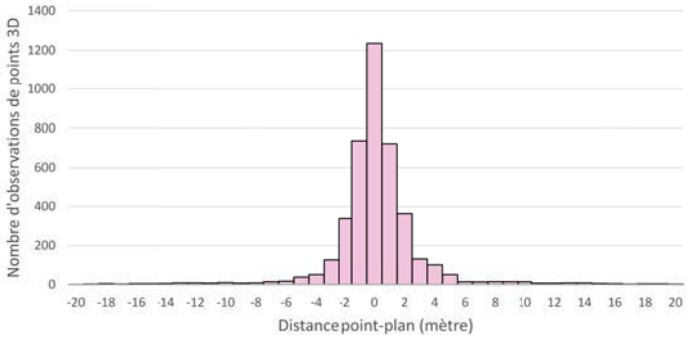
Comme le montre l'exemple de la figure 4.2, la distribution de distance tendant vers une gaussienne, nous proposons de fixer le seuil τ en fonction de sa moyenne et de son écart type. Cependant, deux problèmes se posent :

- ▷ Certaines associations point/plan erronées peuvent être présentes. Bien que peu nombreuses, ces erreurs peuvent faire dévier légèrement la distribution d'une gaussienne ;
- ▷ Le nombre de points observés peut être insuffisant pour avoir une distribution de distance pertinente ce qui empêche une estimation correcte du seuil sur ce secteur angulaire.

Pour résoudre le premier problème, nous proposons de remplacer la moyenne par la médiane et l'écart type par le MAD, ces derniers offrant une estimation plus robuste à la présence de données aberrantes. Pour le second problème, à défaut d'avoir une estimation de la distribution, nous proposons d'utiliser un seuil infini. Ainsi, le seuil τ est donné par :

$$\tau = \begin{cases} +\infty & \text{Si} \quad \text{card}\,(\mathbf{d}) \leq \chi \\ \text{mediane}(\mathbf{d}) + 5.2 * \text{MAD}(\mathbf{d}) & \text{Sinon,} \end{cases} \tag{4.8}$$

(a)



(b)

FIGURE 4.2 – **Exemple de distribution de distance.** (a) illustration de l'image traitée. (b) Histogramme correspondant à la distribution de distance.

où $\mathbf{d}$ est le vecteur concaténant les distances entre chaque point 3D $\mathcal{Q}_i$ associé aux modèles et sa façade correspondante et observé par au moins une des n dernières caméras clés. χ représente le nombre de points 3D minimal pour que la distribution de distance soit pertinente.

Seuil adapté à l'incertitude anisotrope relative au nuage de points reconstruit. Le nuage de points reconstruit par le SLAM est obtenu par triangulation à partir des poses de la caméra. Ceci rend l'incertitude relative aux positions 3D des points plus importante le long de l'axe de déplacement de la caméra comme le montrent les histogrammes de la figure 4.3 et les valeurs du tableau 4.1, d'où la notion de l'anisotropie de cette incertitude. Pour cette raison, il parait inadéquat d'appliquer le même seuil τ pour tous les points 3D reconstruits. Par conséquent, le seuil utilisé dans notre approche varie selon l'orientation de la façade associée au point 3D par rapport à la caméra. L'idée est donc de diviser l'espace en un nombre fini [1] A de secteurs angulaires $\{\Theta_a\}_{a=0}^{A}$. Chaque plan $\mathbf{\Pi}^{h_i}$ dans les modèles 3D des bâtiments est attribué par la suite à un secteur selon sa normale comme l'explique la figure 4.4. Enfin, une distribution de distance est alors calculée pour chaque secteur angulaire Θ_a.

orientation de la façade	**distance médiane** *(mètre)*
Latéral *droite*	0.94
Latéral *gauche*	0.68
Orthogonal	3.53

TABLE 4.1 – **Distance médiane séparant les points 3D et leurs façades dans chaque secteur angulaire observé par la caméra courante.**

La distribution de distance tendant vers une gaussienne dans chaque secteur angulaire, nous proposons d'estimer le seuil τ_a correspondant au secteur angulaire Θ_a de la manière suivante :

$$\tau_a = \begin{cases} +\infty & \text{Si} \quad \text{card}\,(\mathbf{d}_a) \leq \chi \\ \text{mediane}(\mathbf{d}_a) + 5.2 * \text{MAD}(\mathbf{d}_a) & \text{Sinon,} \end{cases} \tag{4.9}$$

1. Expérimentalement, nous avons noté que 4 secteurs est suffisant. Ce nombre est adapté au contexte urbain de type nord Américain.

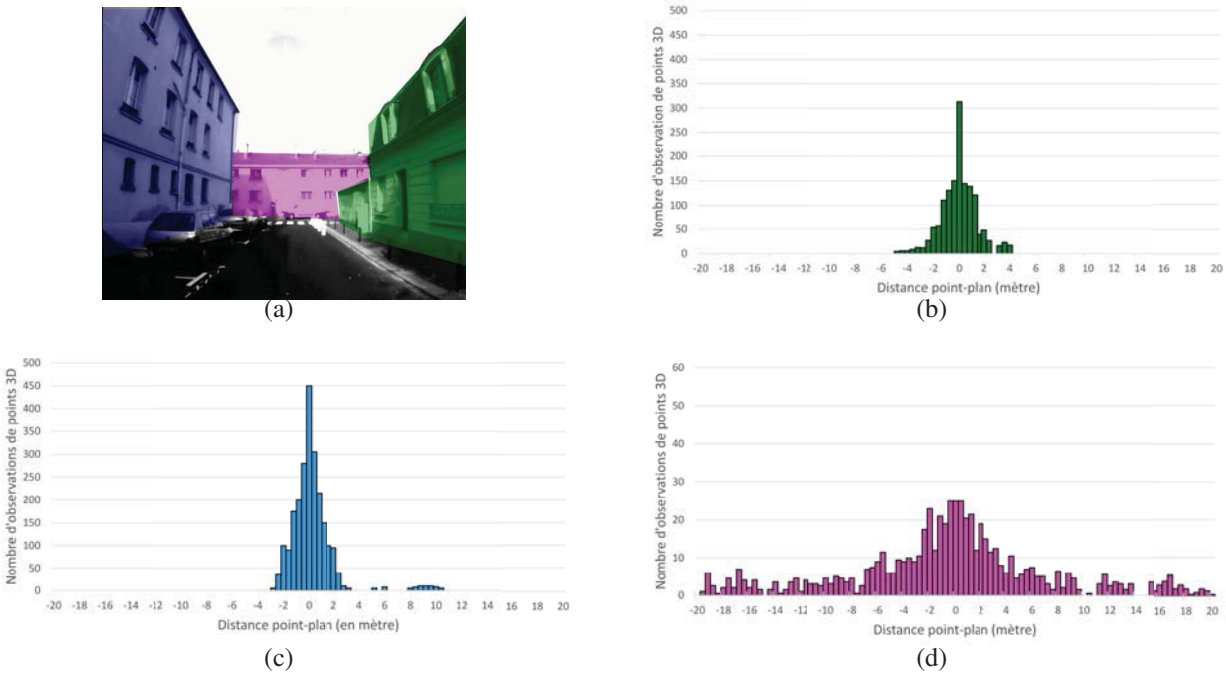


FIGURE 4.3 – **Distribution de distance pour chacune des façades observée.** (a) représente l'image courante observée par la caméra. (b) (resp. (c)) La distribution de distance associée à la façade droite (resp. gauche) observée par la caméra. (d) La distribution de distance associée à la façade orthogonale observée par la caméra.

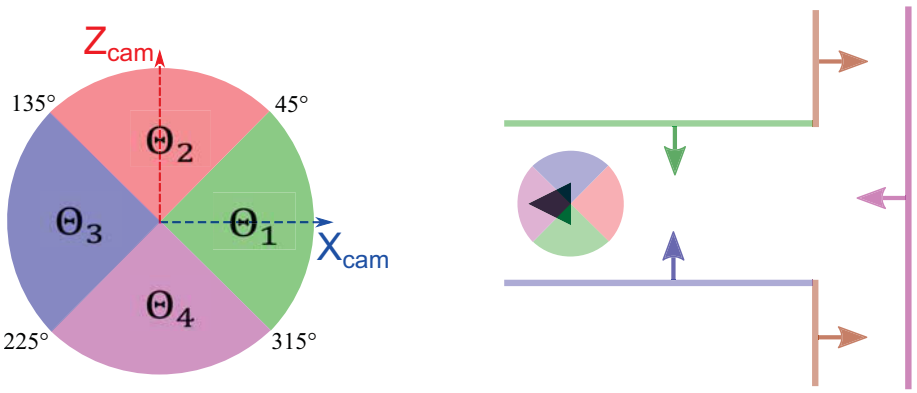


FIGURE 4.4 – **Classification des plans des modèles 3D des bâtiments selon leurs secteurs angulaires**. Exemple avec quatre secteurs angulaires.

où $\mathbf{d}_a$ est le vecteur concaténant les distances entre chaque point 3D $\mathcal{Q}_i$, observé par au moins une des n dernières caméras clés, associé aux modèles des bâtiments et dont le plan correspondant $\mathbf{\Pi}^{h_i}$ appartient au secteur angulaire Θ_a. χ représente le nombre de points 3D minimal pour que la distribution de distance soit pertinente pour un secteur angulaire donné. L'algorithme 5 résume l'algorithme de segmentation réalisé à chaque image clé.

foreach $\{\mathcal{Q}^i\}_{i=1}^{M}$ **do**
  Chercher dans les modèles 3D des bâtiments le plan fini $\mathbf{\Pi}^{h_i}$ le plus proche de $\mathcal{Q}_i$ en termes de la distance d_i (équation 4.7) ;
  if $(d_i \neq +\infty)$ **then**
    Déterminer le secteur angulaire Θ_a auquel appartient le plan $\mathbf{\Pi}^{h_i}$;
    if $(card\,(\mathbf{d}_a) \leq \chi)$ **then**
      Classifier $\mathcal{Q}_i$ comme appartenant aux modèles ($\mathcal{Q}_i \in \mathcal{M}$) ;
      Associer $\mathcal{Q}_i$ à $\mathbf{\Pi}^{h_i}$;
      Ajouter d_i à $\mathbf{d}_a$;
    else
      if $(d_i \leq mediane(\mathbf{d}_a) + 5.2 * mad(\mathbf{d}_a))$ **then**
        Classifier $\mathcal{Q}_i$ comme appartenant aux modèles ($\mathcal{Q}_i \in \mathcal{M}$) ;
        Associer $\mathcal{Q}_i$ à $\mathbf{\Pi}^{h_i}$;
        Ajouter d_i à $\mathbf{d}_a$;
      else
        classifier $\mathcal{Q}_i$ comme appartenant à l'environnement ($\mathcal{Q}_i \in \mathcal{E}$) ;
      end
    end
  else
    classifier $\mathcal{Q}_i$ comme appartenant à l'environnement ($\mathcal{Q}_i \in \mathcal{E}$) ;
  end
end

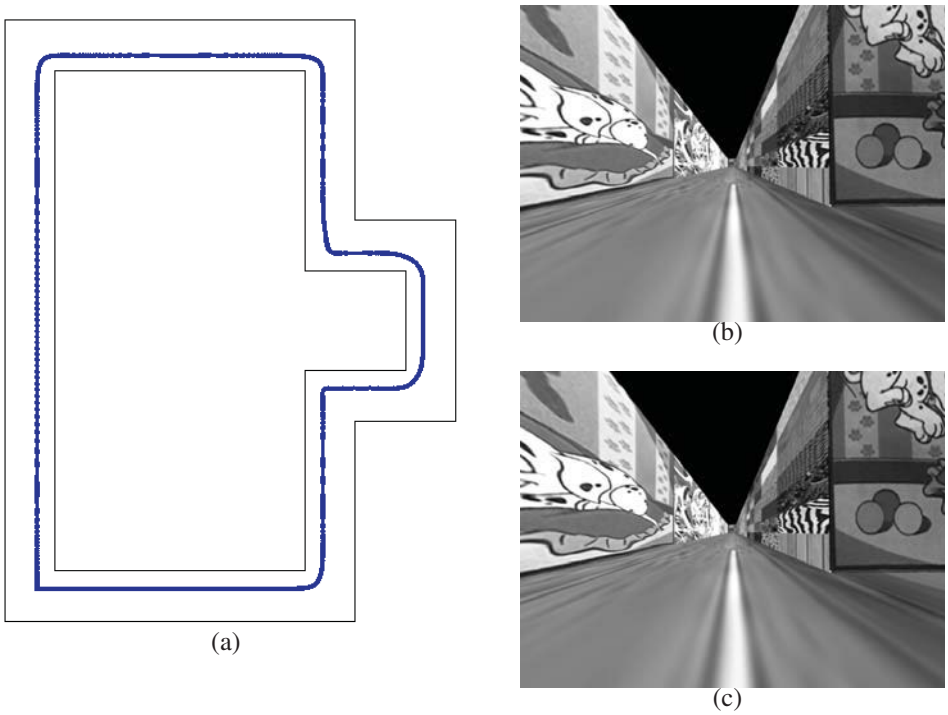
Algorithme 5 : L'algorithme de segmentation du nuage de points proposé.

4.4 Évaluation expérimentale

La présente section est consacrée à l'évaluation de notre méthode. Nous commençons, tout d'abord, par évaluer l'apport de *la contrainte dure en altitude* (section 4.4.1). L'amélioration apportée par la nouvelle méthode de segmentation du nuage de points décrite dans la section 4.3 est, par la suite, mise en évidence dans la section 4.4.2. Nous évaluons également la robustesse de notre approche face aux incertitudes des modèles 3D des bâtiments dans la section 4.4.3. Enfin, dans la section 4.4.4, la précision de notre solution est évaluée sur deux séquences réelles.

4.4.1 Évaluation de la contrainte dure en altitude

L'objectif de cette section consiste à évaluer l'apport de *la contrainte dure en altitude*. Après avoir présenté le protocole expérimental adopté, nous analyserons les résultats obtenus.



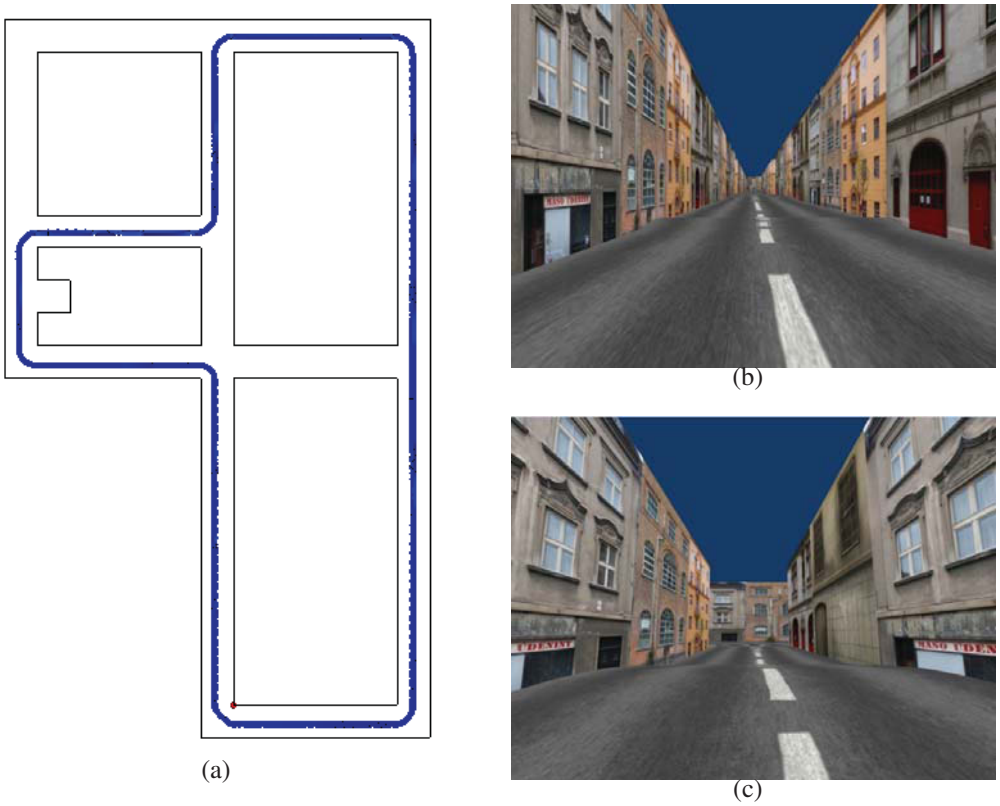
(a) (b) (c)

FIGURE 4.5 – **Illustrations de la première séquence de synthèse**. (a) En noir vue de dessus du modèle 3D des bâtiments, en bleu la trajectoire réelle de la caméra. (b), (c), Illustrations de la séquence utilisée.

Séquences utilisées et protocole expérimental. Afin d'évaluer l'apport de *la contrainte dure en altitude* proposée dans la section 4.1, nous utiliserons dans la suite deux séquences de synthèse :

- ▷ Dans la première séquence de synthèse, illustrée dans la figure 4.5, nous nous placerons dans des conditions idéales. En effet, cette séquence simule un flux vidéo enregistré par une caméra embarquée sur un véhicule à une altitude constante de 1.5m par rapport à une route supposée plate. L'utilisation d'un plan de sol parfaitement plat implique que l'altitude de la caméra sera contrainte à la même altitude dans le repère monde tout au long de la séquence. Ceci permet d'éviter tout problème lié à une association caméra/route erronée. La caméra passe à travers des couloirs dont les murs représentent les modèles géométriques de la ville représentés dans la figure 4.5(a). La vérité terrain décrivant les différentes poses de la caméra est présentée sur la même figure 4.5(a) ;
- ▷ Dans la deuxième séquence de synthèse, illustrée dans les figures 4.6(b) et 4.6(c), nous nous placerons dans un cas plus réaliste. En effet, les plans des routes sont bruités en insérant des dos d'ânes de hauteur croissante de 0.1m à 1m. Dans nos expérimentations deux MET sont utilisés. Le premier modélise les imperfections de la route, tandis que le deuxième est imprécis en modélisant la route par des plans parfaitement plats. Les modèles 3D des bâtiments ainsi que la vérité terrain sont présentés dans la figure 4.6(a)

L'évaluation se basera sur l'analyse de l'évolution des erreurs de la localisation *dans le plan* et en altitude ainsi que celle associée au facteur d'échelle. Les erreurs de la localisation *dans le plan* sont estimées en mesurant la distance entre la position *dans le plan* estimée par le SLAM et la position *dans le plan* correspondante dans la vérité terrain. Ainsi pour la $j^{ème}$ pose de la



(a) (b) (c)

FIGURE 4.6 – **Illustrations de la deuxième séquence de synthèse**. (a) En noir vue de dessus du modèle 3D des bâtiments, en bleu la trajectoire réelle de la caméra. (b), (c), Illustrations de la séquence utilisée.

caméra, l'erreur de la localisation *dans le plan* mesurée est donnée par :

$$e_{1_j} = \| \begin{pmatrix} (\mathbf{t}_j)_x \\ (\mathbf{t}_j)_y \end{pmatrix} - \begin{pmatrix} (\hat{\mathbf{t}}_j)_x \\ (\hat{\mathbf{t}}_j)_y \end{pmatrix} \|. \tag{4.10}$$

L'erreur en altitude, quant à elle, est mesurée comme suit :

$$e_{2_j} = |(\mathbf{t}_j)_z - (\hat{\mathbf{t}}_j)_z|, \tag{4.11}$$

avec $\hat{\mathbf{t}}_j = ((\hat{\mathbf{t}}_j)_x, (\hat{\mathbf{t}}_j)_y, (\hat{\mathbf{t}}_j)_z)^T$ est la position correspondante de la $j^{ème}$ image clé dans la vérité terrain.

En ce qui concerne l'erreur en facteur d'échelle, celle-ci est calculée à partir du rapport entre la distance parcourue entre deux positions successives de la caméra et la distance correspondante dans la vérité terrain. Par conséquent, l'erreur du facteur d'échelle mesurée, $r_{j-1,j}$, est donnée par :

$$r_{j-1,j} = \frac{\|\mathbf{t}_{j-1} - \mathbf{t}_j\|}{\|\hat{\mathbf{t}}_{j-1} - \hat{\mathbf{t}}_j\|}. \tag{4.12}$$

Afin d'établir notre évaluation, nous comparons les deux algorithmes suivants :

- ▷ SLAM contraint au MET et aux modèles des bâtiments.
- ▷ SLAM contraint uniquement aux modèles des bâtiments.

Résultats

- ▷ **Cas d'une route parfaitement plate (figure 4.7)**. Les différentes courbes d'évolution d'erreurs représentées dans la figure 4.7 montrent que la prise en compte de *la contrainte dure en altitude* permet d'améliorer la précision de la localisation de notre algorithme. En effet, tandis que le SLAM contraint aux modèles 3D des bâtiments dérive légèrement en altitude (*e.g.* une erreur moyenne en altitude de 0.48m), l'intégration de cette information supplémentaire dans l'ajustement de faisceaux contraint, sous forme d'*une contrainte dure*, a permis son respect strict (*i.e.* une erreur nulle au niveau de l'altitude comme le montre la figure 4.7(a)). Cette amélioration au niveau de l'estimation de la position *hors plan* de la caméra implique un meilleur résultat de segmentation du nuage de points. Dans le cas de cette séquence simpliste, ceci se traduit par une légère amélioration de l'estimation du facteur d'échelle (voir figure 4.7(c) dont l'erreur maximale passe de 1.09 à 1.05 et donc une localisation *dans le plan* plus précise (*e.g.* l'erreur maximale passe de 0.49m à 0.39m, voir figure 4.7(b)).
- ▷ **Cas d'une route bosselée (figure 4.8).** Afin de mieux évaluer la robustesse de notre algorithme face aux incertitudes du MET, nous traçons, à présent, les évolutions des erreurs en altitude, en localisation *dans le plan* et en facteur d'échelle obtenues par notre processus en exploitant, dans un premier temps, un MET précis modélisant les dos d'ânes et, dans un deuxième temps, un MET imprécis où ces imperfections ne sont pas modélisées. Les différentes courbes d'erreurs de la figure 4.8 mettent en évidence une légère dégradation de la précision pour notre algorithme exploitant un MET imprécis (courbes bleues) par rapport à celle obtenue en utilisant un MET précis (courbes vertes). En effet, comme nous l'avons mentionné, introduire *une contrainte dure* dans l'ajustement de faisceaux implique son respect strict. Ainsi, si cette dernière est erronée, la localisation obtenue est perturbée au niveau de l'altitude (voir la courbe bleu de la figure 4.8(a)) mais pas uniquement. Une dégradation locale au niveau du facteur d'échelle est également notable, comme le montre la figure 4.8(c). En effet, une caméra positionnée

à la mauvaise altitude peut avoir une influence négative sur l'étape de segmentation en augmentant le nombre de mauvaises associations point/plan, d'où une estimation moins précise du facteur d'échelle et par la suite une dégradation au niveau de la localisation *dans le plan*. Cependant, malgré ces imprécisions, nous observons que la prise en compte de la contrainte en altitude même erronée permet d'améliorer les résultats du SLAM contraint aux modèles 3D bâtiments (voir les courbes rouges de la figure 4.8). En effet, sans cette contrainte additionnelle cet algorithme dérive progressivement en altitude pour atteindre une erreur de 3m. Une telle dérive impacte notablement l'étape de segmentation qui associe moins de points aux modèles des bâtiments. La diminution de l'ensemble de points associé aux modèles entraine une mauvaise estimation de la localisation *dans le plan* et du facteur d'échelle qui se trouvent mal contraints (voir figures 4.8(b) et 4.8(c)).

Les expérimentations présentées ci-dessus soulignent les apports de *la contrainte dure en altitude* introduite dans l'ajustement de faisceaux contraint. En effet, cette information supplémentaire empêche les erreurs en altitude. Une meilleure estimation de la position *hors plan* de la caméra permet d'améliorer la segmentation du nuage de point. Ceci impacte positivement l'estimation du facteur d'échelle et donc la localisation dans le plan. Toutefois, nous avons également remarqué que notre approche est sensible face aux imprécisions du MET utilisé.

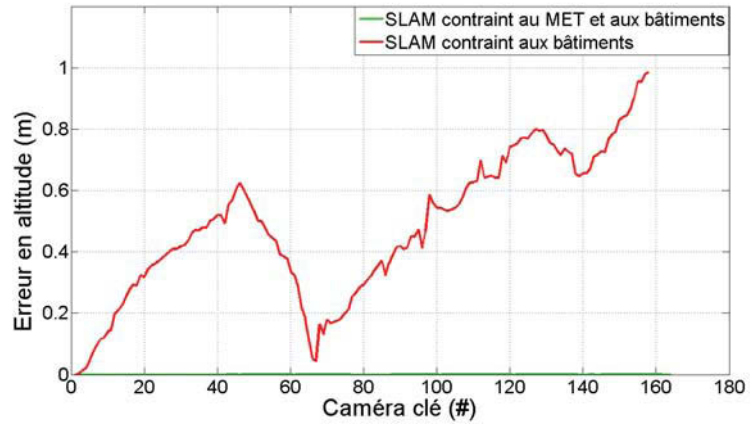
4.4.2 Évaluation de la méthode de segmentation

Après avoir analysé l'apport de *la contrainte dure en altitude* dans notre algorithme, nous nous intéresserons dans cette section à l'évaluation de la méthode de segmentation que nous avons introduite dans la section 4.3.

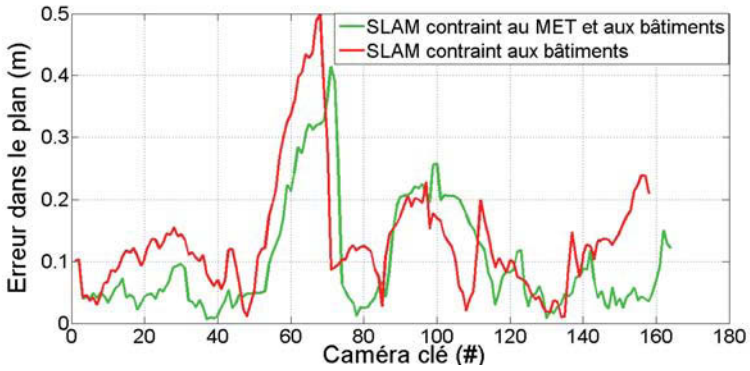
Séquence utilisée et protocole expérimental. Pour établir notre évaluation, nous utilisons la séquence de synthèse illustrée dans les figures 4.9(a) et 4.9(b). Dans cette séquence, la caméra garde tout au long de son parcours une altitude constante de 1.5m par rapport à une route parfaitement plate. Plusieurs objets occultant partiellement les bâtiments sont insérés. Ces objets, représentant la partie inconnue de l'environnent dans cette séquence, correspondent généralement dans le cas réel à des arbres ou des voitures garées.

Afin d'éviter toute mauvaise association point/plan due à une dérive au niveau de l'altitude de la caméra, le SLAM basé sur l'ajustement de faisceaux contraint aux modèles des bâtiments et avec *la contrainte dure en altitude* introduit dans la section 4.2 est utilisé. On compare alors le SLAM contraint en question avec notre méthode de segmentation et celle utilisée par Tamaazousti et al. (2011) et Lothe et al. (2009). Pour établir cette comparaison, nous visualiserons la localisation obtenue par les deux approches. Par ailleurs, nous analyserons également, pour chacune des deux approches, l'évolution du seuil du rejet du M-estimateur qui est utilisé principalement pour assurer une certaine robustesse face aux associations point/plan erronées.

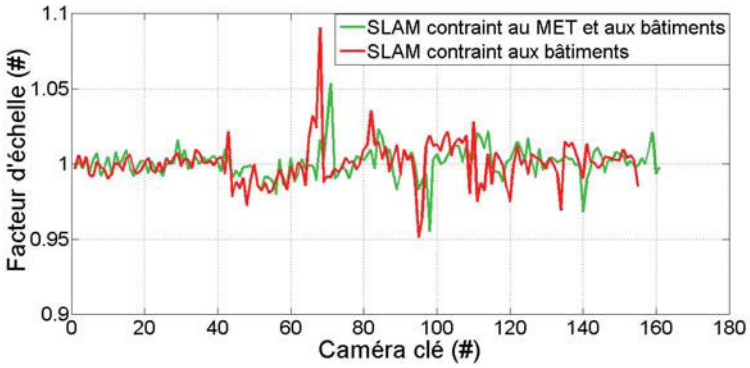
Résultats. La figure 4.9(d) montre qu'une segmentation basée uniquement sur un lancé de rayon introduit un grand nombre d'associations point/plan erronées (les points rouges localisés au milieu de la route). Plus le nombre de mauvaises associations est important, plus le seuil du rejet du M-estimateur est élevé. Ceci est mis en évidence dans la figure 4.10. La valeur de ce paramètre passe de 2 à 7 pixels pour cette méthode de segmentation. Or un seuil de rejet élevé donne une influence trop importante à certaines données aberrantes au cours de l'optimisation. Ceci cause des problèmes de convergence et donc l'échec de l'algorithme. A



(a)

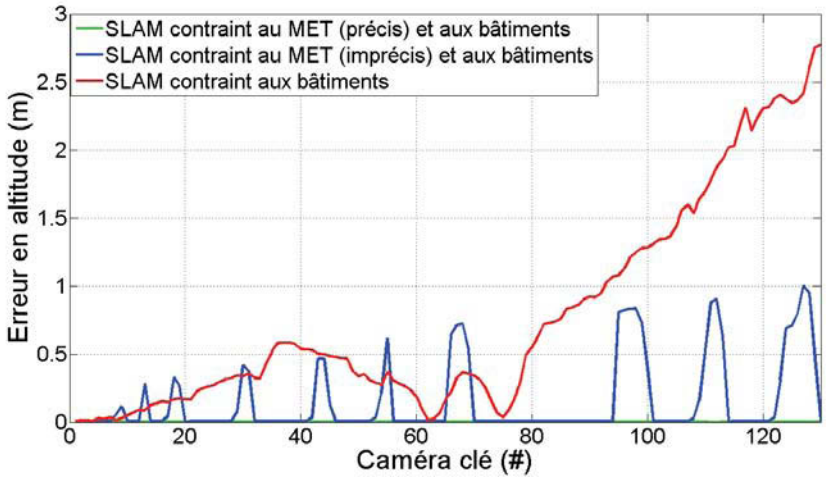


(b)

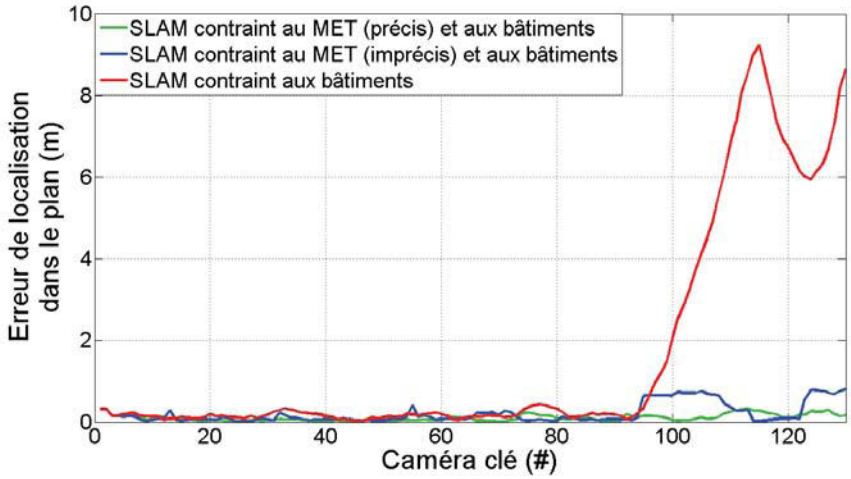


(c)

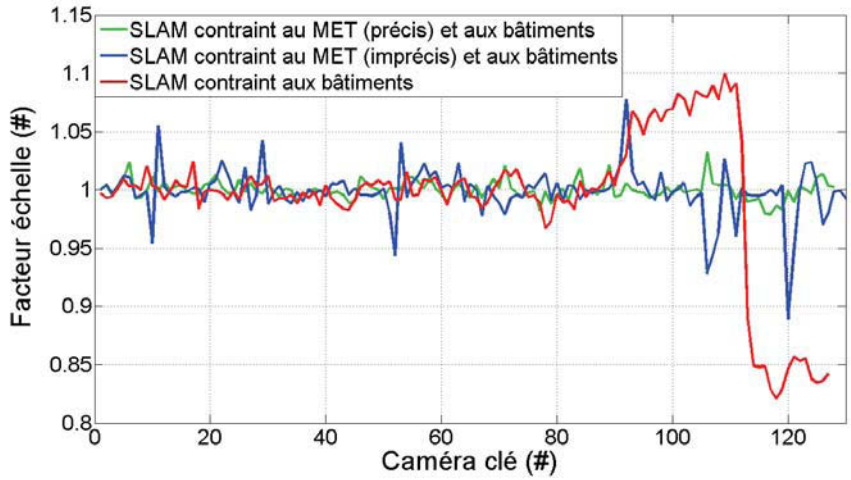
FIGURE 4.7 – **Apport de *la contrainte dure en altitude* : Cas d'une route plate.** En vert, résultat obtenu par un SLAM contraint au MET et aux modèles 3D des bâtiments. En rouge, résultat obtenu pour un SLAM contraint uniquement par les modèles 3D des bâtiments. (a) Évolution de l'erreur en altitude. (b) Évolution de l'erreur de la localisation *dans le plan*. (c) Évolution du facteur d'échelle.



(a)



(b)



(c)

FIGURE 4.8 – **Apport de *la contrainte dure en altitude* : Cas d'une route bosselée.** En vert, SLAM contraint au MET précis (bosses modélisées) et aux modèles 3D des bâtiments. En bleu, SLAM contraint au MET imprécis (bosses non modélisées) et aux modèles 3D des bâtiments. En rouge, résultat obtenu pour un SLAM contraint aux modèles 3D des bâtiments. (a) Évolution du facteur d'échelle. (b) Évolution de l'erreur en altitude. (c) Évolution de la localisation *dans le plan*.

l'inverse, notre approche de segmentation assure un nombre limité de mauvaises associations point/plan comme le montre la figure 4.9(e). Cette amélioration au niveau de la segmentation se traduit dans la figure 4.10 par une estimation plus précise du seuil du rejet du M-estimateur qui reste à peu près constant pour toute la séquence (une moyenne de 1.8 pixels). Ceci garantit une meilleure convergence et donc une localisation plus précise.

Cette expérience montre qu'une segmentation précise du nuage de point permet d'améliorer notablement la convergence d'un ajustement de faisceaux contraint aux modèles des bâtiments. Dans la suite, nous allons évaluer la robustesse de notre processus à savoir un SLAM contraint au MET et aux modèles 3D des bâtiments basé sur la nouvelle méthode de segmentation face aux incertitudes des modèles 3D des bâtiments utilisés.

4.4.3 Évaluation de la robustesse face aux incertitudes des modèles 3D des bâtiments

Dans cette section, l'objectif est d'évaluer la robustesse de notre algorithme face aux incertitudes des modèles 3D des bâtiments.

Séquence utilisée et protocole expérimental. Pour établir notre évaluation, nous avons eu recours à la séquence de synthèse illustrée dans les figures 4.11(a) et 4.11(b). Dans cette séquence la caméra est fixée à une altitude de 1.5m par rapport à la route plate. Toutefois, certains plans des modèles 3D des bâtiments sont bruités (voir modèle précis dans la figure 4.11(c) et le modèle imprécis dans la figure 4.11(d)). L'évaluation est réalisée à travers l'analyse de l'évolution de l'erreur en altitude, de la localisation *dans le plan* et celle du facteur d'échelle. Ces courbes permettront de comparer les comportements des quatre algorithmes suivants :

- ▷ SLAM contraint au MET et aux modèles 3D des bâtiments avec la nouvelle méthode de segmentation en exploitant des modèles précis des bâtiments (le modèle représenté sur la figure 4.11(c)) ;
- ▷ SLAM contraint au MET et aux modèles 3D des bâtiments avec la nouvelle méthode de segmentation en exploitant des modèles imprécis des bâtiments (le modèle représenté sur la figure 4.11(d)) ;
- ▷ SLAM contraint au MET et aux modèles 3D des bâtiments avec la méthode de segmentation proposée par Tamaazousti et al. (2011) en exploitant des modèles imprécis des bâtiments ;
- ▷ SLAM contraint uniquement aux modèles 3D des bâtiments avec la nouvelle méthode de segmentation en exploitant des modèles imprécis des bâtiments.

Résultats Comme c'est le cas pour le MET (voir section 4.4.1), l'utilisation des modèles imprécis de bâtiments introduit une légère dégradation au niveau de la précision (courbes bleues de la figure 4.12) par rapport aux résultats obtenus avec les modèles des bâtiments précis (courbes vertes de la figure 4.12). Par exemple, en utilisant le modèle imprécis, l'erreur de la localisation *dans le plan*, obtenue par notre algorithme, atteint dans la zone (1) une erreur de 1.8m et dépasse le seuil des 4m dans la zone (3) du parcours alors qu'elle a une moyenne de 0.4m en exploitant des modèles précis de bâtiments. En ce qui concerne le facteur d'échelle, nous remarquons une augmentation notable dans la zone (3) du parcours. Ceci se traduit par une dilatation de la reconstruction SLAM due à la projection des points 3D associés à la façade droite dans cette zone qui est mal modélisée. Cette dilatation est suivie par un rétrécissement brusque de

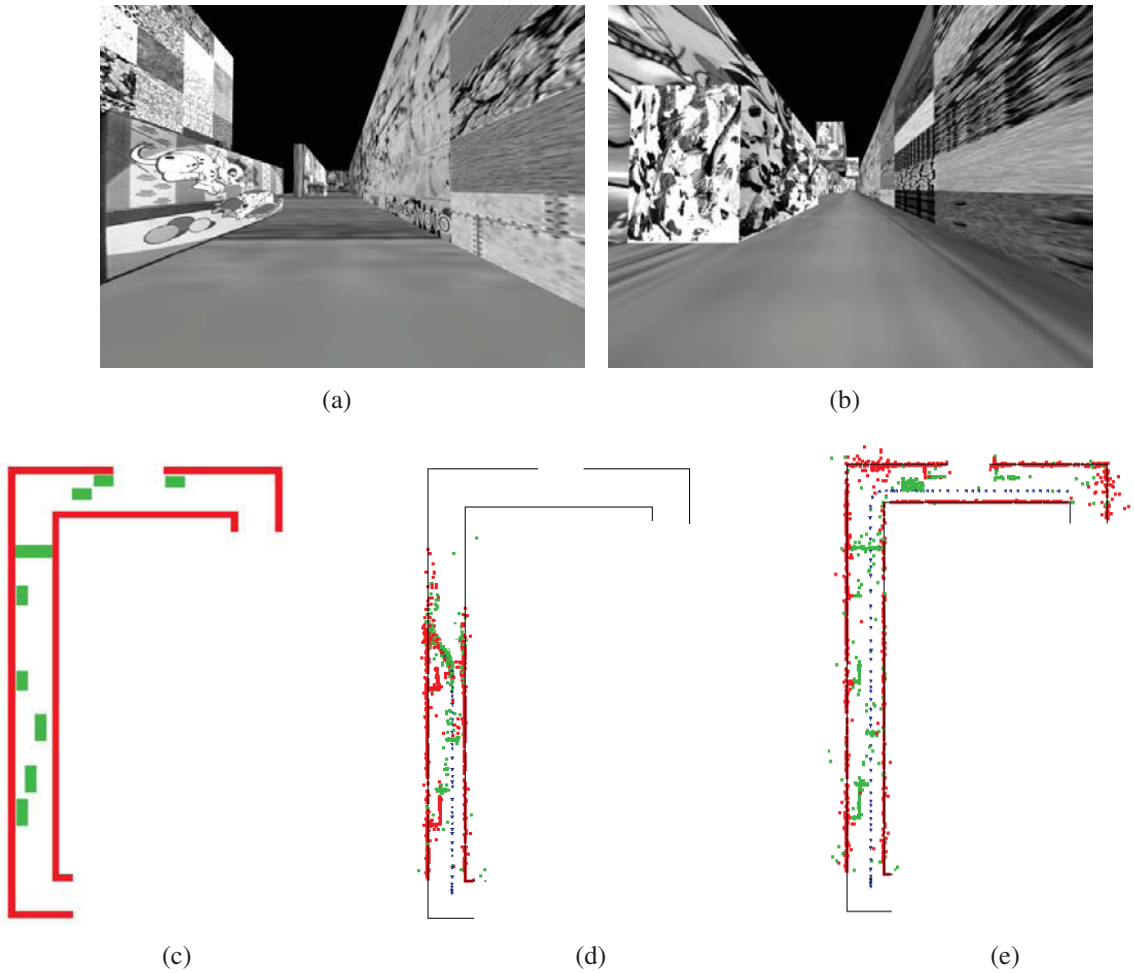


FIGURE 4.9 – **Évaluation de la méthode de segmentation proposée**. (a) (b) Des illustrations de la séquence de synthèse utilisée. Les poses de la caméra sont présentées par des triangles bleus. Les points rouges représentent les points 3D classés comme appartenant aux modèles des bâtiments tandis que les points verts représentent ceux qui sont classés comme appartenant à la partie inconnue de l'environnement. Dans (c) les modèles géométriques (rouge) et les objets insérés (vert) sont représentés. (d) La localisation obtenue par un SLAM intégrant *la contrainte dure en altitude* et celle des modèles des bâtiments et utilisant la méthode de segmentation décrite dans Tamaazousti et al. (2011). (e) La localisation obtenue par un SLAM intégrant *la contrainte dure en altitude* et celle des modèles des bâtiments et utilisant notre méthode de segmentation décrite dans la section 4.3.

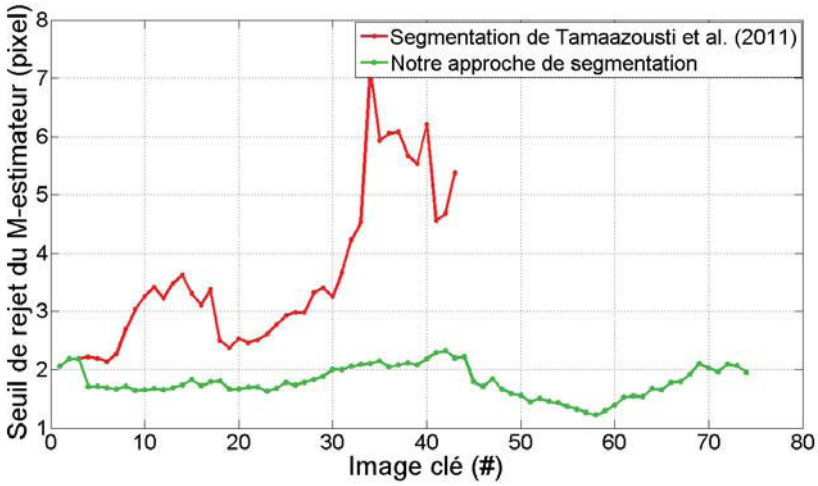


FIGURE 4.10 – **Évolution du seuil du rejet du M-estimateur.** En vert, le résultat obtenu par notre méthode de segmentation. En rouge, le résultat obtenu avec la méthode de segmentation utilisée par Tamaazousti et al. (2011)

la reconstruction SLAM dû à la présence de la façade orthogonale observée dans la zone (4) et qui permet, par la suite, la stabilisation du facteur d'échelle autour de 1. Toutes ces observations mettent en évidence la sensibilité de la méthode face aux modèles des bâtiments utilisés puisque les contraintes intégrées sont des contraintes dures. Cependant, nous remarquons que malgré la dégradation causée par les imprécisions des modèles, notre algorithme demeure plus robuste que le SLAM contraint uniquement aux modèles des bâtiments où le SLAM contraint au MET et au bâtiment mais utilisant une méthode de segmentation imprécise. En effet, en remplaçant notre méthode de segmentation par celle utilisée par Tamaazousti et al. (2011) (courbes roses de la figure 4.12), l'erreur moyenne de localisation *dans le plan* passe de 1.3m à 1.8m. D'autre part, l'absence de *la contrainte dure en altitude* (courbes rouges de la figure 4.12) entraine non seulement une dérive en altitude qui passe de 0m à 3m mais également une dérive de la localisation *dans le plan* dont l'erreur maximale passe quant à elle de 4.2m à 6.5m.

4.4.4 Évaluation de notre algorithme dans des conditions réelles

Dans les sections précédentes, nous avons démontré à travers l'utilisation des données de synthèse que l'exploitation d'un MET même imparfait permet de réduire la dérive des degrés de liberté *hors plan* du SLAM contraint aux bâtiments. Le MET permet également d'améliorer la robustesse de ce dernier lorsque les modèles des bâtiments sont imprécis. La nouvelle méthode de segmentation, quant à elle, améliore grandement la robustesse et la précision des paramètres *dans le plan* lorsque les modèles 3D des bâtiments sont imparfaits ou en présence d'occultation. A présent, nous nous placerons dans des conditions réelles afin de valider les résultats précédents.

Séquences utilisées et protocole expérimental. Dans cette section, nous utilisons deux séquences représentant deux longs parcours (1500m et 1000m, voir figure 4.13(a)), enregistrées dans un quartier de Versailles. Pour ceci, une caméra embarquée sur le toit du véhicule et fournissant 30 images par seconde avec champs de vision de 90° est utilisée. La distance entre la

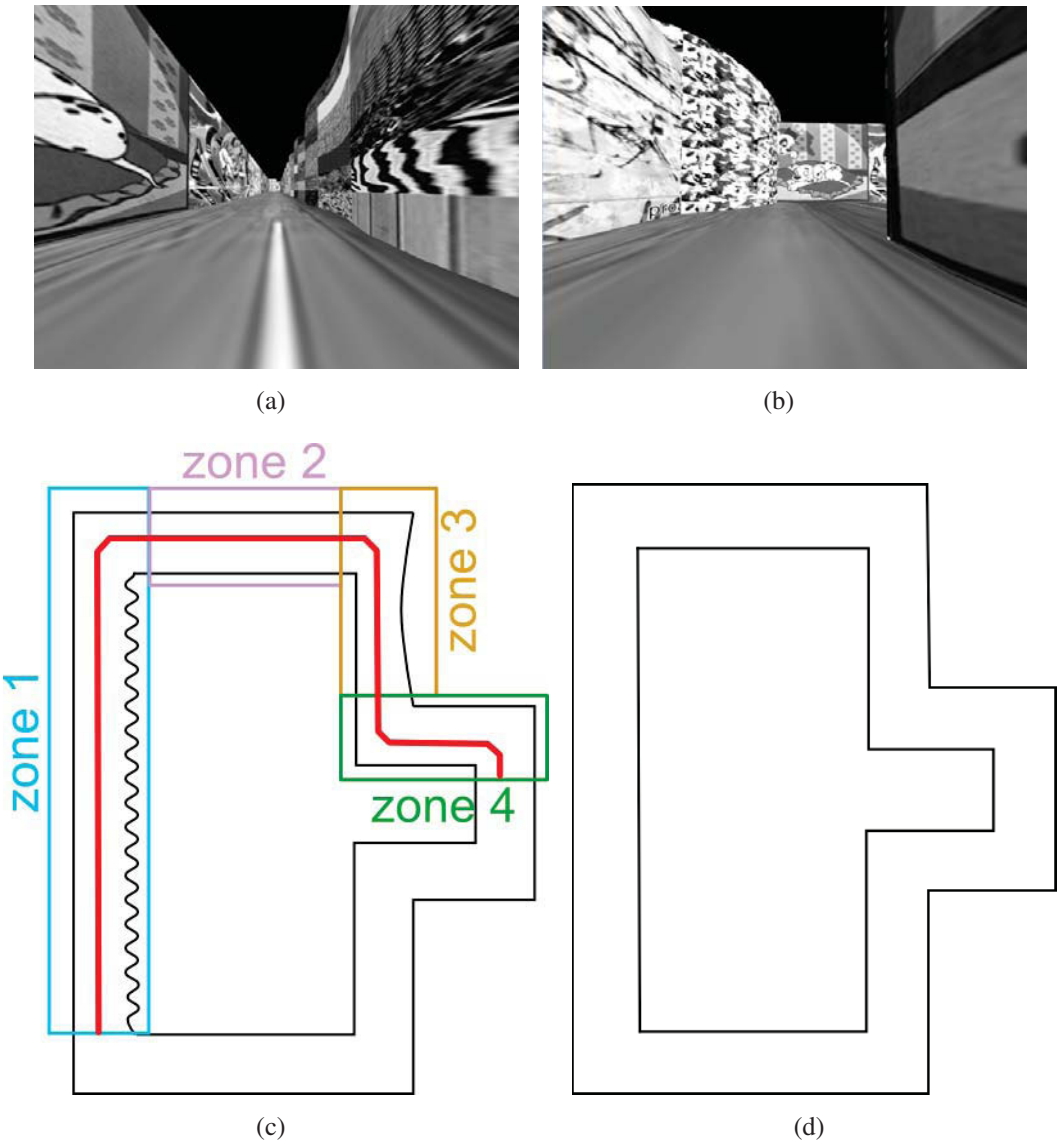
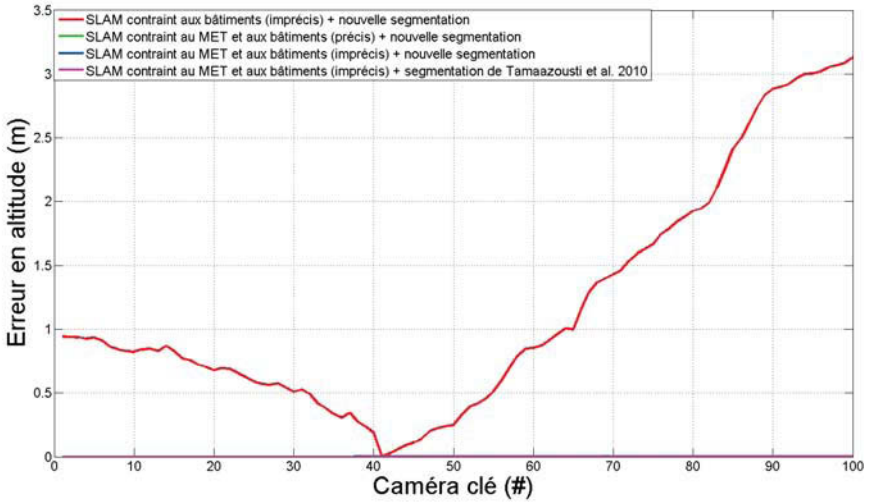
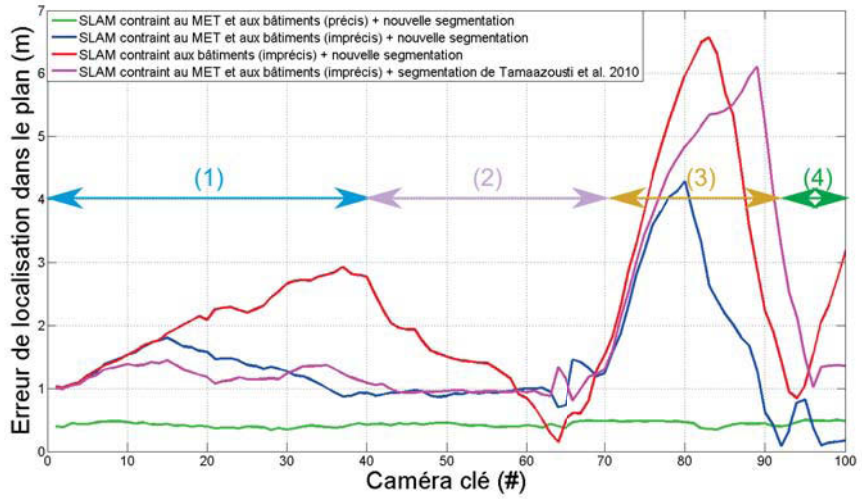


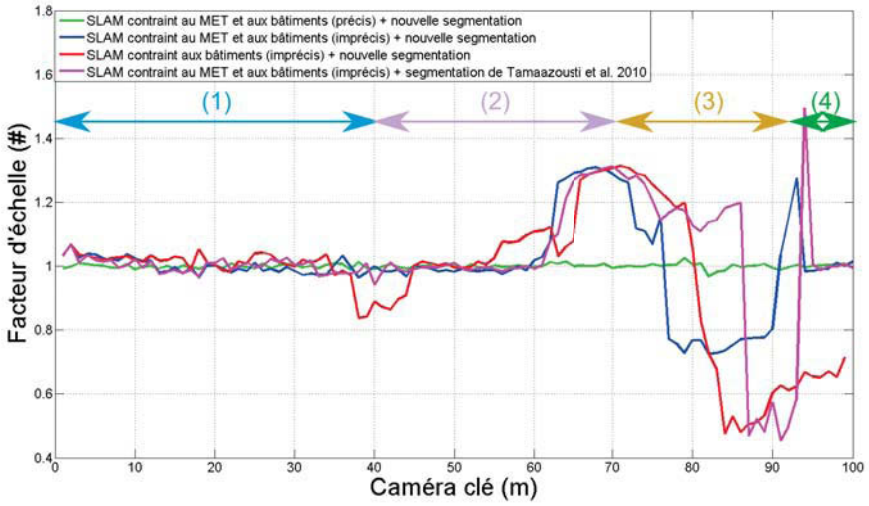
FIGURE 4.11 – **Illustration de deuxième séquence de synthèse**. (a), (b), Illustrations de la séquence utilisée. (c) En bleu vue de dessus des modèles 3D des bâtiments précis, en rouge la trajectoire réelle de la caméra. (d) Vue de dessus des modèles 3D des bâtiments imprécis.



(a)

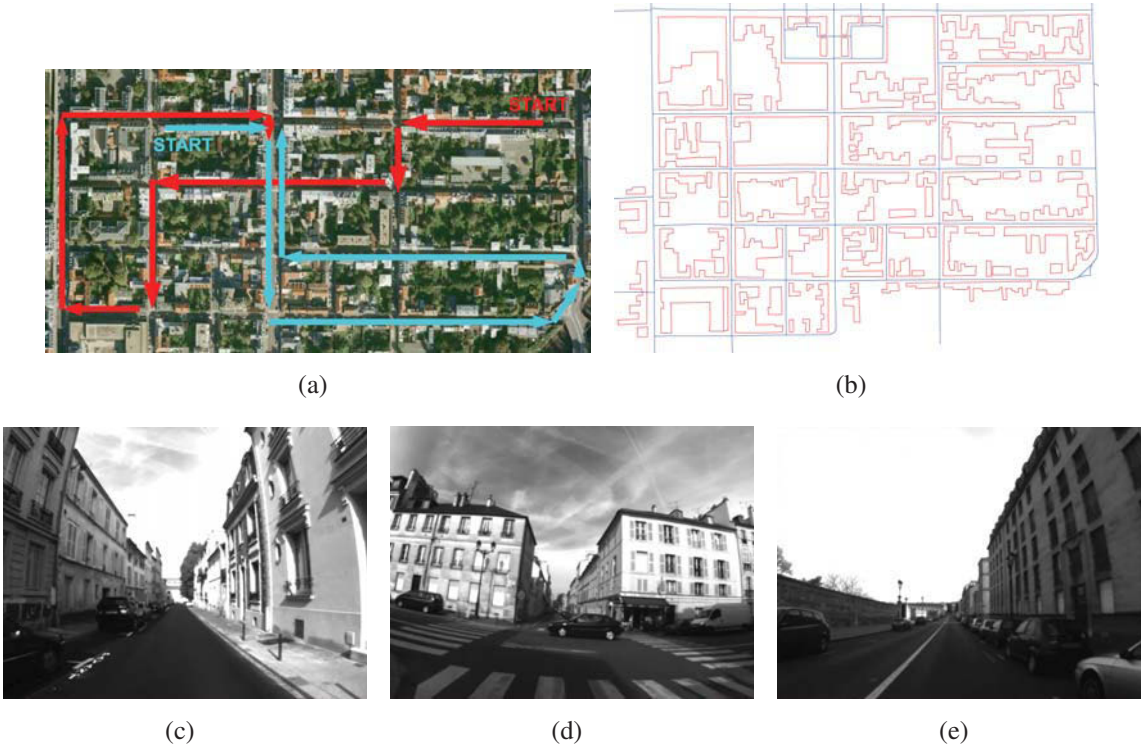


(b)



(c)

FIGURE 4.12 – **Robustesse face aux incertitudes des modèles 3D des bâtiments**. (a) Évolution de l'erreur en altitude, les trois courbes verte, bleue et rose sont confondues. (b) Évolution de l'erreur de la localisation *dans le plan*. (c) Évolution de l'erreur du facteur d'échelle.



(a) (b)

(c) (d) (e)

FIGURE 4.13 – **Séquences utilisées dans la ville de Versailles**. (a) Les deux parcours du véhicule. (b) Les modèles 3D des bâtiments (rouge), le MET (bleu). (c), (d), (e), Illustrations des séquences utilisées.

caméra et le sol est de 1.5m. Les figures 4.13(c), 4.13(d) et 4.13(e) sont des illustrations des séquences utilisées. Une vue de dessus des modèles SIG utilisés est présentée dans la figure 4.13(b). Étant donné qu'aucune vérité terrain n'est disponible pour ces deux séquences, nous allons alors comparer qualitativement les résultats obtenus avec :

- ▷ Un SLAM contraint au MET et aux modèles 3D des bâtiments (notre solution avec la nouvelle segmentation des points 3D) ;
- ▷ Un SLAM contraint uniquement aux modèles 3D des bâtiments (algorithme proposé par Tamaazousti et al. (2011)) ;
- ▷ Un SLAM *classique*(solution proposée par Mouragnon et al. (2006))

Résultats. Comme le montre la figure 4.14(a)), le SLAM *classique*, n'intégrant aucune contrainte supplémentaire, présente des dérives importantes à cause d'une mauvaise estimation du facteur d'échelle et l'accumulation des erreurs.

La figure 4.14(c) souligne l'apport de *la contrainte dure en altitude* ainsi que la méthode de segmentation proposée. En effet, comme le montre la figure 4.14(b), l'approche proposée par Tamaazousti et al. (2011) dérive principalement à cause de l'imprécision des mauvaises associations point/plan. Ces associations erronées sont dues à la fois à la méthode de segmentation peu robuste et aux dérives en altitude de la caméra. D'autre problèmes liés à l'estimation du facteur d'échelle sont notables, pour l'algorithme de Tamaazousti et al. (2011), dans certaines situations critiques où peu de bâtiments sont visibles (*e.g.* les zones A et B pour la première séquence, C et D pour la deuxième). Tous ces facteurs ont causé l'échec de cet algorithme pour les deux séquences utilisées. En exploitant à la fois les contraintes fournies par le MET et les

modèles 3D des bâtiments et en améliorant la méthode de segmentation, le SLAM contraint par notre algorithme assure une localisation plus précise et robuste pour les deux séquences pendant que l'algorithme proposé par Tamaazousti et al. (2011) échoue dans les deux cas.

4.5 Conclusion et perspectives

Dans ce chapitre nous avons proposé une nouvelle solution pour localiser une caméra dans un milieu urbain. Cette solution a l'avantage de fournir une localisation en ligne et temps réel ($30Hz$). De plus, la localisation obtenue est précise sur les six degrés de liberté de la caméra. Pour garantir de telles performances, nous avons proposé de fusionner deux contraintes complémentaires apportées par les modèles 3D des bâtiments et le MET. Alors que la contrainte des bâtiments, s'appliquant sur le nuage de points reconstruit, conditionne les degrés de liberté *dans le plan* de la caméra, la contrainte du MET, appliquée aux poses de la caméra, réduit notablement les dérives sur les degrés de liberté *hors plan*. Pour garantir une meilleure précision, les deux contraintes sont intégrées directement dans le processus de l'ajustement de faisceaux où cinq degrés de liberté par caméra sont optimisés (au lieu de six) grâce à *la contrainte dure en altitude*. Une nouvelle méthode de segmentation a été également proposée permettant de réduire le nombre d'association point/plan erronées. Toutefois, nous avons montré à travers des expériences sur des données de synthèse que cette solution reste sensible aux imprécisions des modèles utilisés. De plus, elle est sensible à l'initialisation (*i.e.* la pose de la première caméra) qui est souvent donnée par un GPS standard. En effet, une mauvaise position initiale entraine une segmentation erronée et donc l'échec de l'algorithme au bout de quelques mètres. Notons également que la précision atteinte est principalement dépendante des contraintes disponibles. En effet, quand peu de bâtiments sont observés par la caméra, les contraintes apportées par les modèles 3D des bâtiments ne permettent pas l'estimation précise du facteur d'échelle. Ceci rend l'exploitation de cette solution limitée aux milieux urbains denses. Pour faire face à ces problèmes, nous étudions, dans le chapitre suivant, une autre façon pour se localiser en ligne tout en garantissant une bonne précision sur les six degrés de liberté de la caméra. Pour ceci les informations fournies par le MET sont exploitées différemment en les fusionnant avec les contraintes liées aux données GPS.

(a) (b)

(c)

FIGURE 4.14 – **Localisation dans un milieu urbain en utilisant un SLAM contraint au MET et aux modèles 3D des bâtiments**. (a) Localisation obtenue avec un SLAM classique Mouragnon et al. (2006). (b) Localisation obtenue avec un SLAM contraint uniquement aux modèles 3D des bâtiments Tamaazousti et al. (2011). (c) Localisation obtenue avec un SLAM contraint à un modèle SIG complet.

CHAPITRE 5

SLAM contraint aux données GPS et au MET

Dans ce chapitre, nous proposons une deuxième solution permettant d'estimer en ligne les six degrés de liberté d'une caméra mobile dans un milieu urbain. Cette solution étend l'approche proposée par Lhuillier (2012) qui exploite les données d'un GPS standard en lui intégrant une contrainte supplémentaire fournie par le Modèle d'Élévation de Terrain (MET). Pour réaliser cette fusion, nous étudierons dans ce chapitre deux solutions possibles. Dans la première partie de ce chapitre (section 5.1), nous présenterons un bref synoptique de nos deux approches de fusion du SLAM avec le MET et le GPS. Ces deux approches seront, par la suite, détaillées dans les sections 5.2 et 5.3. Enfin, une évaluation complète des méthodes proposées sera présentée dans la section 5.4.

Ces travaux ont donné lieu à une publication internationale Larnaout et al. (2013a).

5.1 Introduction

Pour se localiser en ville, la solution la plus commune consiste à exploiter directement la sortie d'un système GPS. Si la précision de ce capteur est peu précise en milieux urbains denses à cause du phénomène du canyon urbain, le GPS assure une bonne localisation *dans le plan* dans les milieux péri-urbains et ruraux où les bâtiments sont moins présents qu'en ville. Par conséquent, intégrer à l'algorithme SLAM, les données fournies par le GPS peuvent, dans le cas péri-urbain et rural, améliorer la précision de la localisation en contraignant les degrés de liberté *dans le plan* de la caméra. Ceci permet également de limiter les dérives du facteur d'échelle et celles causées par l'accumulation des erreurs. Toutefois, à cause de son extrême incertitude en altitude, le GPS ne permet pas de contraindre les degrés de liberté *hors plan*. C'est pour cette raison que nous nous intéressons dans ce chapitre à l'intégration de la contrainte en altitude fournie par le MET qui a l'avantage d'être disponible à la fois en ville et hors agglomération. Nous rappelons que la contrainte en altitude peut être déduite à partir du MET en se basant sur l'hypothèse que la caméra garde une même hauteur par rapport à la route où elle se situe puisqu'elle est rigidement embarquée dans le véhicule.

Pour introduire les différentes contraintes fournies par le MET et le GPS dans le processus SLAM, nous proposons d'étendre la solution introduite par Lhuillier (2012) et détaillée

dans la section 3.3. Cette solution a l'avantage d'assurer, grâce à son ajustement de faisceaux avec contrainte d'inégalité, une certaine robustesse face aux données aberrantes du GPS qui sont notables même en milieux péri-urbain. Nous rappelons que cet ajustement de faisceaux contraint est réalisé en deux étapes. La première étape consiste à effectuer un ajustement de faisceaux classique où l'erreur de re-projection standard $f(\kappa)$ est minimisée, $\kappa = \left(\left\{\alpha_j, \beta_j, \gamma_j, \mathbf{t}_j^T\right\}_{j=1}^{N}, \{X_i, Y_i, Z_i\}_{i=1}^{M}\right)^T$ étant le vecteur des paramètres à optimiser. $(\alpha_j, \beta_j, \gamma_j)$ sont les angles Euler correspondants à la rotation $\mathcal{R}_j$ de la $j^{ème}$ caméra, $\mathbf{t}_j$ est sa translation. Enfin (X_i, Y_i, Z_i) sont les coordonnées du $i^{ème}$ point 3D. Au cours de la deuxième étape, une seconde optimisation non linéaire est effectuée dans laquelle, la distance entre les positions *dans le plan* de la caméra et les mesures GPS sont minimisées. Afin, de conserver une cohérence vis-à-vis de la géométrie multi-vues, cette seconde optimisation intègre, en plus du terme d'accroche aux données GPS, un terme de régularisation basé sur l'erreur de re-projection standard $f(\kappa)$: ce terme interdit toute dégradation de l'erreur de re-projection au-delà d'un seuil prédéfini e_t basé sur le résultat de la première optimisation (*e.g.* une dégradation de 5% de l'erreur de re-projection initiale). Ceci se résume par la fonction de coût suivante :

$$f_I(\kappa) = \frac{\omega}{e_t - f(\kappa)} + \|\mathsf{M}\kappa - \mathbf{v}\|^2, \tag{5.1}$$

avec $\mathbf{v} = (\mathbf{v}_1^T, .., \mathbf{v}_N^T)^T$ le vecteur contenant toutes les données GPS $\mathbf{v}_j = (x_j^{gps}, y_j^{gps})^T$ utilisées. La matrice M permet de récupérer les positions *dans le plan* de la caméra. Elle est définie par $\mathsf{M} = (\mathsf{D}_{2N\times 6N}|\mathsf{0}_{2N\times 3M})$ tel que $\mathsf{D}_{2N\times 6N}$ est une matrice diagonale par bloc où chaque bloc D_j de taille (2×6) est donné par

$$\mathsf{D}_j = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{5.2}$$

Pour intégrer la contrainte d'altitude fournie par le MET, deux possibilités peuvent être envisagées :

1. Introduire la contrainte au niveau du terme de pénalité sous forme d'une *contrainte dure en altitude*.
2. Introduire la contrainte au niveau du terme d'accroche aux données sous forme d'une *contrainte douce en altitude*.

Dans la suite, nous proposons d'explorer ces deux différentes approches.

5.2 Fusion de la contrainte aux données GPS avec *une contrainte dure en altitude*

Dans cette section, nous proposons de fusionner la contrainte GPS proposée par Lhuillier (2012) avec *une contrainte dure en altitude*. Comme nous l'avons expliqué dans le chapitre précédent, une *contrainte dure* implique un respect strict de l'information introduite durant le processus d'optimisation. Pour cette raison, il est important de s'assurer que la contrainte en question a été bien initialisée avant l'étape de l'optimisation. Pour ceci, une étape préliminaire est exigée. Cette étape est similaire à celle introduite dans la section 4.1.1. En effet, à chaque

image clé, la caméra est associée au plan de route le plus proche dans le MET. Une fois l'association caméra/route réalisée, l'altitude est corrigée en projetant la caméra sur le plan parallèle à sa route associée et situé à une distance h de celle-ci.

Après avoir corrigé l'altitude de la caméra, l'étape d'optimisation peut être réalisée. Afin d'assurer la fusion des contraintes GPS et MET tout en garantissant un respect strict de la contrainte en altitude, nous proposons de remplacer, dans le terme de régularisation de l'équation 5.1, l'erreur de re-projection standard par l'erreur de re-projection que nous avons introduit dans la section 4.1.2 et qui intègre *la contrainte dure en altitude*. Ceci implique la nécessité de modifier la paramétrisation de la caméra en exprimant sa pose dans le repère de la route associée (voir figure 4.1). D'autre part, pour faire tendre la position *dans le plan* de la caméra vers la donnée GPS, il est nécessaire que ces paramètres soient exprimés dans le même référentiel. Nous choisissons alors de modifier le terme d'accroche aux données en exprimant chaque mesure GPS dans le repère de la route associée à la caméra clé correspondante. Ainsi, la nouvelle paramétrisation de la $j^{ème}$ donnée GPS associée à la $j^{ème}$ caméra clé est donnée par :

$$\tilde{\mathbf{v}}_j^{k_j} = \tilde{\mathsf{L}}^{k_j} \begin{pmatrix} x_j^{gps} \\ y_j^{gps} \\ z_j^{gps} \\ 1 \end{pmatrix}, \tag{5.3}$$

avec z_j^{gps} est la donnée en altitude fournie par le GPS et $\tilde{\mathsf{L}}^{k_j}$ est la matrice de passage (4×4) du repère monde au repère de la route $\mathbf{\Lambda}^{k_j}$ associée à la $j^{ème}$ caméra clé.

Cependant, une application directe de ce changement de repère à la donnée 3D fournie par le GPS impliquerait la prise en compte de l'erreur en altitude de cette donnée. Pour éviter ce problème, nous proposons de corriger également l'altitude associée à la donnée GPS à partir du MET. On introduit alors le point $\mathcal{G}_j$ représentant le projeté de la $j^{ème}$ donnée GPS sur le plan parallèle à la route et situé à la hauteur souhaitée h comme le montre la figure 5.1. Par conséquent, la nouvelle paramétrisation de la $j^{ème}$ donnée GPS associée à la $j^{ème}$ caméra clé est la suivante :

$$\tilde{\mathbf{v}}_j^{k_j} = \tilde{\mathsf{L}}^{k_j} \tilde{\mathcal{G}}_j \tag{5.4}$$

.

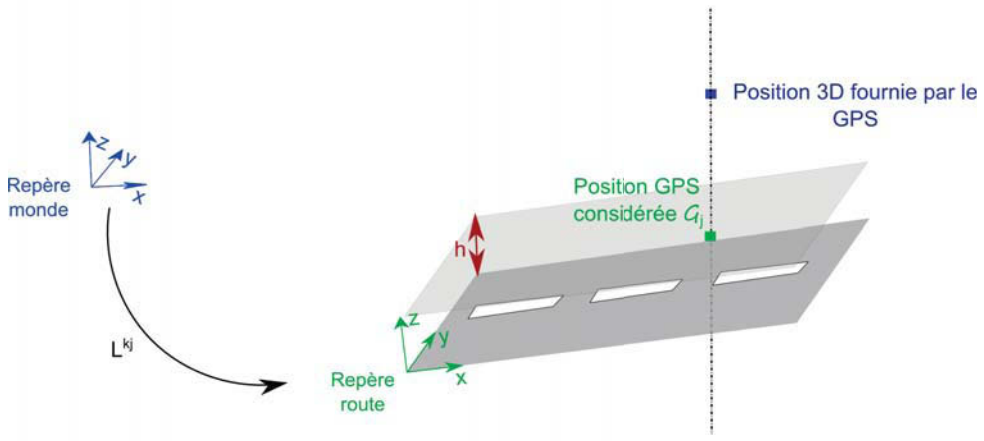


FIGURE 5.1 – **Définition du point G_j**

Pour la clarté des équations, nous noterons, dans la suite,

$$\eta = \left((\alpha_j^{k_j}, \beta_j^{k_j}, \gamma_j^{k_j}, (\mathbf{t}_j^{k_j})_x, (\mathbf{t}_j^{k_j})_y)_{j=1}^{N}, (X_i, Y_i, Z_i)_{i=1}^{M} \right)^T \tag{5.5}$$

le vecteur des paramètres à optimiser durant l'ajustement de faisceaux intégrant la contrainte GPS et *la contrainte dure en altitude*, $(\alpha_j^{k_j}, \beta_j^{k_j}, \gamma_j^{k_j}, (\mathbf{t}_j^{k_j})_x, (\mathbf{t}_j^{k_j})_y)$ étant les paramètres de la $j^{ème}$ pose de la caméra à optimiser exprimés dans le repère de la route Λ^{k_j}. $(\mathbf{t}_j^{k_j})_z$ est supposé constant, il n'est donc pas optimisé. Nous désignerons par ϑ le vecteur concaténant les positions *dans le plan* des vecteurs $\left(\mathbf{v}_j^{k_j}\right)_{j=1}^{N}$. Ainsi, la fonction de coût utilisée dans le présent ajustement de faisceaux est donnée par :

$$f_{1_I}(\eta) = \frac{\omega}{e_t - l(\eta)} + \|\mathsf{M}_1\eta - \vartheta\|^2, \tag{5.6}$$

où $l(\eta)$ est l'erreur de re-projection intégrant *la contrainte dure en altitude* et donnée par :

$$l(\eta) = \sum_{i=1}^{M} \sum_{j \in \mathcal{D}_i} \rho\left(\|\mathbf{l}_{i,j}\|, c\right), \tag{5.7}$$

avec

$$\mathbf{l}_{i,j} = \mathbf{q}_{i,j} - \pi\left(\mathsf{K}(\mathcal{R}_j^{k_j})^T \left[\mathsf{I}_{3\times3}| - \mathbf{t}_j^{k_j}\right] \tilde{\mathsf{L}}^{k_j} \tilde{\mathcal{Q}}\right). \tag{5.8}$$

$\mathsf{M}_1 = (\mathsf{D}_{1_{2N\times5N}}|\mathsf{0}_{2N\times3M})$ tel que $\mathsf{D}_{1_{2N\times5N}}$ est une matrice diagonale par bloc où chaque bloc de taille (2×5) est donné par

$$\mathsf{D}_{1_j} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.9}$$

M_1 permet de récupérer du vecteur des paramètres η le vecteur concaténant uniquement les positions *dans le plan* de la caméra.

Le seuil e_t représente une dégradation de 5% de l'erreur de re-projection initiale et le poids de la contrainte ω est estimée empiriquement, plus de détails sur la détermination de ces paramètres sont disponibles dans Lhuillier (2012). Plus de détails sur la minimisation de cette fonction sont disponibles dans l'annexe, section A.3.2.

Le schéma général du fonctionnement du processus d'optimisation incluant la contrainte GPS et la *contrainte dure en altitude* est représenté dans la figure 5.2.

5.3 Fusion de la contrainte aux données GPS avec *une contrainte douce en altitude*

Une autre solution pour introduire la contrainte en altitude dans l'ajustement de faisceaux avec la contrainte d'inégalité consiste à exploiter le terme d'accroche aux données. Dans cette section, l'information d'altitude est intégrée sous forme d'une *contrainte douce*. Ceci implique que le respect strict de la contrainte associée n'est pas exigé. Par conséquent, l'altitude de la caméra n'est pas corrigée au préalable, seule l'association camera/route est effectuée.

Une fois les associations caméra/route établie, l'étape d'optimisation peut être réalisée. Pour ceci une nouvelle fonction de coût est proposée. En effet, pour prendre en compte *la contrainte*

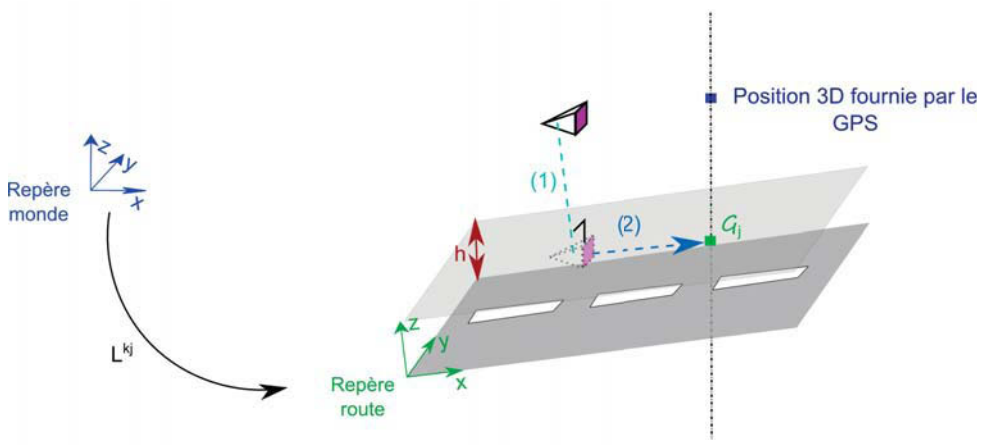


FIGURE 5.2 – **Schéma général du fonctionnement du processus d'optimisation incluant la contrainte GPS et la *contrainte dure en altitude***. La pose initiale de la caméra est représentée par le cône (opaque). (1) représente la première étape du processus d'optimisation : projection de la caméra orthogonalement sur le plan parallèle à sa route associée et situé à la hauteur souhaitée h. La position de la caméra après la projection est représentée par le cône transparent. (2) Ajustement de faisceaux avec la contrainte GPS et la *contrainte dure en altitude*.

douce en altitude dans l'ajustement de faisceaux contraint aux données GPS, nous introduisons un deuxième terme d'accroche aux données mesurant la différence entre la distance séparant la caméra et sa route correspondante et la hauteur souhaitée h. Ceci revient donc à minimiser l'écart entre l'altitude de chaque caméra clé exprimée dans le repère de la route associé $\left\{(\mathbf{t}_j^{k_j})_z\right\}_{j=1}^{N}$ et la hauteur h. Le terme d'accroche aux données global $c(\kappa)$ est donc estimé à travers la somme du terme associé aux mesures GPS et celui associé au MET :

$$c(\kappa) = \|\mathsf{M}\kappa - \mathbf{v}\|^2 + \|((\mathbf{t}_1^{k_1})_z, ..., (\mathbf{t}_N^{k_N})_z)^T - \mathbf{h}\|^2, \tag{5.10}$$

avec $\mathbf{h} = (\underbrace{h...h}_{\text{N fois}})^T$. Le vecteur $({t_1^{k_1}}_z, ..., {t_N^{k_N}}_z)^T$ peut être exprimé en fonction du vecteur des paramètres à optimiser κ de la manière suivante :

$$((\mathbf{t}_1^{k_1})_z, ..., (\mathbf{t}_N^{k_N})_z)^T = \mathsf{Z}\kappa + \mathbf{m}, \tag{5.11}$$

avec $\mathsf{Z} = (\mathsf{D}_{z_{N\times 6N}}|0_{N\times 3M})$ tel que $\mathsf{D}_{z_{N\times 6N}}$ est une matrice diagonale par bloc où chaque bloc de taille 1×6 est donné par $\mathbf{d}_{z_j} = \begin{pmatrix}0 & 0 & 0 & \mathsf{L}^{k_j}(3,1) & \mathsf{L}^{k_j}(3,2) & \mathsf{L}^{k_j}(3,3)\end{pmatrix}$ et $\mathbf{m} = (\mathsf{L}^{k_1}(3,4)...\mathsf{L}^{k_N}(3,4))^T$.

Ainsi, le terme d'accroche aux données global peut s'écrire de la façon suivante :

$$\begin{aligned} c(\kappa) &= \|\mathsf{M}\kappa - \mathbf{v}\|^2 + \|\mathsf{Z}\kappa + \mathbf{m} - \mathbf{h}\|^2 \\ &= \|(\mathsf{M}_2\kappa + \mathbf{m}_2) - \mathbf{w}\|^2, \end{aligned} \tag{5.12}$$

tel que

▷ $\mathsf{M}_2 = (\mathsf{D}_{2_{3N\times 6N}}|0_{3N\times 3M})$ tel que $\mathsf{D}_{2_{3N\times 6N}}$ est une matrice diagonale par bloc où chaque bloc de taille (3×6) est donné par $\mathsf{D}_{2_j} = \begin{pmatrix}0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & \mathsf{L}^{k_j}(3,1) & \mathsf{L}^{k_j}(3,2) & \mathsf{L}^{k_j}(3,3)\end{pmatrix}$.

$\triangleright$ $\mathbf{m}_2 = (\mathbf{n}_1^T ... \mathbf{n}_N^T)^T$ où $\mathbf{n}_j = \begin{pmatrix} 0 \\ 0 \\ \mathsf{L}^{k_j}(3,4) \end{pmatrix}$.

$\triangleright$ $\mathbf{w} = (\mathbf{u}_1^T ... \mathbf{u}_N^T)^T$ où $\mathbf{u}_j = \begin{pmatrix} x_j^{gps} \\ y_j^{gps} \\ h \end{pmatrix}$, les données GPS *dans le plan* sont exprimées dans le repère monde et h est l'altitude réelle de la caméra par rapport au plan de la route.

Finalement, la fonction de coût utilisée pour l'ajustement de faisceaux avec la contrainte GPS et *la contrainte douce en altitude* est la suivante :

$$f_{\mathcal{Z}_I}(\kappa) = \frac{\omega}{e_t - f(\kappa)} + \|(\mathsf{M}_2\kappa + \mathbf{m}_2) - \mathbf{w}\|^2, \tag{5.13}$$

Plus de détails sur la minimisation de cette fonction sont disponible dans l'annexe, section A.3.3.1. Le schéma général du fonctionnement du processus d'optimisation incluant la contrainte GPS et la *contrainte douce en altitude* est représenté dans la figure 5.3.

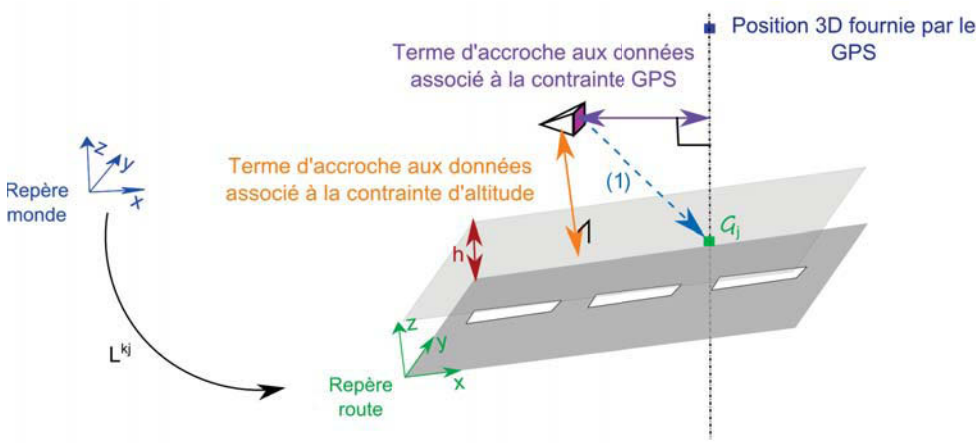


FIGURE 5.3 – **Schéma général du fonctionnement du processus d'optimisation incluant la contrainte GPS et la *contrainte douce en altitude***. La pose initiale de la caméra est représentée par le cône. (1) Ajustement de faisceaux avec la contrainte GPS et la *contrainte douce en altitude*.

5.4 Évaluation expérimentale

Dans cette section, deux différentes expérimentations sont présentées pour évaluer les solutions précédemment décrites en termes de robustesse et précision. La première expérimentation évalue la robustesse de chacune de nos deux solutions face à l'imprécision du MET (section 5.4.1). La deuxième expérimentation évalue la qualité de la localisation sur une séquence réelle (section 5.4.2).

5.4.1 Évaluation de la robustesse des solutions proposées

Dans cette section, notre objectif est d'évaluer la robustesse des solutions proposées face aux imprécisions du MET. Après avoir décrit la séquence utilisée et le protocole expérimental

adopté, nous analyserons les résultats obtenus.

Séquence utilisée et protocole expérimental. Les routes sont souvent objet d'imperfections qui ne sont pas nécessairement modélisées dans le MET utilisé. Dans le cadre de notre étude, les MET utilisés offrent une représentation simpliste du réseau routier. En effet, chaque route dans le MET est schématisée par un segment 3D représentant son axe. Ainsi, cette représentation ne modélise pas les éventuelles imperfections des routes qui peuvent apparaitre sous forme d'un bruit local (*e.g.* creux, dos d'ânes ...). En plus de sa représentation simpliste, le MET peut présenter également des imprécisions sous forme d'un biais. Même si ce biais peut atteindre 2m, ce type d'imprécisions ne présente pas une réelle limitation puisqu'il n'influence pas la cohérence de la reconstruction SLAM. Toutefois, les imperfections sous forme de bruit local et qui ne sont pas modélisés dans le MET peuvent perturber nos algorithmes entrainant des dérives importantes de localisation. Pour évaluer l'impact de telles imperfections sur la localisation, nous utilisons, dans cette expérimentation, une séquence de synthèse illustrée dans la figure 5.4. Dans cette séquence, le plan de la route est bruité à travers l'insertion de dos-d'ânes dont l'altitude varie entre 0.1m et 1m. Ce bruit n'est pas modélisé dans le MET utilisé dont les routes sont supposées parfaitement plates. Tout au long du parcours, les mesures GPS utilisées représentent les positions *dans le plan* de la caméra données par la vérité terrain.

(a) (b)

FIGURE 5.4 – **Illustrations de la séquence de synthèse utilisée**.

Pour réaliser notre évaluation nous comparons nos deux solutions de fusion du GPS et du MET avec la solution introduite par Lhuillier (2012). L'évaluation se fera en se basant sur l'analyse des courbes d'erreurs en altitude, de la localisation *dans le plan* ainsi que l'erreur angulaire. Nous rappelons que ces erreurs peuvent être mesurées, pour la $j^{ème}$ caméra, de la manière suivante :

▷ Erreur en altitude :

$$e_{2_j} = |(\mathbf{t}_j)_z - (\hat{\mathbf{t}}_j)_z|, \tag{5.14}$$

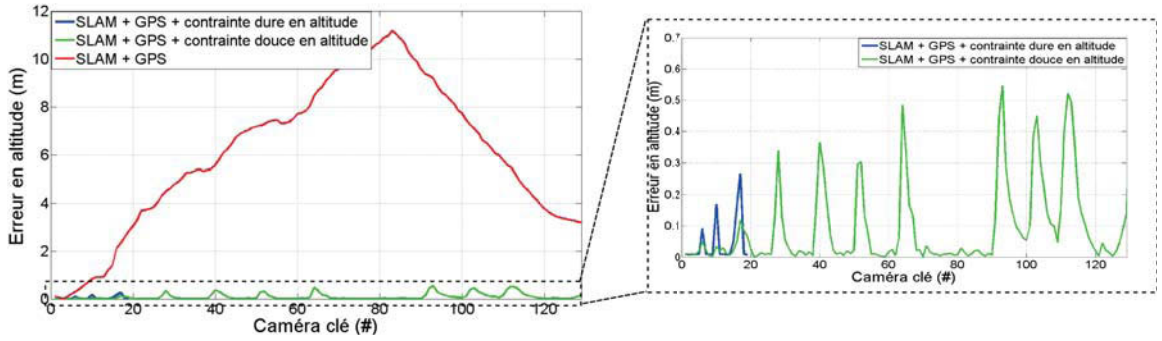
▷ Erreur de la localisation *dans le plan* :

$$e_{1_j} = \| \begin{pmatrix} (\mathbf{t}_j)_x \\ (\mathbf{t}_j)_y \end{pmatrix} - \begin{pmatrix} (\hat{\mathbf{t}}_j)_x \\ (\hat{\mathbf{t}}_j)_y \end{pmatrix} \|. \tag{5.15}$$

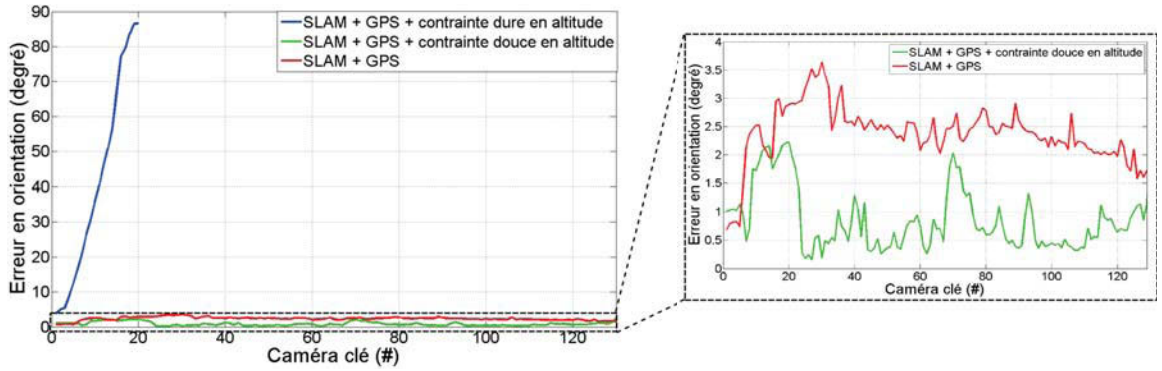
▷ Erreur en orientation :

$$\mathcal{R}_j^{err} = \mathcal{R}_j \left(\hat{\mathcal{R}}_j\right)^T, \tag{5.16}$$

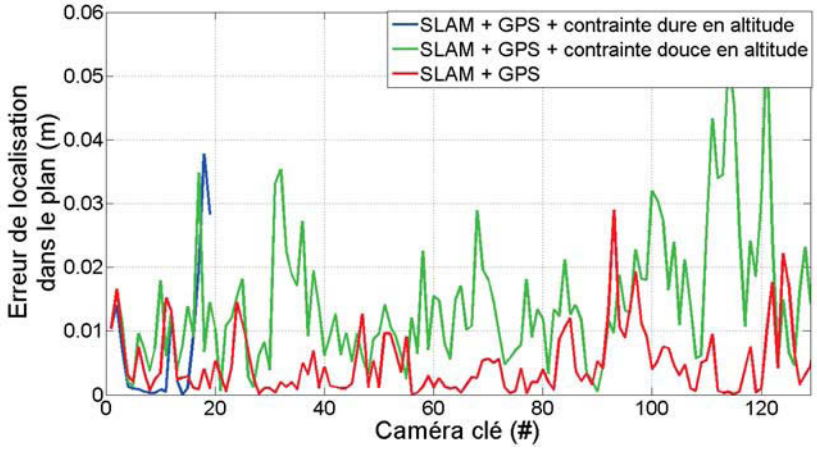
avec $\hat{\mathbf{t}}_j = ((\hat{\mathbf{t}}_j)_x, (\hat{\mathbf{t}}_j)_y, (\hat{\mathbf{t}}_j)_z)^T$ et $\hat{\mathcal{R}}_j$ sont respectivement la position et l'orientation correspondante de la $j^{ème}$ image clé dans la vérité terrain.



(a)



(b)



(c)

FIGURE 5.5 – **Évaluation de la robustesse face aux imperfections du MET**. (a) A gauche, évolution de l'erreur en altitude, à droite, zoom sur l'évolution de l'erreur en altitude pour le SLAM intégrant la contrainte GPS et *la contrainte douce ou dure en altitude*. (b) A gauche, évolution de l'erreur angulaire, à droite, zoom sur l'évolution de l'erreur angulaire pour le SLAM intégrant la contrainte GPS avec et sans *la contrainte douce en altitude*. (c) Évolution de l'erreur de la localisation dans le plan.

Résultats. Les différentes courbes d'évolution d'erreurs sont présentées dans la figure 5.5. Ces résultats sont également résumés dans le tableau 5.1.

Concernant la solution proposée par Lhuillier (2012) et intégrant uniquement la contrainte GPS, nous observons une dérive considérable au niveau de l'estimation de l'altitude de la caméra qui atteint 11.186m. Une dérive en orientation est également notable. Par exemple, l'erreur au niveau de l'angle tangage dépasse $2°$, tandis que celle associé à l'angle roulis atteint $\simeq 3°$.

Comme le montre la figure 5.5(a), nous observons que notre approche intégrant la contrainte GPS et la *contrainte dure en altitude* permet le respect strict de l'information d'altitude introduite au niveau du processus d'optimisation. Étant donné que cette information est bruitée, l'altitude de la caméra à l'issu de l'ajustement de faisceaux contraint l'est aussi. Ainsi, nous distinguons plusieurs pics (voir la courbe bleue de la figure 5.5(a)) représentant les dos d'ânes non modélisés dans le MET. Quand ces imprécisions non modélisés deviennent importantes, une dérive importante en orientation, principalement au niveau de l'angle roulis, est observée. Atteignant la valeur de $86.61°$, le SLAM basé sur un ajustement de faisceaux incluant la contrainte GPS et *la contrainte dure en altitude* échoue au bout d'une centaine de mètres plus précisément au niveau du premier virage.

Assurant une localisation pour toute la séquence, notre solution basée sur *la contrainte douce en altitude* réalise la meilleure performance en garantissant une faible erreur en altitude (erreur médiane $0.027m$). En effet, en comparant avec *la contrainte dure en altitude*, *la contrainte douce en altitude* permet d'atténuer les erreurs au niveau de la position *hors plan* de la caméra quand l'altitude introduite est erronée (voir les premiers pics des courbes bleue et verte de la figure 5.5(a)). En plus de la faible dérive en altitude, comme le montre la figure 5.5(b), cette approche permet aussi de réduire l'erreur en orientation (erreur maximale pour l'angle roulis $1.08°$, erreur maximale pour l'angle tangage $1.95°$). Notons également que cette performance sur les degrés de liberté *hors plan* n'a pas perturbé la précision de la localisation *dans le plan* (figure 5.5(c)). En effet, cette approche assure une erreur *dans le plan* proche de celle obtenue avec l'algorithme proposé par Lhuillier (2012) (*e.g.* maximum d'erreur atteint pour notre approche est 0.078m *vs* maximum d'erreur atteint pour l'approche de Lhuillier (2012) est 0.030m).

		Erreur angulaire (degré)		Erreur altitude (m)	Erreur dans le plan (m)
		Tangage	Roulis		
GPS	médiane	0.722	2.213	6.055	0.003
	max	2.260	2.969	11.186	0.030
GPS + Altitude Dure	médiane	0.191	38.116	0.009	0.002
	max	0.531	86.610	0.267	0.038
GPS + Altitude Douce	médiane	0.242	0.356	0.027	0.012
	max	1.955	1.087	1.518	0.078

TABLE 5.1 – **Tableau récapitulatif des erreurs de localisation obtenues sur la séquence de synthèse**. Notons que les performances de l'approche basée sur *la contrainte dure en altitude* sont calculées uniquement sur une petite partie de la séquence utilisée étant donné qu'elle échoue au bout d'une centaine de mètres.

Pour conclure, cette expérimentation montre que les deux solutions proposées dans ce chapitre permettent d'introduire une contrainte supplémentaire au SLAM contraint aux données GPS sans perturber sa précision de la localisation *dans le plan* de la caméra. Toutefois, contrairement aux contraintes fournies par les modèles des bâtiments exploitées dans le chapitre pré-

cédent, les mesures GPS ne permettent pas de contraindre l'angle roulis de la caméra. Ainsi, quand l'information d'altitude imposée est erronée, notre solution basée sur *la contrainte dure en altitude* peut entrainer une dérive importante au niveau de ce paramètre causant l'échec de l'algorithme en question. Notre solution basée sur *la contrainte douce en altitude*, quant à elle, améliore l'estimation des degrés de liberté *hors plan* et semble être plus robuste face aux imperfections du MET.

5.4.2 Localisation dans un milieu péri-urbain

Dans cette section, notre objectif est d'évaluer la précision de nos approches dans des conditions réelles.

Séquence utilisée et protocole expérimental. Dans la suite, une séquence réelle représentant un parcours de 4000m d'un véhicule dans un milieu péri-urbain (Saint Quentin en Yveline) est utilisée. Cette séquence est enregistrée dans des conditions de conduite normale (50 km/h maximum). Le véhicule a été équipé par un GPS standard (1 Hz) et une caméra VGA fournissant 30 images par seconde et ayant un champs de vision $90°$. Le MET utilisé a une incertitude de $2m$.

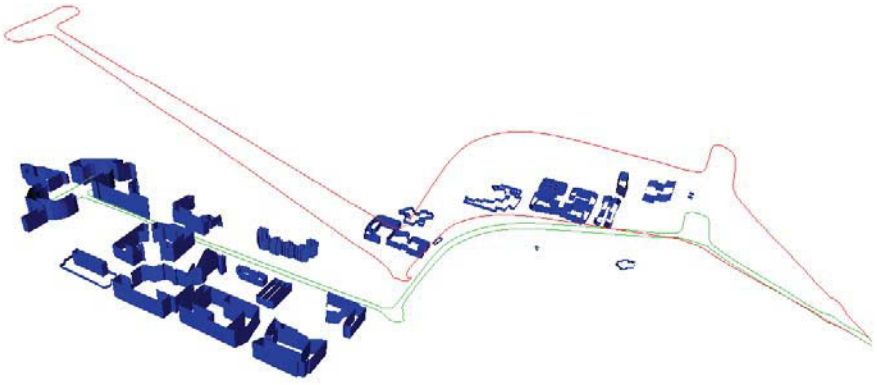
Dans cette expérimentation, nous comparons nos deux solutions fusionnant les données GPS avec le MET à l'algorithme proposé par Lhuillier (2012). La vérité terrain sur la position *dans le plan* du véhicule est obtenue par un trajectomètre IXSEA-LandINS. Étant donné qu'aucune vérité de terrain sur l'orientation de la caméra n'est disponible, l'évaluation ci-dessous est limitée à la position 3D.

Résultats. Pendant que notre solution basée sur *la contrainte dure en altitude* échoue au bout de quelques mètres à cause des dérives au niveau de l'angle roulis, nous observons dans la figure 5.6(a) que notre approche basée sur *la contrainte douce en altitude* permet d'avoir une bonne localisation *dans le plan* (erreur moyenne sur la position 2D de la caméra est $< 0.5m$, voir le tableau 5.2).

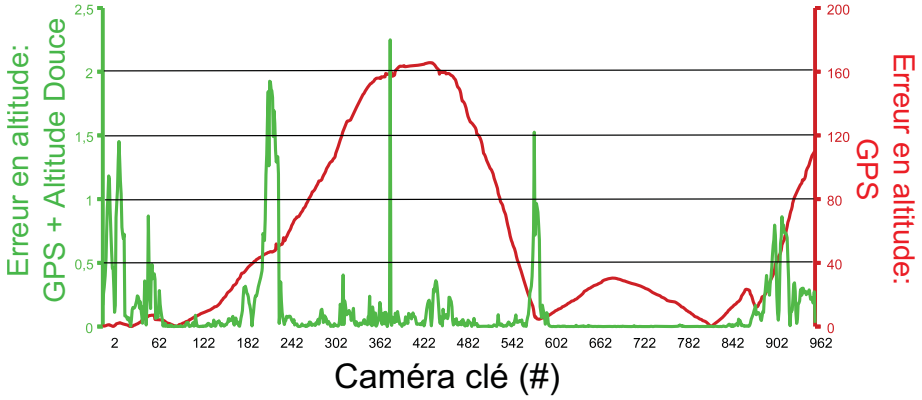
La courbe de l'évolution de l'erreur en altitude exposée dans la figure 5.6(b) montre qu'un SLAM n'intégrant que la contrainte GPS (approche de Lhuillier (2012)) est sujet à une importante dérive en altitude. Dans cette séquence, cette dérive atteint une valeur > 160m. La prise en compte de *la contrainte douce en altitude* dans l'ajustement de faisceaux contraint permet de réduire considérablement l'erreur au niveau de ce paramètre (*e.g.* une erreur maximale de 2.2m peut être due aux imprécisions du MET dont l'incertitude peut atteindre 2m).

	SLAM + GPS + *Altitude Douce*	SLAM + GPS (Lhuillier (2012))
Moyenne (m)	0.49	0.41
Écart type (m)	1.02	0.83

TABLE 5.2 – **Tableau récapitulatif des erreurs de localisation *dans le plan* pour la séquence réelle enregistrée en Saint Quentin en Yveline**.



(a)



(b)

FIGURE 5.6 – **Localisation dans le contexte péri-urbain.** Les modèles 3D des bâtiments sont représentés en bleus. En rouge le résultat de la localisation obtenu avec l'algorithme de Lhuillier (2012). En vert la localisation obtenue par notre solution basée sur l'ajustement de faisceaux contraint aux données GPS et *la contrainte douce en altitude*. (a) Une vue 3D des deux localisations obtenues. (b) Évolution de l'erreur en altitude. Notons que deux échelles de l'axe des ordonnées sont utilisées pour tracer les deux courbes.

5.5 Conclusion et perspectives

Dans ce chapitre nous avons étudié deux solutions pour fusionner les mesures GPS et la contrainte en altitude fournie par le MET afin de mieux estimer les six degrés de liberté d'une caméra mobile dans un contexte urbain. Pour les deux solutions, les différentes contraintes sont introduites directement dans le processus d'optimisation du SLAM permettant ainsi une localisation en ligne et temps réel ($30Hz$). Les deux solutions proposées ont également l'avantage d'être facile à déployer puisqu'elles ne nécessitent pas de matériels sophistiqués (*i.e.* uniquement un GPS standard et le MET largement disponible). La première solution étend *la contrainte dure en altitude* introduite dans la section 4.1 en l'intégrant dans le SLAM contraint aux données GPS proposé par Lhuillier (2012). Cette contrainte supplémentaire permet le respect strict de l'information en altitude introduite au niveau du processus d'optimisation. Toutefois, quand le MET utilisé est imprécis, la contrainte en altitude introduite est erronée ce qui peut perturber la convergence de l'ajustement de faisceaux contraint. En effet, contrairement aux contraintes fournies par les modèles 3D des bâtiments, les mesures GPS ne permettent pas de contraindre l'angle roulis de la caméra. Par conséquent, notre solution intégrant au SLAM la contrainte GPS et *la contrainte dure en altitude* peut dériver d'une façon notable au niveau de ce paramètre peu contraint. Ceci rend cette solution inexploitable dans le contexte réel où les imperfections de la route ne sont pas modélisées dans le MET. Pour apporter plus de robustesse face aux incertitudes du MET, nous avons introduit une deuxième approche basée sur *la contrainte douce en altitude*. En plus de sa robustesse face aux données aberrantes du GPS et aux incertitudes du MET, les expérimentations ci-dessus montrent que cette approche permet d'améliorer la qualité de la localisation en réduisant les dérives des degrés de liberté *hors plan* tout en assurant une précision *dans le plan* équivalente à celle obtenue par Lhuillier (2012). Toutefois, malgré ces améliorations, certaines imperfections au niveau de l'angle roulis peuvent se produire étant donné que ce paramètre reste mal contraint. Pour faire face à cette limitation, il est possible d'introduire une nouvelle contrainte en orientation dans l'ajustement de faisceaux. Le principe de cette contrainte sera détaillé dans le chapitre dédié aux perspectives à la fin de ce mémoire.

Bilan

Dans cette première partie de ce mémoire nous avons proposé deux approches différentes pour estimer en ligne les six degrés de liberté d'une caméra mobile dans des milieux urbains et péri-urbains :

- ▷ **SLAM contraint aux modèles des bâtiments et au MET**. Dans la première solution, nous avons proposé une formulation du problème de localisation qui comporte à la fois une contrainte géométrique sur la reconstruction apportée par les modèles 3D des bâtiments et une contrainte sur la trajectoire de la caméra basée sur le MET. Les deux contraintes utilisées sont intégrées directement dans le processus d'optimisation. Tandis que les bâtiments fournissent des informations géométriques contraignant le facteur d'échelle et la position *dans le plan* de la caméra ainsi que son angle roulis, la contrainte apportée par le MET, appliquée sur la trajectoire de la caméra, permet quant à elle de contraindre les degrés de liberté restant (l'angle tangage et principalement l'altitude). Pour ceci, nous avons étendu l'approche de Tamaazousti et al. (2011) en lui intégrant une *contrainte dure en altitude* basée sur une nouvelle paramétrisation de la pose de la caméra. En plus de la réduction de nombres de degrés de liberté à optimiser (*i.e.* de six à cinq paramètres pour chaque pose de la caméra), cette contrainte supplémentaire permet un respect strict de la contrainte en altitude. Les expérimentations réalisées dans la section 4.4 montre que cette solution permet de réduire les imprécisions de la localisation dans les milieux urbains denses. Toutefois, cette précision atteinte reste dépendante des imprécisions des modèles des bâtiments et se dégrade considérablement dans les milieux péri-urbains où peu de bâtiments sont visibles. Pour conclure, notre solution basée sur un SLAM avec un ajustement de faisceaux contraint aux modèles des bâtiments et au MET est précise mais peu robuste aux imprécisions et à la disponibilité des modèles des bâtiments.
- ▷ **SLAM contraint aux données GPS et au MET**. Cette solution intègre les contraintes GPS et MET au niveau de son processus d'optimisation. Tandis que les mesures GPS contraignent la position *dans le plan* de la caméra, le MET permet de limiter les dérives sur les degrés de liberté *hors plan*. Pour ceci, nous nous sommes basés sur la solution introduite par Lhuillier (2012) qui a l'avantage d'être robuste face aux données aberrantes des GPS et nous lui avons intégré la contrainte du MET. Deux approches ont été étudiées pour assurer la fusion des deux contraintes. La première est basée sur *la contrainte dure en altitude* qui est semblable à celle proposée pour le SLAM contraint aux mo-

dèles des bâtiments et au MET. La deuxième nommée *contrainte douce en altitude* où l'information d'altitude est intégrée au niveau du terme d'accroche aux données de la fonction de coût de l'ajustement de faisceaux avec la contrainte inégalité. Les différentes expérimentations montrent que la solution basée sur *la contrainte douce en altitude* est plus robuste que celle basée sur *la contrainte dure en altitude* face aux imperfections du MET offrant ainsi une meilleure précision des degrés de liberté *hors plan* de la caméra. Toutefois, la précision de la localisation *dans le plan* de ces solutions dépend de celle du GPS. Ceci rend leur utilisation limitée aux milieux ruraux et péri-urbains où la précision du GPS s'améliore. Par conséquent, ces approches sont robustes mais avec une précision des degrés de liberté *dans le plan* limitée à celle du GPS.

Dans la deuxième partie de ce mémoire, nous nous intéresserons à la fusion des différentes contraintes (*i.e.* bâtiments, MET et GPS) afin d'avoir une solution à la fois robuste, précise et générique pour les milieux urbains.

Deuxième partie

Fusion des contraintes apportées par le GPS, le MET et les modèles des bâtiments pour une localisation précise d'une caméra dans un milieu urbain : Application à la Réalité Augmentée

Contenu de la partie

Présentation des méthodes proposées **115**

6 Fusion hors ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Création d'une base d'amers géo-référencée précise **117**

6.1 Introduction 117
6.2 Positionnement et principe de notre approche 118
6.3 Création d'une base d'amers 3D géo-référencée 119
6.4 Évaluation expérimentale 122
6.5 Conclusion et perspectives 128

7 Fusion en ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Correction du biais du GPS **135**

7.1 Positionnement et principe de notre approche 135
7.2 GPS différentiel basé sur les modèles 3D des bâtiments 137
7.3 SLAM contraint au MET et aux données du GPS corrigées 142
7.4 Évaluation expérimentale 143
7.5 Conclusion et perspectives 152

8 Applications de Réalité Augmentée en milieu urbain **155**

8.1 Introduction 155
8.2 Approche proposée 157
8.3 Navigation en zone disposant d'une base d'amers 157
8.4 Navigation en zone dépourvue de base d'amers et mise à jour de la base 159
8.5 Transition entre zone avec base d'amers vers une zone sans base d'amers valide 160
8.6 Évaluation expérimentale 160
8.7 Discussion 161

Présentation des méthodes proposées

Dans la deuxième partie de ce mémoire, notre objectif consiste à localiser une caméra dans un milieu urbain (dense et péri-urbain) d'une façon à la fois robuste et précise tout en exploitant des matériels peu couteux et disponibles au grand public. Pour ceci, nous allons étudier la faisabilité de fusionner les contraintes apportées par le GPS, le MET et les modèles 3D des bâtiments. Alors que le MET est utilisé pour améliorer l'estimation des degrés de liberté hors plan, les données GPS ainsi que les modèles 3D des bâtiments permettent quant à eux de contraindre les degrés de liberté dans le plan respectivement en milieux péri-urbains et en milieux urbains denses. Deux solutions sont proposées en se basant sur les observations mentionnées ci-dessus. La première solution réalise la fusion des contraintes d'une façon hors ligne tandis que la deuxième solution assure une fusion en ligne des contraintes.

Objectif détaillé de l'étude réalisée

Dans la première partie de ce mémoire, nous avons proposé deux solutions en ligne pour estimer les six degrés de liberté de la caméra dans un milieu urbain ou péri-urbain. La première fusionne les contraintes géométriques apportées par les modèles 3D des bâtiments et le MET, tandis que la deuxième combine les contraintes GPS avec celles du MET. Chacune de ces solutions revisite et améliore de manière notable les solutions existantes (*i.e.* Tamaazousti et al. (2011) et Lhuillier (2012)), que ce soit en terme de robustesse ou de précision. Néanmoins, ces solutions restent insuffisantes pour un certain nombre d'applications (*e.g.* aide à la navigation via la Réalité Augmentée). En effet, le SLAM basé sur un ajustement de faisceaux contraint aux modèles 3D des bâtiments et au MET (*i.e. contrainte dure en altitude*), proposé dans le chapitre 4, est sensible aux incertitudes des modèles SIG et à l'indisponibilité des modèles 3D des bâtiments. Ceci limite son exploitation aux milieux urbains denses. D'autre part, notre approche contraignant la reconstruction SLAM avec les données GPS et *la contrainte douce en altitude*, introduite dans le chapitre 5, est robuste face aux données aberrantes du GPS et aux imperfections du MET. Cependant, la précision de la localisation obtenue reste sensiblement liée à celle du GPS. Or, l'imprécision de ce capteur est plus importante dans les milieux urbains denses à cause du phénomène de canyon urbain comme le montre la figure 5.7. Ceci limite donc l'utilisation de l'approche basée sur le GPS aux milieux péri-urbains et ruraux.

Pour assurer à la fois un algorithme de localisation précis et robuste, nous essayons dans

cette partie de fusionner efficacement l'ensemble des contraintes évoquées ci-dessus tout en évitant les éventuels problèmes de convergence. En effet, si les solutions optimales (*i.e.* le maximum de vraisemblance) pour la fusion SLAM/GPS et la fusion SLAM/Bâtiments étaient atteintes pour les mêmes paramètres, alors ce serait envisageable de fusionner ces contraintes dans un même ajustement de faisceaux. Cela supposerait que l'erreur affectant les données GPS suivent une loi de probabilité gaussienne. Or, les mesures GPS sont caractérisées, en plus du bruit gaussien, par un biais qui est important en ville en raison des canyons urbain. Pour éviter alors tout problème de convergence lié à cette fusion, deux stratégies ont été introduites :

- ▷ **Fusion hors ligne des contraintes : Création d'une base d'amers géo-référencée (chapitre 6).**
  Lors de la fusion hors ligne, les données GPS sont uniquement utilisées pour fournir une reconstruction SLAM géo-référencée initiale. Les contraintes fournies par le MET et les modèles 3D des bâtiments interviennent par la suite à travers deux ajustements de faisceaux globaux. L'ensemble de ce processus permet ainsi d'assurer une localisation hors ligne robuste et précise ;
- ▷ **Fusion en ligne des contraintes : Correction de l'incertitude du GPS (chapitre 7).**
  Pour assurer la fusion en ligne des contraintes, les bâtiments sont utilisés pour corriger le biais du GPS ce qui permet de s'approcher d'un modèle d'erreur gaussien pour les mesures GPS. Cette correction est réalisée en s'inspirant du principe de fonctionnement du GPS différentiel. Les mesures GPS, ainsi corrigées, sont exploitées dans le SLAM introduit dans la section 5.3 qui fusionne ces données avec *la contrainte douce en altitude*. Ceci permet d'offrir une localisation en ligne robuste et précise.

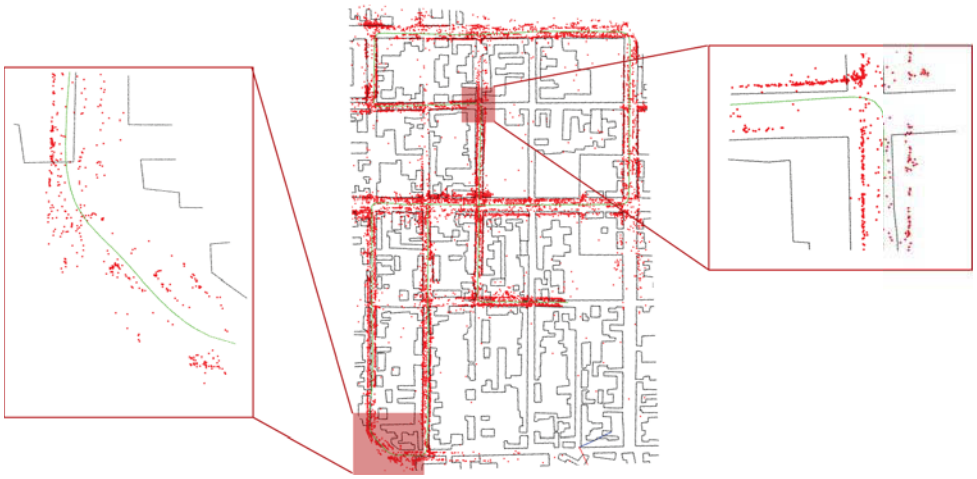


FIGURE 5.7 – **Résultat de la localisation en ligne du SLAM contraint aux données GPS et au MET dans un milieu urbain dense.** En vert : la localisation obtenue. Des erreurs de localisation *dans le plan* sont notables à cause de l'imprécision du GPS.

Une illustration de l'exploitation de ces deux approches de fusion via des applications de Réalité Augmentée sera présentée dans le chapitre 8.

CHAPITRE 6

Fusion hors ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Création d'une base d'amers géo-référencée précise

Dans ce chapitre, nous proposons une méthode hors ligne permettant d'obtenir la trajectoire du véhicule et une reconstruction éparse de son environnement qui soient géo-référencées et précises tout en exploitant des capteurs bas coût et des données standards. Pour cela, cette solution exploite non seulement les contraintes multi-vues d'un SLAM monoculaire (caméra classique) mais aussi l'ensemble des contraintes présentées dans la partie précédente de ce mémoire, à savoir les contraintes au MET, aux façades de bâtiments et aux données d'un GPS standard. Les expériences menées sur des séquences réelles montrent que, en dépit de la qualité des données et des capteurs utilisés, la solution proposée permet d'obtenir une reconstruction précise avec un coût calculatoire réduit (quelques minutes en utilisant un calculateur "mono coeur" CPU pour une trajectoire de plusieurs kilomètres). Une illustration de l'exploitation d'une telle reconstruction pour la localisation en ligne de véhicule sera présentée dans le chapitre 8.

Ces travaux ont donné lieu à deux publications internationales Larnaout et al. (2013b) et Larnaout et al. (2013c).

6.1 Introduction

Comme nous l'avons montré dans les chapitres précédents, les contraintes *douces* ou *dures* apportées par GPS, le MET ou les modèles 3D des bâtiments assurent une localisation soit robuste mais peu précise soit précise mais peu robuste. Il semble donc naturel de vouloir fusionner ces données pour en combiner les bénéfices. Cependant, l'exploitation de l'ensemble de ces contraintes peut introduire des problèmes de convergence étant donné que le SLAM contraint aux modèles 3D des bâtiments et celui contraint aux données GPS ont des solutions "optimales" différentes en raison du biais du GPS, d'où la difficulté de les combiner dans un seul ajustement de faisceaux.

La principale contribution de ce chapitre est donc de proposer un processus hors ligne permettant d'exploiter l'ensemble de ces contraintes de manière à offrir une précision élevée. Pour cela, nous proposons d'utiliser une approche *coarse to fine* de manière à exploiter en premier des données peu précises (*i.e.* GPS) mais dont l'étape d'établissement de contrainte est immédiatement disponible (*i.e.* association image clé/mesure GPS est établie à partir de la datation des données) afin d'initialiser les données aux précisions les plus élevées (*i.e.* le MET et les modèles 3D des bâtiments) et nécessitant une étape de mise en correspondance précise avec la reconstruction SLAM (*i.e.* association caméra/route et point/plan). En plus de l'utilisation des données et des capteurs standards et disponibles, la solution proposée a l'avantage d'avoir un temps de traitement réduit sur un simple calculateur "mono coeur" CPU.

Après avoir expliqué le principe général de notre méthode dans la section 6.2, celle-ci sera par la suite détaillée (voir section 6.3). Enfin les résultats seront exposés dans la section 6.4.

6.2 Positionnement et principe de notre approche

Comme nous l'avons précisé précédemment, notre objectif consiste à fusionner d'une façon hors ligne les contraintes multi-vues avec celles apportées par le GPS, le MET et les modèles 3D des bâtiments dans le but d'obtenir une base d'amers géo-référencée précise. Pour ceci, nous adoptons une approche *coarse to fine* semblable à celle utilisée par Lothe et al. (2009).

En effet, Lothe et al. (2009) ont démontré la possibilité de corriger et géo-référencer une reconstruction SLAM en exploitant des données GPS et des modèles 3D des bâtiments grossiers à travers des transformations non-rigides. Pour ceci, ils proposent un processus se déroulant en deux étapes : une première correction grossière de la reconstruction SLAM suivie par une optimisation plus fine.

Concernant la première étape, son objectif est de fournir une reconstruction SLAM géo-référencée avec peu de dérives. Pour ceci, Lothe et al. (2009) considèrent que la dérive du SLAM est observable principalement au niveau des virages. Sous cette hypothèse, ils proposent de modéliser la dérive en question à travers une transformation affine par morceaux où chaque morceau représente une ligne droite. Par conséquent, l'ensemble des poses de la caméra appartenant à une même ligne subissent la même correction. Les poses de la caméra aux extrémités, appartenant quant à elles à deux lignes droites, doivent respecter la transformation affine associées à chacun des deux morceaux. Ceci garantit que la trajectoire SLAM reste unie. Pour estimer la transformation en question, les données GPS au niveau des virages sont exploitées.

Une fois le premier recalage grossier terminé, la reconstruction résultante est raffinée à l'aide d'un algorithme d'ICP non rigide. Au cours de cette étape, l'hypothèse de dérive uniquement au niveau des virages n'est pas encore remise en cause. Toutefois, des nouvelles données plus précises que le GPS (*i.e.* modèles 3D des bâtiments) sont exploitées afin de raffiner la reconstruction SLAM résultante. Ceci est réalisé en minimisant la distance entre le nuage de points corrigé et les modèles 3D de la ville. La reconstruction SLAM n'incluant pas uniquement des points appartenant aux façades, une étape de segmentation du nuage de points est utilisée pour identifier les points modélisant des façades. L'ensemble des points restant ne participe pas à l'ICP non rigide.

Afin d'accroitre la précision de la reconstruction, une seconde étape d'optimisation reposant sur un ajustement de faisceaux global contraint aux bâtiments (voir section 3.2.2) est utilisée. Cette seconde étape permet de remettre en cause l'hypothèse de dérive principalement au niveau des virages. Par ailleurs, à la différence de l'ICP non rigide, cette étape favorise un respect de la

contrainte multi-vues sur l'ensemble de la trajectoire, y compris au niveau des virages (ce qui n'est pas le cas de l'ICP non rigide). Ici aussi, seuls les points identifiés comme appartenant à une façade de bâtiment sont utilisés au cours du raffinement.

Malgré les résultats prometteurs, cette solution présente certaines limitations. En effet, son étape d'initialisation sous-exploite les données GPS (uniquement aux virages) et ne traite pas les variations d'altitude ce qui entraine une reconstruction initiale peu précise pouvant perturber l'étape de raffinement à l'aide des modèles de bâtiments (*i.e.* l'ICP non-rigide et l'ajustement de faisceaux global). D'autre part, utiliser uniquement des points associés aux modèles des bâtiments rend cette solution limitée aux zones urbaines denses. Par ailleurs, durant l'optimisation, tous les degrés de liberté sont raffinés tandis que les bâtiments contraignent principalement les degrés de liberté *dans le plan*. Par conséquent, les paramètres *hors plan* (principalement l'altitude et l'angle tangage) peuvent être détériorés ceci est d'autant plus vrai si la qualité de l'initialisation est mauvaise.

Pour résoudre ces problèmes, plusieurs améliorations seront introduites dans la suite de ce chapitre. Premièrement, nous proposons de tirer un apport plus important du GPS afin d'améliorer l'estimation des degrés de liberté *dans le plan* au cours de l'initialisation. Pour apporter plus de précision aux degrés de liberté hors plan, le MET est également exploité. Par ailleurs, dans le but d'étendre la solution de Lothe et al. (2009) aux milieux péri-urbains et garantir plus de robustesse quand peu de bâtiments sont observables, nous proposons de prendre en compte, dans l'étape de raffinement de la reconstruction initiale à partir des modèles SIG, à la fois les contraintes géométriques des points 3D associés aux modèles et les contraintes multi-vues fournies par l'ensemble de points 3D représentant le reste de l'environnement. Les différentes étapes de notre méthode seront détaillées dans la section suivante.

6.3 Création d'une base d'amers 3D géo-référencée

Dans cette section, nous détaillerons notre approche de création d'une base d'amers géo-référencée de l'étape de l'initialisation exploitant les données GPS et le MET (section 6.3.1) à l'étape du raffinement à partir du MET et les modèles des bâtiments (section 6.3.2).

6.3.1 Base d'amers géo-référencée initiale

Pour obtenir une localisation hors ligne précise, nous proposons de raffiner une reconstruction SLAM géo-référencée préalablement créée. Toutefois, le raffinement à l'aide des bâtiments ne peut garantir une précision élevée que si la reconstruction initiale est proche de la solution optimale. Pour cette raison, nous avons choisi de créer notre base d'amers initiale (*i.e.* reconstruction SLAM) à l'aide du SLAM contraint aux données GPS et au MET décrit dans le chapitre précédent (voir chapitre 5). Plus précisément, nous adoptons la solution basée sur *la contrainte douce en altitude* (section 5.3). Celle-ci permet d'obtenir une reconstruction géo-référencée de manière entièrement automatique et avec une bonne robustesse en raison de la contrainte d'inégalité. De plus, les performances de cet algorithme sont compatibles avec une exécution en ligne (la reconstruction peut être réalisée simultanément à l'acquisition des données) réduisant ainsi le temps de traitement a posteriori. Néanmoins, comme le montre la figure 5.7, l'erreur affectant les degrés de liberté *dans le plan* reste généralement élevée en raison du biais des données GPS. Afin de corriger ces imprécisions sans perturber les degrés de liberté *hors plan*, un raffinement a posteriori est réalisé incluant les contraintes aux modèles 3D des bâtiments et au

MET.

6.3.2 Raffinement de la base d'amers géo-référencée initiale

Comme il est mentionné ci-dessus, après l'étape d'initialisation, une base d'amers géo-référencée est obtenue. A cause des importantes incertitudes affectant les données GPS, les erreurs au niveau des degrés de liberté *dans le plan* sont généralement supérieures à celles caractérisant les paramètres *hors plan*. Pour corriger ces imprécisions, Lothe et al. (2009) effectuent un ajustement de faisceaux global exploitant uniquement les modèles des bâtiments. Pour réaliser cette étape, seuls les points 3D associés aux modèles des bâtiments sont utilisés ce qui peut entrainer des problèmes de convergence quand peu de bâtiments sont visibles. De plus, tous les degrés de liberté de la reconstruction SLAM sont raffinés tandis que les bâtiments contraignent principalement les paramètres *dans le plan*. Ceci peut détériorer la précision des degrés de liberté *hors plan* si la qualité de l'initialisation est mauvaise. Pour faire face à ces deux limitations, nous souhaitons alors :

- ▷ Exploiter à la fois les points 3D associés aux modèles des bâtiments $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ et ceux appartenant au reste de l'environnement $\{\mathcal{Q}_i\}_{i\in\mathcal{E}}$. Ceci permet plus de robustesse dans les zones où peu de bâtiments sont visibles ;
- ▷ Exploiter à la fois les contraintes fournies par les bâtiments et celles apportées par le MET. Ceci permet de contraindre l'ensemble des degrés de liberté de la reconstruction SLAM.

Une solution possible consiste à utiliser l'ajustement de faisceaux introduit dans la section 4.2 qui intègre *des contraintes dures* fournies par le MET et les modèles 3D des bâtiments. Néanmoins, nous avons montré expérimentalement dans la sections 4.4.1 que *la contrainte dure en altitude* est peu robuste aux imprécisions du MET. Pour assurer plus de robustesse face à ce problème, nous privilégions alors l'utilisation de *la contrainte douce en altitude*. Pour ceci, nous réalisons un ajustement de faisceaux avec contrainte d'inégalité dont le principe est similaire à celui introduit dans la section 5.3. Cependant, à présent, seul le terme d'accroche aux données lié à *la contrainte douce en altitude* (voir equation 5.10) est utilisé. Les données GPS, quant à elles, ne sont pas exploitées à cause de leurs imprécisions importantes. Ainsi leur utilisation est plus susceptible de perturber la convergence du raffinement. D'autres modifications sont apportées à l'ajustement de faisceaux avec contrainte d'inégalité pour prendre en compte *les contraintes dures* apportées par les modèles des bâtiments. Ces modifications interviennent dans les deux étapes nécessaires à la réalisation de l'ajustement de faisceaux avec contrainte d'inégalité (plus de détails sur ces étapes sont disponibles dans la section 3.3.3) :

- ▷ Lors du premier ajustement de faisceaux nécessaire à l'estimation de la dégradation maximale tolérée de l'erreur de re-projection e_t, l'ajustement de faisceaux en question doit intégrer la contrainte aux bâtiments afin que cette dernière soit respectée avant l'intégration de la contrainte au MET via *la contrainte douce* ;
- ▷ Lors du deuxième ajustement de faisceaux permettant de réaliser la fusion de la contrainte aux bâtiments et *la contrainte douce en altitude*, le terme de régularisation doit inclure les contraintes aux bâtiments afin d'empêcher une détérioration significative des résultats du premier ajustement de faisceaux.

Dans ce qui suit, ces deux différentes étapes seront détaillées.

6.3.2.1 Recalage dans le plan utilisant les modèles 3D des bâtiments

Comme nous l'avons mentionné ci-dessus, afin de s'assurer que la contrainte aux bâtiments soit respectée avant l'intégration des informations issues du MET, le premier ajustement de faisceaux global doit inclure cette contrainte. Pour ceci nous remplaçons la fonction de coût standard par la fonction de coût bi-objective introduite par Tamaazousti et al. (2011). En plus d'intégrer *la contrainte dure* aux bâtiments, cette dernière a l'avantage d'exploiter à la fois les points 3D associés aux modèles des bâtiments et ceux appartenant au reste de l'environnement. L'objectif de cet ajustement de faisceaux global contraint est double. En effet, il permet d'une part d'estimer les différents paramètres qui seront utilisés par la suite lors du deuxième ajustement de faisceaux intégrant *la contrainte douce en altitude* (*i.e.* poids du terme d'accroche aux données et la dégradation maximale de l'erreur de re-projection). D'autre part, il permet d'établir un premier raffinement de la base d'amers géo-référencée initiale afin que la contrainte aux bâtiments soit respectée. Étant donné que les bâtiments contraignent principalement les degrés de liberté *dans le plan* et pour ne pas perturber la précision des degrés de liberté peu contraints, seuls les paramètres *dans le plan* sont optimisés au cours de ce premier ajustement de faisceaux. Par conséquent, si nous notons κ' le vecteur de paramètres incluant les points 3D et les paramètres *dans le plan* des poses de la camera à optimiser alors la fonction de coût minimisée est donnée par :

$$g\left(\kappa'\right) = \sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{D}_i} \rho\left(\|\mathbf{f}_{i,j}\|, c\right) + \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{D}_i} \rho\left(\|\mathbf{g}_{i,j}\|, c\right), \tag{6.1}$$

avec $\mathbf{f}_{i,j}$ est l'erreur de re-projection standard du $i^{ème}$ point 3D $\tilde{\mathcal{Q}}_i$ observé par la $j^{ème}$ caméra :

$$\mathbf{f}_{i,j} = \mathbf{q}_{i,j} - \pi(\mathsf{K}\mathcal{R}_j^T\left[\mathsf{I}_{3\times3}\right| - \mathbf{t}_j]\,\tilde{\mathcal{Q}}_i) \tag{6.2}$$

$\mathbf{g}_{i,j}$ est l'erreur de re-projection intégrant la contrainte aux bâtiment du $i^{ème}$ point 3D $\mathcal{Q}_i^{h_i}$, associé au modèle, observé par la $j^{ème}$ caméra :

$$\mathbf{g}_{i,j} = \mathbf{q}_{i,j} - \pi(\mathsf{K}\mathcal{R}_j^T\left[\mathsf{I}_{3\times3}\right| - \mathbf{t}_j]\,(\tilde{\mathsf{M}}^{h_i})^{-1}\tilde{\mathcal{Q}}_i^{h_i}), \tag{6.3}$$

et c est le seuil de rejet du M-estimateur fixé comme décrit dans la section 3.2.3.

6.3.2.2 Fusion des contraintes des modèles 3D des bâtiments et du MET

Une fois le premier ajustement de faisceaux global terminé, le deuxième ajustement de faisceaux global avec la contrainte d'inégalité peut être réalisé. Au cours de ce deuxième raffinement, tous les degrés de liberté de la reconstruction SLAM sont contraints. Tandis que les bâtiments contraignent les degrés de liberté *dans le plan* et l'angle roulis, le MET fournit des informations relatives aux degrés de liberté restant. Par conséquent, la fonction de coût utilisée à présent est composée par un terme de régularisation incluant l'erreur de re-projection qui prend en compte la contrainte des bâtiments et un terme d'attache aux données calculé à partir de la contrainte MET. Inclure la contrainte aux bâtiments dans le terme de régularisation est indispensable afin que le premier recalage ne soit pas trop détérioré. Le terme d'attache aux données, quant à lui, représente la distance entre l'altitude de chaque pose de la caméra $(\mathbf{t}_j^{k_j})_z$ exprimée dans le repère de la route associé et l'altitude h souhaitée. La fonction de coût résultante est donnée par :

$$f_{3_I}\left(\varsigma\right) = \frac{\omega}{e_t - g\left(\varsigma\right)} + \|\mathsf{M}_3\varsigma + \mathbf{m} - \mathbf{h}\|^2, \tag{6.4}$$

avec $\varsigma = \left(\left\{\alpha_j, \beta_j, \gamma_j, \mathbf{t}_j^T\right\}_{j=1}^{N}, \{X_i, Y_i, Z_i\}_{i\in\mathcal{E}}, \left\{X_i^{h_i}, Y_i^{h_i}\right\}_{i\in\mathcal{M}}\right)^T$ où $\alpha_j, \beta_j, \gamma_j$ sont les angles Euler associés à la $j^{ème}$ orientation de la caméra, $\left\{X_i^{h_i}, Y_i^{h_i}\right\}$ sont les coordonnées non nulles du point $\mathcal{Q}_i^{h_i}$ dans le repère du plan du bâtiments $\mathbf{\Pi}^{h_i}$.

$\mathsf{M}_3 = (\mathsf{D}_{3_{N\times 6N}}|\mathsf{0}_{N\times S_1}|\mathsf{0}_{N\times S_2})$, avec $S_1 = 3 \times \text{card}(\mathcal{E})$, $S_2 = 2 \times \text{card}(\mathcal{M})$ et tel que $\mathsf{D}_{3_{N\times 6N}}$ est une matrice diagonale par bloc de 1×6 où chaque bloc $\mathbf{d}_{3_j} = \begin{pmatrix}0 & 0 & 0 & \mathsf{L}^{k_j}(3,1) & \mathsf{L}^{k_j}(3,2) & \mathsf{L}^{k_j}(3,3)\end{pmatrix}$. D'autre part, $\mathbf{m} = (\mathsf{L}^{k_1}(3,4)...\mathsf{L}^{k_N}(3,4))^T$.

Le poids ω et la dégradation e_t sont estimés comme suit :

$$\omega = \frac{e_t - g(\kappa^*)}{10} \times \|\mathsf{M}_3\kappa^* + \mathbf{m} - \mathbf{h}\|^2, \tag{6.5}$$

et

$$e_t = 1.05 \times g(\kappa^*), \tag{6.6}$$

où κ^* représente le vecteur des paramètres, incluant les degrés de liberté *dans le plan* et *hors plan* de la caméra, à l'issue du premier ajustement de faisceaux. $\tilde{\mathsf{L}}^{k_j}$ est la matrice de passage (3×4) du repère monde au repère du plan de la route $\mathbf{\Lambda}^{k_j}$ (voir figure 4.1). Plus de détails sur la minimisation de cette fonction de coût sont disponibles dans l'annexe (section A.3.3.2). L'algorithme 6 résume le processus d'optimisation avec la contrainte d'inégalité intégrant les informations du MET et des modèles 3D des bâtiments.

6.4 Évaluation expérimentale

Dans cette section, nous proposons une évaluation complète de notre processus de fusion de contraintes sur des données de synthèse et d'autres réelles (section 6.4.1). L'objectif de cette évaluation expérimentale est double :

1. Tout d'abord, nous souhaitons démontrer la robustesse du processus en évaluant ce dernier sur des séquences de synthèse et d'autres réelles de plusieurs kilomètres (voir section 6.4.2).
2. Ensuite, nous souhaitons démontrer les performances de notre algorithme en comparant les résultats de ce dernier avec la méthode Lothe et al. (2009) (voir section 6.4.3).

Dans ce qui suit nous commençons par présenter les séquences utilisées.

6.4.1 Séquences de tests utilisées

Pour évaluer notre algorithme de création de base d'amers géo-référencée à travers la fusion hors ligne des contraintes, multi-vues, GPS, MET et modèles 3D des bâtiments, nous avons eu recours à des données synthétiques et d'autres réelles.

Séquence de synthèse La séquence de synthèse utilisée est illustrée dans la figure 6.1. Cette séquence simule le parcours d'un véhicule dans un milieu urbain dense. La caméra est embarquée sur le véhicule à une altitude de 1.5m par rapport au plan de la route supposé parfaitement plat. Pour simuler les données GPS, les positions *dans le plan* de la caméra, fournie par la vérité terrain, sont perturbées selon le modèle de bruit décrit dans la section 1.4.2.1. Ainsi pour chaque position *dans le plan*, un biais d'amplitude 5m auquel est additionné un bruit Gaussien d'amplitude 1m est rajouté.

$\texttt{nouveauMAD} = 0$;
$\texttt{ancienMAD} = 0$;
repeat
  Segmenter $\{\mathcal{Q}_i\}_{i=1}^{M}$ en $\{\mathcal{Q}_i\}_{i\in\mathcal{E}}$ et $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Projeter les points 3D $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ sur leurs plans correspondants ;
  $\texttt{ancienMAD} = \texttt{MAD}$ calculé sur les erreurs de re-projection de points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Calculer le seuil de rejet du M-estimateur c ;
  Minimiser la fonction de coût 6.1 en utilisant l'algorithme de Levenberg Marquardt ;
  $\texttt{nouveauMAD} = \texttt{MAD}$ calculé sur les erreurs de re-projection de points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Triangulation des points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ en prenant en compte les nouvelles poses de la caméra ;
until $(\texttt{nouveauMAD} - \texttt{ancienMAD}) < 0.1$;
$\texttt{nouveauMAD} = 0$;
$\texttt{ancienMAD} = 0$;
Calculer le poids ω (equation 6.5) et la dégradation maximale tolérée e_t (equation 6.6) ;
repeat
  Associer chaque caméra au plan de la route le plus proche dans le MET ;
  Segmenter $\{\mathcal{Q}_i\}_{i=1}^{M}$ en $\{\mathcal{Q}_i\}_{i\in\mathcal{E}}$ et $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  Projeter les points 3D $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$ sur leurs plans correspondants ;
  $\texttt{ancienMAD}$ = MAD calculé sur les erreurs de re-projection des points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
  En utilisant l'algorithme du Levenberg Marquardt, minimiser la fonction de coût 6.4 avec la contrainte d'inégalité introduite dans la section 6.3.2.2 ;
  $\texttt{nouveauMAD}$ = MAD calculé sur les erreurs de re-projection des points $\{\mathcal{Q}_i\}_{i\in\mathcal{M}}$;
until $(\texttt{nouveauMAD} - \texttt{ancienMAD}) < 0.1$ *et*
(*les associations caméra/plan de routes soient stabilisées*);

Algorithme 6 : Processus d'optimisation avec la contrainte d'inégalité intégrant les informations du MET et des modèles 3D des bâtiments.

(a) (b)

FIGURE 6.1 – **Illustrations de la séquence de synthèse utilisée.**

Séquences réelles Les performances de notre méthode sont également vérifiées sur trois séquences réelles, voir figure 6.2. Ces séquences sont enregistrées dans deux quartiers différents (deux séquences dans la ville de Versailles et une séquence dans la ville de Saclay) dans des conditions de conduite normale (50Km/h). Pour ceci, le véhicule a été équipé par un GPS standard 1Hz et une caméra RGB fournissant 30 images par seconde et avec un champs de vision de 90°. Les modèles SIG utilisés sont issus de la base Géoportail de l'IGN et ont généralement une erreur ne dépassant pas les 2m.

Les trois séquences utilisées représentent des parcours de 2400m, de 1800m et 1200m. Notons que même si ces séquences ne couvrent pas tout un quartier, il est possible, comme le montre la figure 6.3, de fusionner plusieurs bases d'amers d'un même quartier pour créer une base d'amers à l'échelle d'une ville.

6.4.2 Évaluation de l'ensemble de notre solution pour la création d'une base d'amers géo-référencés

Protocole expérimental. Pour évaluer la capacité de notre algorithme à fonctionner dans des conditions normales de circulation, nous proposons d'étudier la qualité de la reconstruction fournie par notre méthode sur les séquences réelles décrites ci-dessus. Ne disposant pas de vérité terrain pour ces expériences, cette première évaluation se limite à une appréciation visuelle, que ce soit à travers la visualisation de la trajectoire et des points 3D reconstruits par rapport aux modèles 3D des bâtiments (voir figure 6.4) ou la projection de ces modèles dans différentes vues des séquences de test (voir figure 6.5). Afin de mettre en évidence l'impact du processus d'optimisation, ces résultats seront présentés pour les différentes étapes du processus.

Résultat sur des données réelles. Pour créer la base d'amers souhaitée, seuls quelques minutes sont nécessaires en utilisant un code peu optimisé [1] exécuté sur un Intel(R) Xeon(R) CPU quad cores 2.4 GHz. En effet, pour la séquence de Versailles parcourant une distance de 2400m, la base initiale, de volume 8Mo, contenant 548 vues géo-référencées et 34178 points 3D est obtenue en ligne. Ensuite, 50 secondes sont uniquement demandées pour réaliser le recalage dans le plan décrit dans la section 6.3.2.1 et 90 secondes sont nécessaires pour effectuer le dernier raffinement introduit dans la section 6.3.2.2.

1. Le GPU n'est pas utilisé.

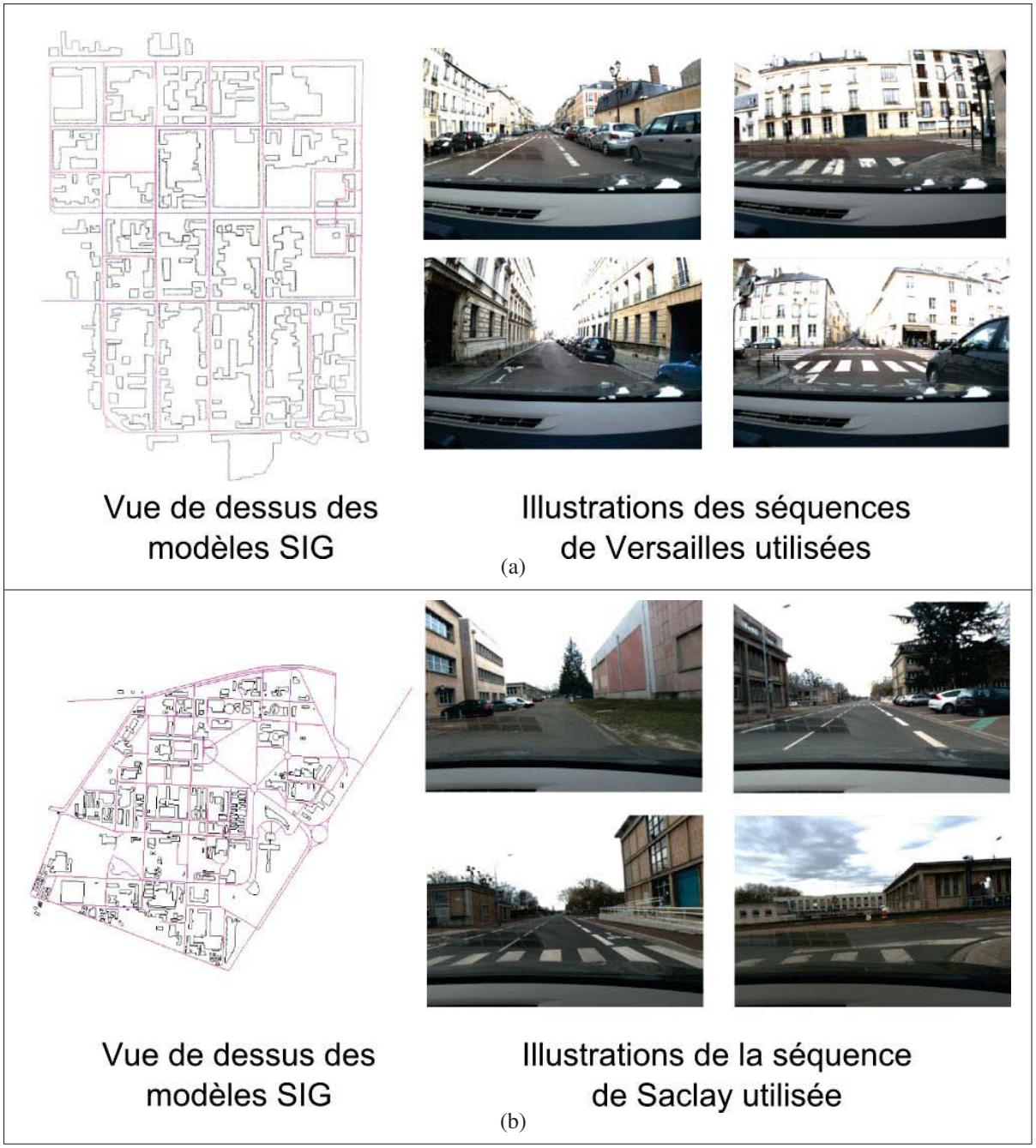


FIGURE 6.2 – **Illustrations des séquences réelles utilisées et les modèles SIG associés.** (a) Séquences de Versailles. (b) Séquence de Saclay. La colonne de gauche représente une vue de dessus des modèles SIG utilisés pour chacune des séquences : en noir les modèles 3D des bâtiments, en rose les MET.

Comme le montre la figure 6.4, la fusion du SLAM avec les données GPS et MET, décrit dans la section 5.3, permet de créer automatiquement des bases de données géo-référencées (les reconstructions rouges dans la figure 6.4). Toutefois, ces reconstructions initiales sont objet d'importantes imprécisions au niveau des degrés de liberté *dans le plan* comme il est montré dans la figure 6.4 et mis en évidence dans les figures 6.5(d) et 6.5(a). Une fois le recalage *dans le plan* effectué, ces incertitudes sont corrigées. Par conséquent, les re-projection des modèles des bâtiments sont mieux alignés avec les façades observées dans le flux vidéo (voir les figures

FIGURE 6.3 – **Concaténation de deux bases de données différentes (bleue et verte) dans le quartier de Versailles.**

6.5(e) et 6.5(b). Enfin, le dernier raffinement exploitant le modèle SIG complet permet de corriger les imperfections restantes sur les degrés de liberté *hors plan* (voir figures 6.5(f) et 6.5(c)). Cette correction est réalisée sans perturber les paramètres *dans le plan* comme le montre la figure 6.4 où le nuage de points final représenté en bleu est parfaitement recalé sur les modèles des bâtiments.

6.4.3 Comparaison avec l'approche de Lothe et al. (2009)

Protocole expérimental Dans cette section, nous comparons la précision de nos bases d'amers avec celles obtenues avec la méthode de Lothe et al. (2009). Principalement, nous souhaitons comparer les deux méthodes d'optimisation utilisées : la fusion des contraintes aux bâtiments et au MET à travers une optimisation intégrant une contrainte d'inégalité contre l'ICP non rigide suivi par l'ajustement de faisceaux global contraint uniquement aux modèles des bâtiments introduit par Lothe et al. (2009). Pour ceci, les deux méthodes d'optimisation sont initialisées en utilisant le SLAM contraint aux données GPS et la contrainte douce en altitude qui fournit une reconstruction initiale assez précise. La comparaison en question est effectuée dans un premier temps sur la séquence de synthèse dont nous possédons la vérité terrain. Une évaluation sur les séquences réelles est par la suite proposée.

Pour la séquence de synthèse, l'évaluation est réalisée en analysant l'évolution de l'erreur de la localisation *dans le plan*, l'erreur en altitude, ainsi que l'erreur angulaire pour les deux bases obtenues.

La comparaison entre notre solution et celle proposée par Lothe et al. (2009) est également

établie sur les séquences réelles. Étant donné que la vérité terrain n'est pas disponible pour ces séquences, nous labellisons manuellement les coins des bâtiments [2] dans quelques images extraites des flux vidéos (environs 20 images distribuées uniformément dans chaque séquence) comme vérité terrain. Nous calculons par la suite l'erreur de re-projection entre les coins labellisés et la re-projection des coins des modèles des bâtiments.

Évaluation sur la séquence de synthèse La figure 6.6 inclut l'erreur de la localisation *dans le plan*, l'erreur en altitude, ainsi que l'erreur angulaire pour les deux bases obtenues. Nous remarquons que la localisation dans le plan est assez précise pour l'algorithme de Lothe et al. (2009). Toutefois, notre solution améliore cette précision en réduisant la moyenne des erreurs de 0.3m à 0.1m. La même observation est notée pour les degrés de liberté hors plan. En effet, notre solution réduit la moyenne des erreurs en altitude de 0.5m à une erreur négligeable ($\simeq 0$m) tandis que la moyenne des erreurs en orientation est passée de 0.012rad ~ 0.68deg à 0.003rad ~ 0.17deg.

Évaluation sur les séquences réelles Les résultats de comparaison entre notre solution et celle proposée par Lothe et al. (2009) sont résumés dans le tableau 6.1. La mesure d'erreur de re-projection inclut surement les incertitudes des modèles des bâtiments et celles de la labellisation manuelle. Cependant, ces incertitudes restent faibles en comparaison avec l'amélioration notable que notre solution apporte. En effet, pour les séquences de Versailles, l'approche proposée réduit de moitié la moyenne des erreurs de re-projections obtenue par l'algorithme de Lothe et al. (2009). L'écart-type des erreurs mesurées a également baissé considérablement en utilisant notre méthode de 13.3 pixels à 1.91 pixels. Ces résultats mettent en évidence la bonne précision que notre solution assure contrairement à la méthode de Lothe et al. (2009) qui présente des imprécisions locales et globales. En effet, utiliser uniquement les points 3D associés aux modèles des bâtiments cause des imprécisions locales quand peu de bâtiments sont observables comme le montre la figure 6.7 où le nuage de points (en orange) n'est pas aligné avec les empreintes des bâtiments. De plus, optimiser tous les degrés de liberté à travers un ajustement de faisceaux contraint uniquement aux modèles des bâtiments entraine des imprécisions globales principalement observées au niveau des paramètres *hors plan* (voir figure 6.8).

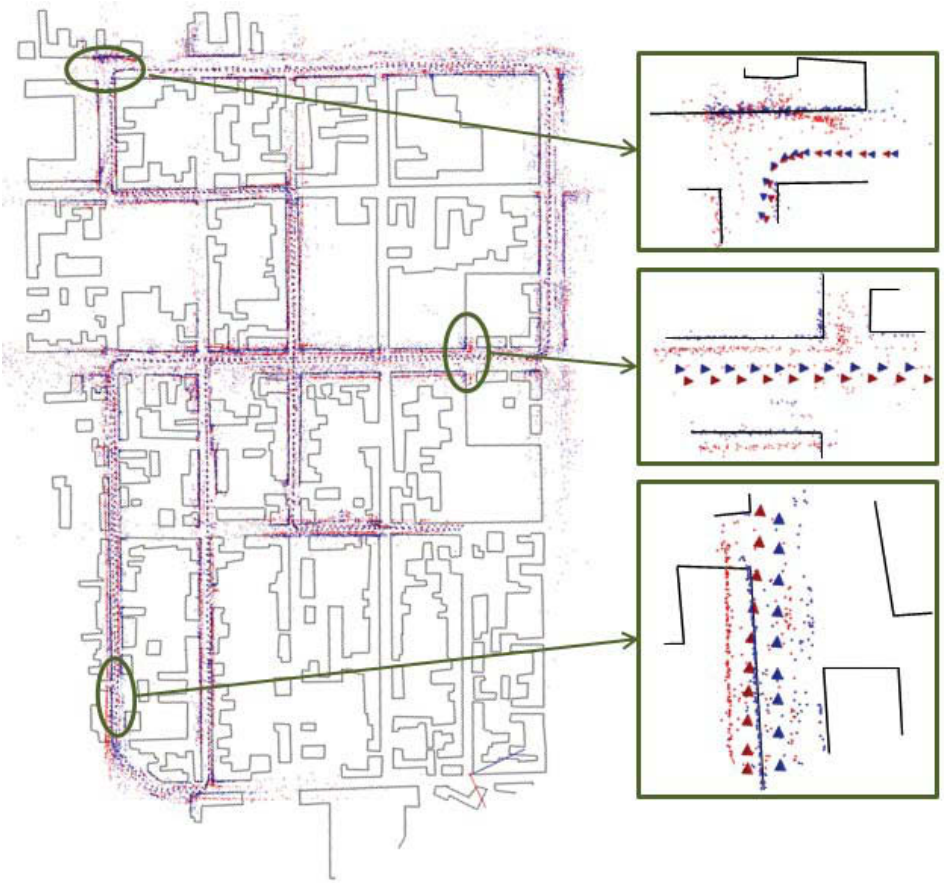
		Erreur de re-projection (pixels)			
		Moyenne	Écart type	Max	Min
Versailles	Lothe et al. (2009)	17.85	13.30	40.90	4.11
	Notre méthode	7.32	1.91	10.90	4.04
Saclay	Lothe et al. (2009)	14.92	7.02	30.14	5.56
	Notre méthode	8.37	3.25	16.31	3.52

TABLE 6.1 – **Comparaison de notre approche de création de base d'amers avec l'approche de Lothe et al. (2009) : mesure de l'erreur de re-projection entre les coins labellisés et les coins des modèles de bâtiments re-projetés.**

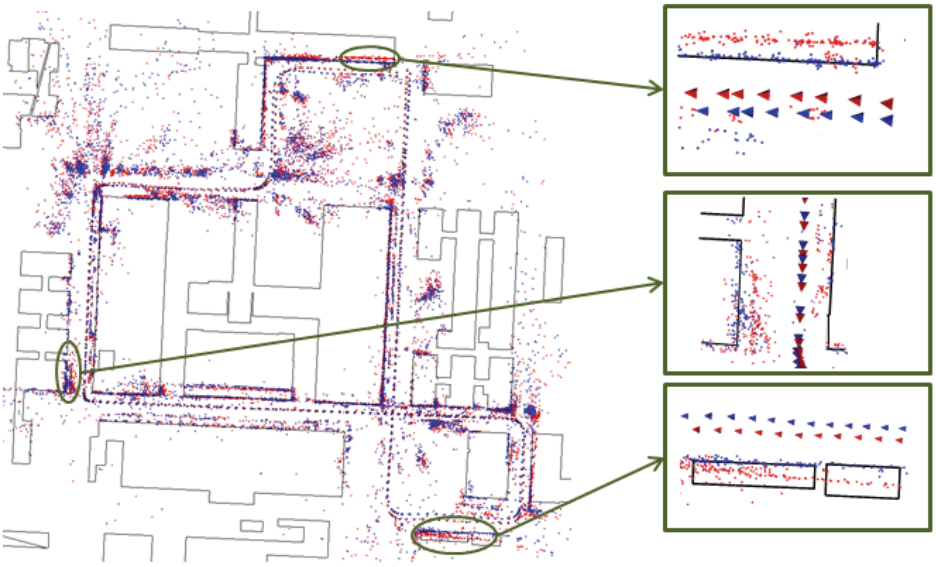
2. Plusieurs clics sont réalisés pour chaque coin. Ensuite, la moyenne de ces clics est considérée comme la vérité terrain

6.5 Conclusion et perspectives

Dans ce chapitre, nous avons proposé une solution permettant d'obtenir, hors ligne, une reconstruction SLAM géo-référencée précise en exploitant les contraintes issues des données d'un GPS standard, d'un MET et des modèles 3D des bâtiments. Afin de pouvoir fusionner les contraintes apportées par ces données sans être sujet à des problèmes de convergence, notre solution ne repose pas sur une utilisation simultanée de l'ensemble des contraintes mais sur une succession de raffinements non linéaires de la reconstruction, chaque raffinement exploitant une partie de ces contraintes. Le choix des contraintes et des degrés de liberté optimisés à chaque étape a été établi de manière à débuter avec l'ajustement de faisceaux assurant la localisation la moins précise pour terminer l'ajustement de faisceaux offrant la localisation la plus précise. La robustesse de cette solution a été démontrée sur des séquences réelles de plusieurs kilomètres en conditions normales de circulation. L'amélioration de la qualité de la reconstruction a été également évaluée et comparée à la méthode de Lothe et al. (2009). En termes de déploiement, cette solution présente l'avantage de reposer uniquement sur des capteurs (caméra, GPS) standards et bas coût, ainsi que des données (modèle d'élévation de terrain, modèles 3D de bâtiments) disponibles dans les systèmes d'information géographiques actuels (IGN, Openstreetmap,...). Le raffinement ne s'appuyant pas sur une fermeture de boucle, la méthode n'impose aucune contrainte sur la nature de la trajectoire du véhicule. En termes de temps de traitement, celui-ci est relativement réduit puisqu'une partie du traitement est réalisée en ligne alors que la partie hors ligne ne prend que quelques minutes pour une séquences de plusieurs kilomètres sur un simple CPU. En revanche, la méthode proposée dans ce chapitre reste majoritairement hors-ligne. Malgré sa précision, elle se limite donc à géo-référencer a posteriori le véhicule. Comme nous le montrerons dans le chapitre 8, cette méthode présente néanmoins un intérêt pour le déploiement en ligne. Dans le chapitre suivant, nous proposons une autre stratégie visant à combiner les informations des différents capteurs et données en ligne.

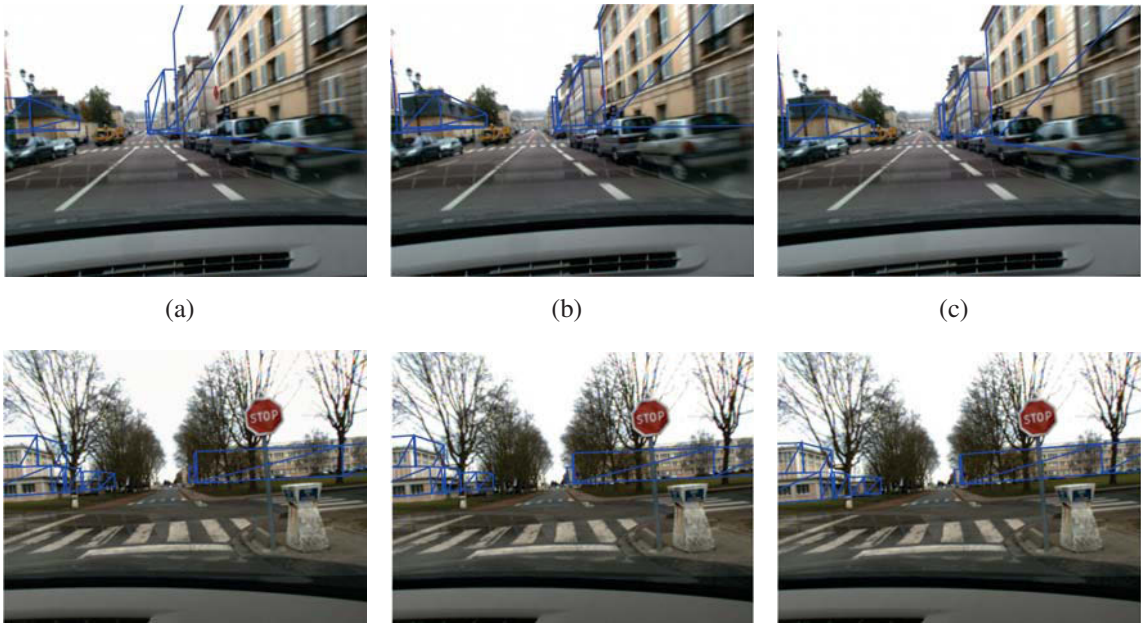


(a)



(b)

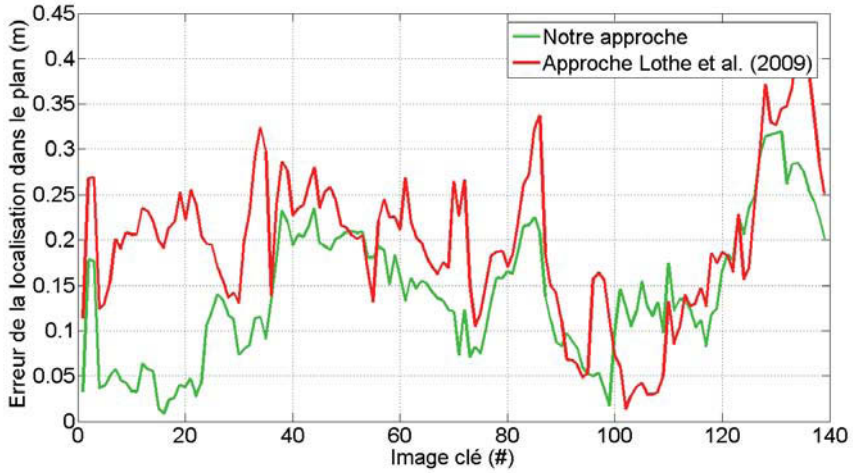
FIGURE 6.4 – **Validation de notre processus de création de bases d'amers géo-référencées**. Vues de dessus des bases de données obtenues dans les quartiers de Versailles et Saclay après la première étape (SLAM contraint aux données GPS et la contrainte douce en altitude) (rouge) et à l'issue de notre processus d'optimisation complet (section 6.3.2) (bleu).



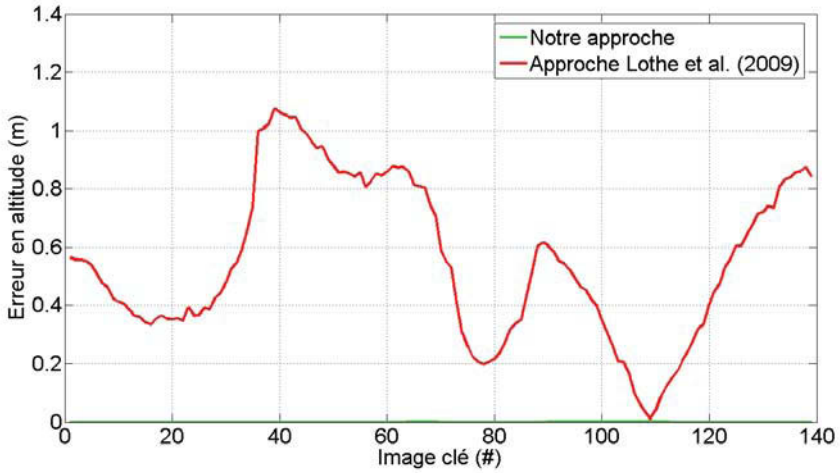
(a) (b) (c)

(d) (e) (f)

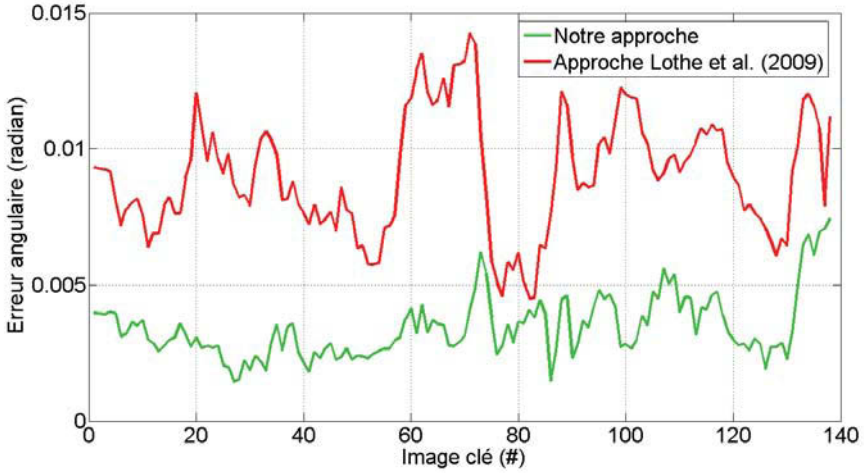
FIGURE 6.5 – **Exemples de re-projection des modèles des bâtiments.** Les résultats d'une des séquences de Versailles sont présentés au niveau de la ligne de dessus. Les résultats de la séquence de Saclay sont présentés dans la ligne de dessous. (a) et (d) présentent les résultats après la première étape de la création de la base de données (SLAM contraint aux données GPS et la contrainte douce en altitude , section 5.3) ; (b) et (e) sont les résultats après le premier ajustement de faisceaux global (section 6.3.2.1), (c) et (f) présentent les résultats finaux obtenus après le deuxième ajustement de faisceaux global (section 6.3.2.2).



(a)

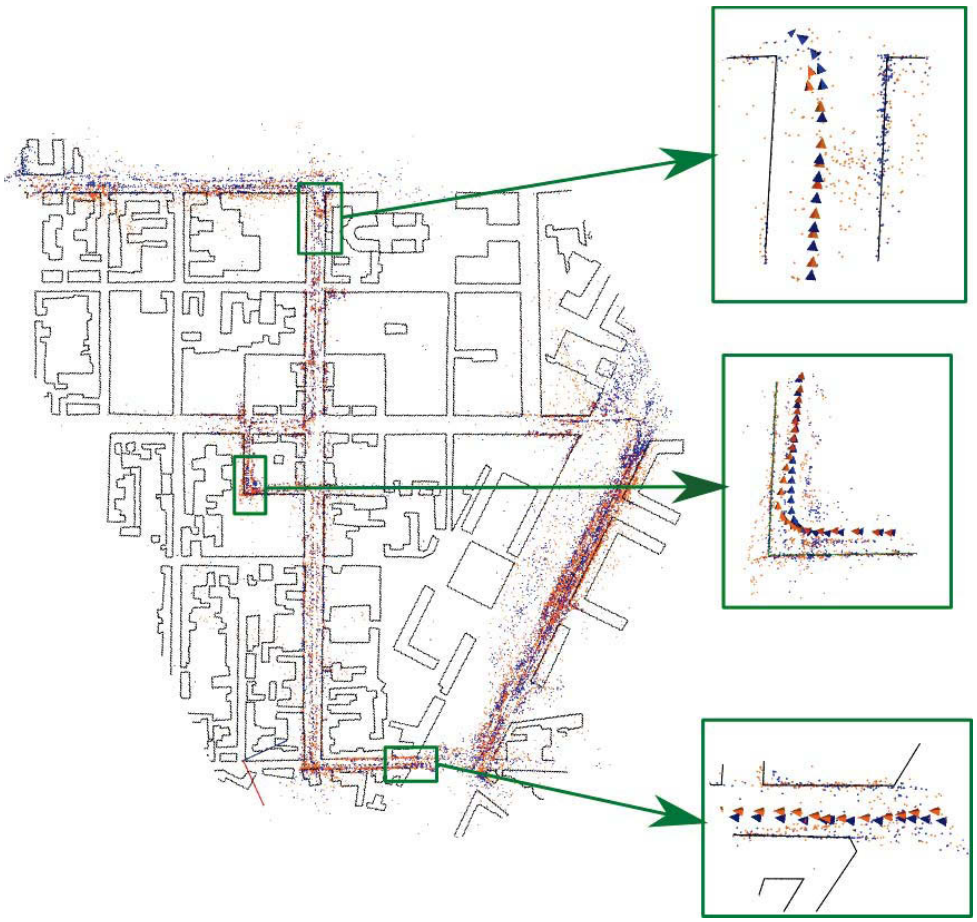


(b)

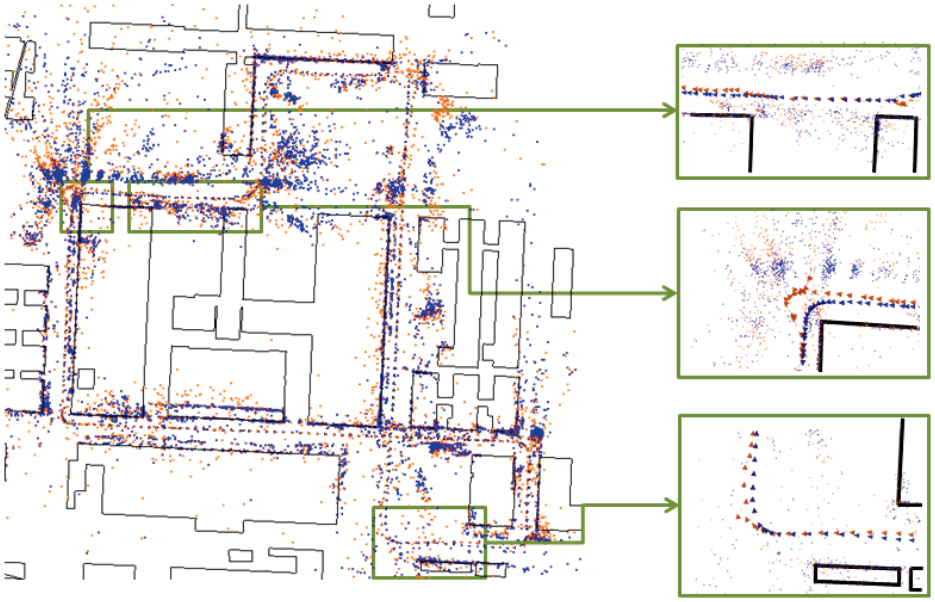


(c)

FIGURE 6.6 – **Comparaison avec la méthode de Lothe et al. (2009) en utilisant la séquence de synthèse**. (a), (b) et (c) représentent respectivement l'évolution de l'erreur de la localisation *dans le plan*, en altitude et en orientation. Les résultats obtenus avec la méthode de Lothe et al. (2009) sont tracés en rouge tandis que les résultats obtenus avec notre algorithme sont représentés en vert.



(a)



(b)

FIGURE 6.7 – **Comparaison avec la méthode de Lothe et al. (2009) sur des séquences réelles.** Vue de dessus des bases de données obtenues par notre méthode (en bleu) et celles obtenues par la méthode proposée par Lothe et al. (2009) (en orange). (a) représente les résultats obtenus pour la deuxième séquence de Versailles. (b) représente les résultats obtenus pour la séquence de Saclay.

FIGURE 6.8 – **Exemples de re-projection des modèles des bâtiments sur des images extraites de la première séquence de Versailles.** La colonne de gauche représente le résultat de la méthode proposée. La colonne de droite représente le résultat obtenu avec la méthode introduite par Lothe et al. (2009). Les erreurs de re-projections associées sont : (a) 8.23 pixels, (b) 37.66 pixels, (c) 6.50 pixels, (d) 13.95 pixels (e) 6.24 pixels and (f) 7.19 pixels.

CHAPITRE 7

Fusion en ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Correction du biais du GPS

Dans le chapitre précédent, nous avons proposé une solution hors ligne qui fusionne les contraintes multi-vues avec celles apportées par le GPS, le MET et les modèles 3D des bâtiments. Cette méthode permet de géo-référencer a posteriori la trajectoire de la caméra et la base d'amers associée. Dans ce chapitre, nous souhaitons proposer une nouvelle stratégie permettant de fusionner en ligne les contraintes évoquées ci-dessus. En particulier, nous proposons d'utiliser la connaissance a priori des bâtiments et la reconstruction éparse fournie par le SLAM pour déterminer dynamiquement le biais affectant les données GPS au cours du temps.

Ces travaux sont soumis dans une conférence nationale et font également l'objet d'une soumission d'un brevet.

7.1 Positionnement et principe de notre approche

Tandis que le GPS garantit une bonne localisation *dans le plan* en zones rurales et péri-urbaines, son incertitude est de plus en plus notable en présence des bâtiments à cause du phénomène de canyon urbain (voir la localisation dans le quartier de Versailles illustrée dans la figure 5.7).

Comme nous l'avons détaillé dans la section 1.4.2.1, Chausse et al. (2005) ont démontré qu'il était raisonnable de modéliser l'erreur affectant les mesures du GPS par un biais local auquel s'ajoute un bruit de plus faible amplitude. Bien que notre solution de SLAM contraint aux données GPS et au MET (voir chapitre 5) assure une certaine robustesse face aux données aberrantes grâce à la contrainte d'inégalité, cette solution ne permet pas de pallier le problème du biais. En effet, contrairement aux données aberrantes, une incertitude sous forme d'un biais local n'entraine pas une augmentation significative de l'erreur de re-projection au cours de l'optimisation. Par conséquent, la fusion avec les données GPS biaisées est prise en compte, comme le montre la figure 3.9, d'où le manque de précision dans les milieux urbains qui se manifeste par le décalage des points 3D reconstruits par rapport aux modèles des bâtiments. L'utilisation

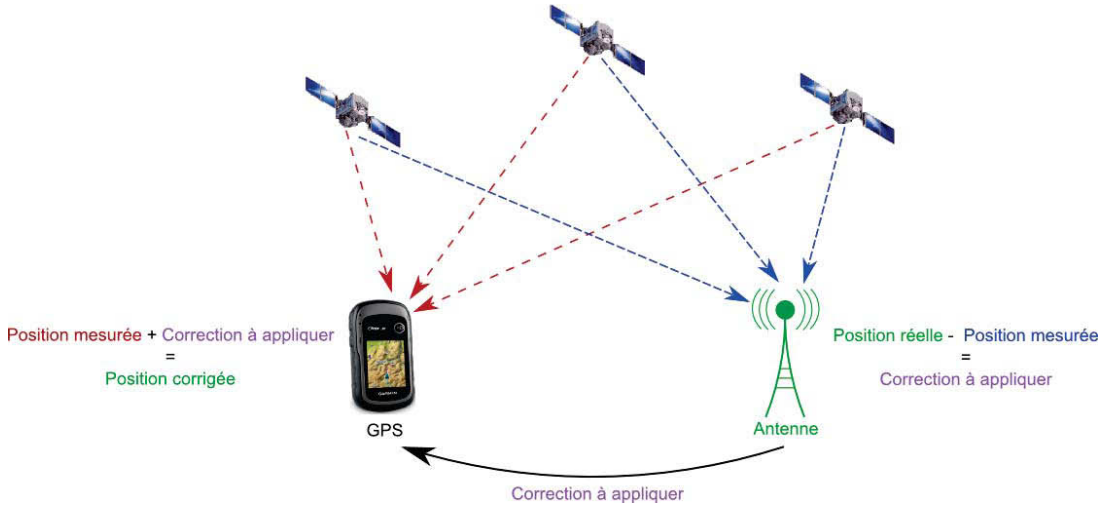


FIGURE 7.1 – **Principe de fonctionnement du GPS différentiel.**

d'un GPS plus précis, tel qu'un GPS différentiel, permettrait d'améliorer grandement la qualité de la localisation. En effet, afin de corriger l'incertitude de la localisation, ce type de GPS utilise un réseau de stations fixes de référence (des antennes géo-référencées) qui transmet au récepteur GPS l'écart entre les positions indiquées par les satellites et leurs positions réelles connues, comme le montre la figure 7.1. Toutefois, le GPS différentiel a des inconvénients. En effet, cette technologie dépend de la présence des antennes géo-référencées qui ne sont pas disponibles partout. Par ailleurs, à cause de son coût très élevé, elle ne représente pas un produit grand public que les utilisateurs des véhicules peuvent exploiter. Enfin, la correction estimée par cette technologie reste sujet de plusieurs erreurs telles que celles dues au décalage d'horloge et celles dues aux réflexions du signal sur les façades des bâtiments.

Récemment, certaines études, notamment celle proposée par Chausse et al. (2005) ou celle introduite par Kichun et al. (2013), se sont inspirées du principe du GPS différentiel afin de corriger le biais du GPS standard en remplaçant le réseaux d'antennes par des informations géo-référencées extraites des modèles numériques de la route à savoir les marquages au sol. Par exemple, Kichun et al. (2013) détectent les lignes blanches délimitant les voies de la route à partir du flux vidéo enregistré par une caméra embarquée dans le véhicule et les mettent en correspondance avec les éléments des modèles numériques de la route afin estimer le biais latéral du GPS. Les passages piétons sont, quant à eux, exploités pour calculer le biais selon la direction du déplacement du véhicule. Cependant, en zone urbaine, de tels marquages ne sont pas toujours présents (*e.g.* ruelles à sens unique), ni toujours visibles (*e.g.* masquage par les voitures se trouvant sur la chaussée, passage piéton représenté par des pavages, ou encore usure des marquages). Il est alors peu probable que le système en question parvienne à détecter de telles informations visuelles afin de ré-estimer le biais à chaque fois que celui-ci change. Aussi, notons que la faible fréquence des passages piéton rend l'estimation du biais particulièrement sensible à des données aberrantes (*e.g.* passage piéton déplacé). Enfin, ce type d'information est aujourd'hui peu répandu dans les SIG, ce qui rend ce type d'approche difficile à déployer. Pour pallier ce problème, nous proposons dans ce chapitre de reprendre le principe de GPS différentiel basé sur des modèles SIG en lui apportant certaines modifications. Nos contributions s'articulent autour des deux principaux axes suivants :

- ▷ Estimer le biais du GPS à partir de la reconstruction SLAM et les modèles 3D des bâtiments qui sont largement disponibles et plus visibles en milieux urbains (section

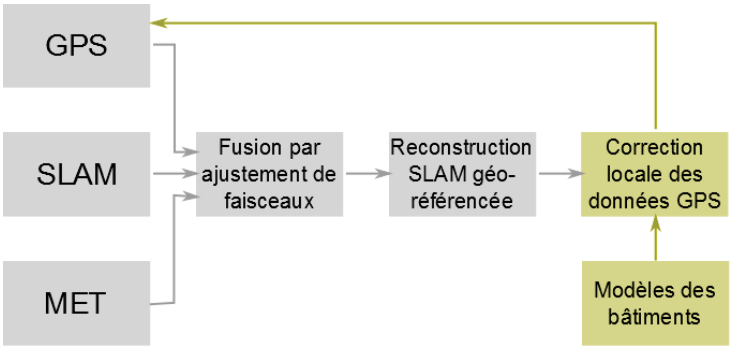


FIGURE 7.2 – **Principe général de l'approche proposée.** En gris la fusion du SLAM visuel avec les données GPS et le MET introduit dans la section 5.3. En vert, le module de correction des données GPS proposé dans ce chapitre afin d'améliorer la précision de la localisation.

7.2) ;

▷ Considérer ce biais dans la fusion SLAM/GPS afin d'assurer une meilleure localisation (section 7.3).

Le schéma général de notre approche est présentée dans la figure 7.2.

7.2 GPS différentiel basé sur les modèles 3D des bâtiments

Dans ce qui suit nous considérons que la localisation en ligne est assurée à travers l'algorithme du SLAM intégrant la contrainte aux données GPS et *la contrainte douce en altitude* détaillée dans la section 5.3. Après avoir présenté globalement l'approche utilisée pour estimer le biais du GPS dans la section 7.2.1, les étapes de notre méthode seront détaillées dans les sections 7.2.2 et 7.2.3.

7.2.1 Présentation générale

Dans le but d'améliorer la localisation à l'issue de la fusion du SLAM avec le GPS et le MET, nous proposons dans ce chapitre d'exploiter les contraintes locales fournies par les modèles 3D des bâtiments pour corriger les imprécisions des données GPS dans les milieux urbains. Pour ceci, nous nous basons sur l'observation suivante. Le biais du GPS n'est pas directement observable, contrairement à l'erreur de la reconstruction SLAM engendrée par ce biais qui peut se manifester via le décalage entre le nuage de points 3D reconstruit et les modèles des bâtiments. Pour estimer ce décalage, notre solution repose sur l'hypothèse suivante : l'erreur affectant localement la reconstruction SLAM après la fusion correspond à une transformation rigide dans le plan du sol. Cette hypothèse s'avère être une approximation suffisante comme démontré dans les expérimentations (section 7.4). Or, comme les poses de la camera ne suivent pas exactement les données GPS, en raison de la contrainte d'inégalité introduite dans l'ajustement de faisceaux, une deuxième étape est nécessaire pour corriger le biais du GPS. Ainsi, pour établir cette correction, nous procédons comme suit :

▷ **Estimation de l'erreur *dans le plan* de la fusion du SLAM avec le GPS et le MET**. La première étape de notre processus consiste à estimer les positions *dans le plan* de la caméra que nous souhaitons obtenir après la fusion par l'ajustement de faisceaux contraint (voir section 7.2.2). Pour cela le nuage de points 3D reconstruit par le SLAM est comparé avec les modèles des bâtiments ;

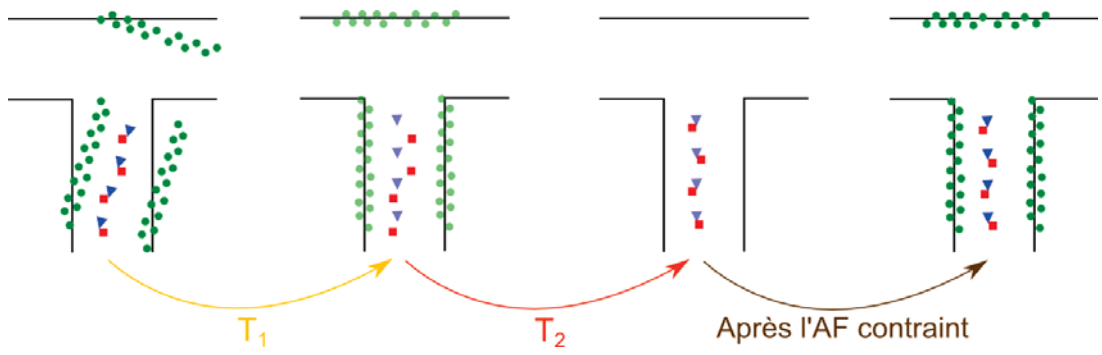


FIGURE 7.3 – **Schéma global de la solution proposée.** Les points verts représentent l'ensemble des points 3D associés aux modèles des bâtiments. Les différentes poses de la caméra sont représentées par les triangles bleus. Les positions de la caméra et les points 3D que nous souhaitons obtenir sont représentés en transparent. Les positions réelles sont quant à elles opaques. Les positions GPS sont modélisées par les carrés rouges. L'image de gauche représente la reconstruction initiale obtenue par SLAM contraint au GPS et au MET (*contrainte douce en altitude*). T_1 est la transformation rigide permettant d'estimer les positions *dans le plan* des images clés que nous souhaitons obtenir après la fusion du SLAM avec les données GPS corrigées. T_2 représente la correction à appliquer aux données GPS pour tendre après fusion vers les positions *dans le plan* prédites. Le nuage de points 3D résultant sera alors aligné avec les modèles 3D des bâtiments. L'image de droite schématise le résultat obtenu après l'ajustement de faisceaux contraint aux données GPS corrigées et au MET.

- ▷ **Correction du biais du GPS.** La deuxième étape de notre processus consiste à estimer la correction *dans le plan* à appliquer localement aux données GPS afin d'atteindre, après fusion, les positions *dans le plan* prédites de la caméra et ainsi obtenir un nuage de points 3D recalé sur les modèles des bâtiments (voir section 7.2.3).

Ci-dessous, nous détaillerons chacune de ces deux étapes.

7.2.2 Estimation de l'erreur *dans le plan* de la fusion du SLAM avec le GPS et le MET

Comme nous l'avons mentionné, à l'issue de la fusion du SLAM avec le GPS et le MET, le nuage de points reconstruit n'est pas parfaitement aligné avec les modèles des bâtiments principalement à cause du biais du GPS. Par conséquent, la première étape de notre processus cherche à corriger la reconstruction SLAM à l'issue de la fusion afin d'obtenir les positions *dans le plan* de la camera les plus précises possibles. Pour des raisons de clarté, nous nous placerons, dans un premier temps (section 7.2.2.1), dans le cas simple où la scène observée présente des façades couvrant l'ensemble des orientations. En d'autres termes, les contraintes fournies par les modèles 3D des bâtiments permettent d'estimer avec précision le biais de la fusion. Ensuite, nous présenterons, dans la section 7.2.2.2, les modifications à apporter à notre approche afin de gérer le cas de figure où le biais de la fusion est mal contraint par les modèles 3D des bâtiments.

7.2.2.1 Estimation de l'erreur *dans le plan* de la fusion dans le cas simple

Dans cette section, nous souhaitons corriger l'erreur *dans le plan* de la reconstruction SLAM afin d'obtenir les positions *dans le plan* de la caméra les plus précises possibles. Pour

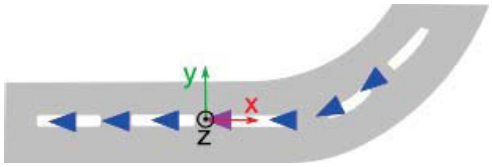


FIGURE 7.4 – **Repère utilisé pour le calcul de la transformation T_1.** Les différentes poses de la caméra sont représentées par les triangles bleus. Le triangle violet représente la pose de la caméra définissant l'origine du repère considéré.

ceci, nous faisons l'hypothèse que la correction idéale est celle qui aligne au mieux le nuage de points avec les façades des bâtiments. Si cette correction peut être globalement complexe et non linéaire, elle peut être approximée localement par une transformation rigide *dans le plan* T_1, comme le schématise la figure 7.3. Nous choisissons alors d'adopter le principe de fenêtre glissante définie par les n dernières images clés. Un nombre n élevé implique plus de contraintes pour estimer la transformation T_1 et donc une correction plus robuste des données GPS. Toutefois, un retard dans la correction du biais du GPS peut se produire. De plus, ceci peut influencer la précision de la correction en augmentant le risque de modéliser plusieurs biais par un seul. Inversement si n est petit, la correction peut être mal estimée vu le manque de contraintes disponibles pour estimer T_1. Le choix de la largeur de la fenêtre n est donc un compromis entre robustesse et précision. Quoiqu'il en soit le nombre de caméra utilisé pour estimer T_1 est inférieur au nombre d'images clés utilisées pour réaliser la fusion du SLAM avec le GPS et le MET. Ainsi, appliquer localement la transformation T_1 à la reconstruction SLAM pourrait briser les contraintes multi-vues sur la fenêtre d'images clés nécessaires à la fusion. Par conséquent, la reconstruction SLAM localement corrigée par T_1 n'est pas conservée. Seules les positions *dans le plan* prédites de la caméra sont stockées temporairement pour estimer le biais du GPS.

Afin de garantir une meilleure estimation de T_1, seuls les points 3D associés aux modèles des bâtiments sont utilisés au cours de cette première étape. Dans la suite de ce chapitre, nous noterons $\{\mathcal{Q}_i\}_{i\in\mathcal{M}'}$ l'ensemble de points associés aux modèles des bâtiments et observés dans les n dernières images clés. Ces points sont obtenus grâce à la méthode de segmentation introduite dans la section 4.3.

Une fois la segmentation effectuée, la transformation rigide *dans le plan* T_1 peut être estimée. Pour ceci, nous nous plaçons dans un nouveau référentiel schématisé dans la figure 7.4 où :

- ▷ L'origine du repère est défini par le barycentre des n positions 3D de la caméra ;
- ▷ L'orientation du repère est l'orientation de la route la plus proche de l'origine.

Dans ce nouveau repère, la transformation T_1 est obtenue en minimisant la distance euclidienne séparant chaque point de l'ensemble $\{\mathcal{Q}_i\}_{i\in\mathcal{M}'}$ à son plan correspondant.

Ainsi, la transformation $\mathsf{T}_1 = \left(\mathcal{R}_1^\gamma|\mathbf{u}_1\right)$, composée par la rotation autour de l'axe de lacet $\mathcal{R}_1^\gamma$ et la translation *dans le plan* $\mathbf{u}_1$, est obtenue en minimisant la fonction suivante :

$$f_{\mathsf{T}_1} = \sum_{i\in\mathcal{M}} \rho\left(d\left(\mathbf{\Pi}^{h_i}, \mathsf{S}^{-1}\tilde{\mathcal{Q}}_i'\right), c_1\right), \tag{7.1}$$

où

$$\mathcal{Q}_i' = \mathcal{R}_1^\gamma \mathsf{S}\tilde{\mathcal{Q}}_i + \mathbf{u}_1, \tag{7.2}$$

avec h_i l'indice de la façade associée au point $\mathcal{Q}_i$ et $d(\mathbf{\Pi}, \tilde{\mathcal{Q}})$ est la distance euclidienne séparant le point $\tilde{\mathcal{Q}}$ de sa façade $\mathbf{\Pi}$. S est la matrice de passage (3×4) du repère monde au nouveau référentiel. Le M-estimateur Geman-McClure est utilisé pour garantir plus de robustesse face aux données aberrantes et aux mauvaises associations point/plan.

7.2.2.2 Gestions des degrés de liberté mal contraints

Malheureusement, les contraintes apportées par les modèles 3D des bâtiments ne sont pas toujours suffisantes pour estimer avec précision la transformation recherchée. Par exemple, lorsque le véhicule circule dans une allée longue, l'ensemble des façades observées sont parallèles à l'axe de la route. Dans ce cas, certains degrés de libertés de la transformation T_1 ne seront pas ou sont mal contraints (*e.g.* la translation le long de l'axe de la route). Pour éviter de dégrader l'estimation du biais de la fusion à cause des degrés de liberté non contraints, il est important d'analyser les informations géométriques apportées par les modèles 3D des bâtiments. Selon les contraintes disponibles, il est possible de déterminer le nombre de degrés de liberté à estimer pour la transformation T_1. En effet, comme l'explique la figure 7.5, si les points 3D considérés $\{\mathcal{Q}_i\}_{i \in \mathcal{M}'}$ sont associés uniformément à des plans orthogonaux et latéraux par rapport au point de vue de la dernière caméra alors la transformation recherchée aura trois degrés de liberté : un pour la rotation $\mathcal{R}_1^\gamma$ autour de la normale à la route (*i.e.* angle lacet) et deux degrés de liberté pour la translation *dans le plan* de la route $\mathbf{u_1} = ((\mathbf{u_1})_x, (\mathbf{u_1})_y, 0)^T$. Dans le cas contraire, la transformation en question n'aura que deux degrés de liberté : l'angle lacet et un seul paramètre pour la translation soit $(\mathbf{u_1})_x$ soit $(\mathbf{u_1})_y$.

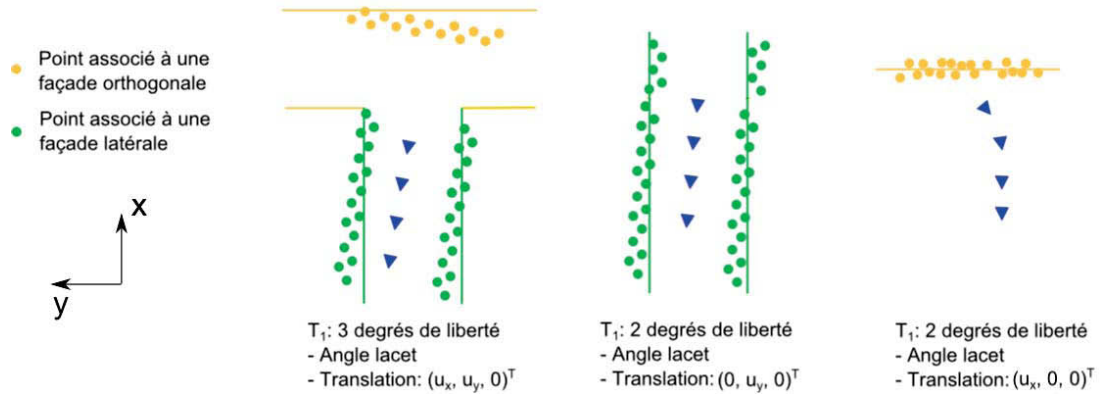


FIGURE 7.5 – **Nombre de degrés de liberté de la transformation rigide T_1 en fonction des contraintes disponibles.** Dans la figure à gauche, les points 3D sont uniformément associés aux façades latérales et celle orthogonale par rapport à la direction de la caméra. Dans ce cas, la transformation T_1 possédera trois degrés de liberté. Dans les deux autres cas, seuls deux degrés de liberté sont estimés.

7.2.3 Correction du biais du GPS

La transformation *dans le plan* T_1 décrite ci-dessus permet de fournir une estimation des positions *dans le plan* des n dernières images clés que nous souhaitons obtenir après la fusion une fois les données GPS corrigées. A partir de cette prédiction, il est possible d'estimer la

correction à appliquer aux n dernières données GPS.

Une idée naïve consiste à appliquer directement la transformation T_1 aux données GPS. Même si ceci peut sembler juste dans certains cas particuliers, l'application de T_1 s'avère être moins adaptée dans le cas général. En effet, cette transformation est estimée à partir du nuage de points issu de la fusion du SLAM avec le GPS et le MET. Or, en raison de l'utilisation de la contrainte d'inégalité et à cause de la variation du biais, le résultat de la fusion ne suit pas exactement les données GPS utilisées. En d'autres termes, certaines données GPS sont ignorées lors de la fusion. Par conséquent, dans ce cas de figure, la transformation T_1 ne peut pas être appliquée pour corriger les données en question.

La deuxième idée intuitive pour corriger le biais affectant les données GPS est de translater *dans le plan* chaque données GPS vers la position *dans le plan* prédite de l'image clé correspondante. Toutefois cette solution n'est pas bonne. En effet, à cause de la contrainte d'inégalité, certaines poses de caméra se trouvent mal contraintes après l'ajustement de faisceaux. Par conséquent, certaines images clés peuvent avoir leur position *dans le plan* mal estimée à l'issue de la fusion. Il en est de même pour les positions *dans le plan* prédites puisque T_1 est une transformation rigide.

Pour éviter ces deux problèmes, nous choisissons de corriger les données GPS à travers une transformation rigide en suivant le même principe expliqué dans la section 7.2.2.1. Nous cherchons, donc, la transformation rigide *dans le plan* $\mathsf{T}_2 = \left(\mathcal{R}_2^\gamma | \mathbf{u}_2\right)$ qui, appliquée aux données GPS brutes, permet, à l'issue de l'ajustement de faisceaux contraint, d'avoir un nuage de points aligné avec les modèles des bâtiments. Si nous partons de l'hypothèse que l'estimation des positions *dans le plan* prédites de la caméra est bonne, alors T_2 a pour but d'aligner les données GPS avec les positions *dans le plan* en question. Ainsi, T_2 est obtenue en minimisant les distances euclidiennes séparant les données GPS et les positions *dans le plan* prédites de la caméra comme le montre la figure 7.3.

Pour calculer la correction recherchée, le référentiel considéré pour le calcul de T_1 est utilisé. Ainsi, la fonction de coût résultante est donc donnée par :

$$f_{\mathsf{T}_2} = \sum_{j=N-n}^{N-1} \rho\left(\|\mathcal{R}_1^\gamma \mathsf{S}\tilde{\mathbf{t}}'_j + \mathbf{u}_1 - \mathcal{R}_2^\gamma \mathsf{S}\tilde{\mathbf{v}}_j + \mathbf{u}_2\|, c_2\right), \tag{7.3}$$

avec $\tilde{\mathbf{t}}'_j = ((\mathbf{t}_j)_x, (\mathbf{t}_j)_y, 0, 1)^T$ et $\tilde{\mathbf{v}}_j = (x_j^{gps}, y_j^{gps}, 0, 1)^T$. Comme précédemment, le M-estimateur Geman-McClure est utilisé pour garantir plus de robustesse face aux données aberrantes.

Contrairement à la transformation T_1 dont le nombre de degrés de liberté dépend des contraintes disponibles, les trois degrés de liberté *dans le plan* (*i.e.* l'angle lacet, la translation selon l'axe x et la translation selon l'axe y) doivent être estimés pour la correction T_2 étant donné que les contraintes apportées par les positions *dans le plan* prédites de la caméra sont suffisantes.

Cette transformation permet alors de corriger le biais local des n dernières données GPS. En ce qui concerne la correction de la prochaine donnée GPS et en l'absence de la connaissance a priori d'un éventuel changement de biais, nous choisissons de conserver la même transformation T_2 préalablement estimée. Ainsi, si aucun changement n'est notable au niveau du biais, cette solution permet de corriger efficacement la prochaine mesure du GPS. Dans le cas

contraire, un délai de quelques images clés est nécessaire pour ré-estimer la nouvelle correction. Malgré ce retard, nous montrerons dans la section 7.4 que les mauvaises prédictions locales ont peu d'influence sur la localisation en ligne après la fusion. Ceci est assuré grâce à la contrainte d'inégalité utilisée dans l'ajustement de faisceaux contraint qui empêche les poses de la caméra de converger vers les mesures GPS si ces dernières dégradent significativement les contraintes multi-vues.

7.3 SLAM contraint au MET et aux données du GPS corrigées

Dans la section précédente, nous avons expliqué comment les biais du GPS sont corrigés en exploitant les modèles 3D des bâtiments. Une fois corrigées, les mesures GPS peuvent être introduites dans le processus SLAM à travers l'ajustement de faisceaux avec la contrainte d'inégalité détaillée dans la section 5.3. Après avoir expliqué comment le processus de correction de biais s'insère dans l'algorithme de SLAM contraint, nous expliquerons ci-dessous la stratégie adoptée pour l'estimation de la correction du biais GPS et nous résumerons les différentes étapes nécessaires pour la réalisation du SLAM contraint au MET et aux données du GPS différentiel basé sur les modèles 3D des bâtiments.

Intégration du processus de correction de biais dans l'algorithme SLAM L'intégration du processus de correction de biais dans le SLAM est réalisé à travers trois étapes : étape d'initialisation, une correction du biais des n dernières mesures GPS et enfin une correction de la prochaine donnée GPS. En effet, initialement (*i.e.* pour les n premières images clés estimées), le biais de GPS est supposé nul. Dans ce cas, le SLAM contraint aux données GPS brutes et au MET introduit dans la section 5.3 est réalisé. A partir de la $n^{ème}$ image clé et une fois l'ajustement de faisceaux contraint effectué, le processus de correction du biais de GPS, introduit dans la section précédente, est réalisé pour la première fois. A l'issue de cette étape, les n dernières données GPS utilisées se trouvent corrigées. A partir de ce moment et à chaque image clé, la nouvelle donnée GPS brute est également corrigée en lui appliquant la dernière correction estimée. Les données GPS ainsi corrigées participeront par la suite dans l'ajustement de faisceaux contraint.

Stratégie d'exploitation de la correction du biais GPS. Afin de garantir une correction optimale du biais du GPS, notre processus n'est appliqué que si les contraintes disponibles permettent de calculer la transformation rigide T_1. En d'autres termes, si le nombre de points $\{\mathcal{Q}^i\}_{i\in\mathcal{M}'}$ associés aux modèles des bâtiments et observés dans les n dernières images clés est très faible (*i.e.* inférieur à un seuil $s_1 = 10$), le calcul de T_1 n'est plus fiable. Dans ce cas nous conservons la dernière correction estimée. Toutefois, si le nombre de points de l'ensemble $\mathcal{M}'$ reste faible pendant un certain temps, nous considérons que la caméra se situe dans un milieu péri-urbain où les données GPS sont assez précises. Par conséquent, la dernière correction estimée n'est plus propagée. Inversement en milieux urbains denses où le cardinal de l'ensemble $\mathcal{M}'$ est souvent élevé, nous remarquons que la correction fréquente (*i.e.* à chaque image clé) des données GPS peut entrainer une instabilité dans la localisation en ligne (*i.e. "jitter effect"*). Par conséquent, en plus des contraintes disponibles, notre processus de correction du GPS n'est exécuté que si la distance moyenne d_{moy} séparant les points 3D $\{\mathcal{Q}^i\}_{i\in\mathcal{M}'}$ de leurs plans correspondants est supérieure à un seuil s_2 (dans nos expérimentations $s_2 = 1$m). Dans le cas

contraire, la précédente correction estimée est conservée. La distance d_{moy} est obtenue comme suit :

$$d_{moy} = \frac{\sum_{i \in \mathcal{M}'} d\left(\mathbf{\Pi}^{h_i}, \mathcal{Q}^i\right)}{\text{card}(\mathcal{M}')} \tag{7.4}$$

avec $d\left(\mathbf{\Pi}, \mathcal{Q}\right)$ est la distance euclidienne séparant le point $\mathcal{Q}$ du plan $\mathbf{\Pi}$.

Processus d'optimisation. Les différentes étapes du processus d'optimisation associé au SLAM contraint au MET et aux données du GPS corrigées sont résumées dans l'algorithme 7. Initialement, les transformations T_1 et T_2 sont égale à $(\mathsf{I}_{3\times3}|\mathbf{0}_{3\times0})$. Dans ce cas les données GPS corrigées sont identiques aux données GPS brutes.

Appliquer la dernière transformation T_2 calculée à la nouvelle donnée GPS ;
Réaliser l'ajustement de faisceaux avec la contrainte d'inégalité détaillée dans la section 5.3 en exploitant les données GPS corrigées ;
Déterminer l'ensemble de points $\{\mathcal{Q}_i\}_{i \in \mathcal{M}'}$;
if $(card(\mathcal{M}') > s_1)$ *et* $(d_{moy} > s_2)$ **then**
    Déterminer les degrés de liberté de la transformation T_1 à partir du nuage de points $\{\mathcal{Q}_i\}_{i \in \mathcal{M}'}$;
    Estimer la transformation T_1 à partir du nuage de points $\{\mathcal{Q}_i\}_{i \in \mathcal{M}'}$;
    Déduire la reconstruction SLAM prédite ;
    Calculer la correction T_2 à partir de la reconstruction prédite ;
    Corriger les n dernières données GPS ;
end

Algorithme 7 : Différentes étapes du processus d'optimisation associé au SLAM contraint au MET et aux données du GPS corrigées (Processus appliqué à chaque image clé).

7.4 Évaluation expérimentale

Dans la suite de cette partie, nous nous intéresserons à la précision de la localisation a posteriori obtenue par le SLAM contraint aux données GPS corrigées et au MET, c'est-à-dire la précision de la reconstruction SLAM obtenue une fois que toute la séquence est traitée. En d'autres termes, à la fin de la séquence, chaque caméra clé est optimisée N fois, avec N est le nombre de caméras clés considéré dans l'ajustement de faisceaux contraint utilisé. L'évaluation de la localisation instantanée, nécessaire aux applications de Réalité Augmentée, sera présentée dans le chapitre 8.

Plusieurs expériences vont être menées sur des données de synthèse et des séquences réelles. Les données de synthèse, grâce à leur vérité terrain, vont permettre d'évaluer la précision de la correction du biais de GPS et de la localisation issue du SLAM contraint aux données GPS corrigées et au MET, de manière quantitative. Dans le cas réel, ne disposant pas d'une vérité terrain, nous proposons d'évaluer cette précision en mesurant l'erreur de re-projection des modèles des bâtiments dans certaines images extraites du flux vidéo.

Dans nos expérimentations, nous fixons le paramètre n à 10. Cette valeur a été estimée empiriquement. En effet, expérimentalement nous avons remarqué que cette valeur assure un bon compromis entre précision et robustesse.

7.4.1 Évaluation sur les données de synthèse

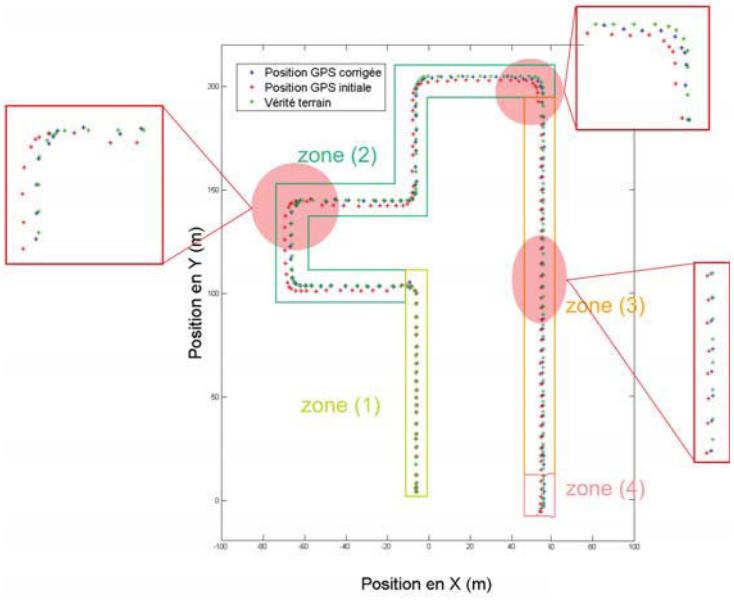
Dans ce qui suit, après avoir présenté le protocole expérimental et la séquence utilisée, nous présenterons les résultats obtenus sur les données de synthèse.

Protocole expérimental et séquence utilisée. Pour établir notre évaluation, nous comparons les positions *dans le plan* des données GPS avant et après la correction du biais par rapport à la vérité terrain. Nous analyserons également l'évolution de la localisation après la fusion du SLAM avec les données GPS corrigées. Plus de détails sur le calcul de l'erreur mesurée sont disponibles dans la section 4.4.1.

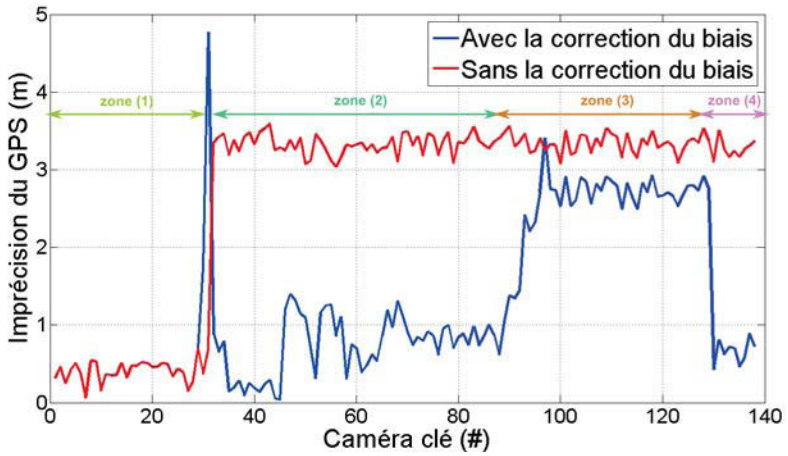
L'évaluation est établie en utilisant la séquence de synthèse décrite dans la section 6.4.1. Pour simuler les données GPS, les positions *dans le plan* de la caméra, fournies par la vérité terrain, sont perturbées en ajoutant un biais constant par morceaux d'amplitude maximale de 3m auquel est additionné un bruit Gaussien d'amplitude 0.5m. La direction de ce biais change après chaque virage comme le montre la figure 7.6(a).

Résultats. Comme le montre la figure 7.6(a) et le confirment les courbes de la figure 7.6(b), notre méthode permet de réduire notablement les imprécisions du GPS dont la valeur médiane passe de 3.2812m à 0.8527m après correction. Nous remarquons également que le résultat de la correction est fortement dépendant des contraintes disponibles, de la direction et de l'amplitude de la perturbation. En effet, avant le premier virage (la zone (1) de la figure 7.6(b)), seul le bruit gaussien de 0.5m est appliqué. Dans cette zone, aucune correction n'est appliquée. En effet, la contrainte d'inégalité permet de pallier ces imprécisions. Le nuage de points reconstruit est donc aligné avec les modèles des bâtiments. Ceci explique également l'absence de correction dans cette zone. Par ailleurs, quand le biais passe brutalement de 0m à 3m, une première correction est estimée. L'application de cette correction sur les données GPS brutes permet de réduire considérablement leur imprécision, comme nous pouvons l'observer dans la zone (2) des figures 7.6(a) et 7.6(b). Cette zone est caractérisée par la présence d'importantes contraintes liées aux nombreux virages traversés. Ainsi, la correction à appliquer est estimée avec plus de précision. Contrairement à la zone (2), la zone (3) représente une longue ligne droite. Dans cette zone, les données GPS sont principalement bruitées dans la direction de l'axe de la route. Même si les modèles des bâtiments permettent d'estimer la composante du biais selon l'axe orthogonal à la direction du déplacement, les contraintes disponibles sont insuffisantes pour estimer avec précision la composante du biais correspondent à l'axe de la route. Par conséquent, les données GPS corrigées sont moins précises que celles de la zone (2). Ces imprécisions sont par la suite corrigées dans la zone (4) grâce aux contraintes supplémentaires apportées par le virage. Malgré cette amélioration globale au niveau de la précision des données GPS corrigées, quelques imprécisions locales sont observées notamment la présence du pic de la courbe bleue entre la zone (1) et (2) dans la figure 7.6(b). En effet, vu que la même correction est appliquée sur les n dernières données, certaines mesures n'ayant pas le même biais peuvent être perturbées. Cependant, même si le principe de fenêtre glissante de taille n peut générer quelques données GPS moins précises là où le biais change, ces dernières sont considérées comme des données aberrantes, elles sont donc filtrées grâce à la contrainte d'inégalité de l'ajustement de faisceaux utilisé.

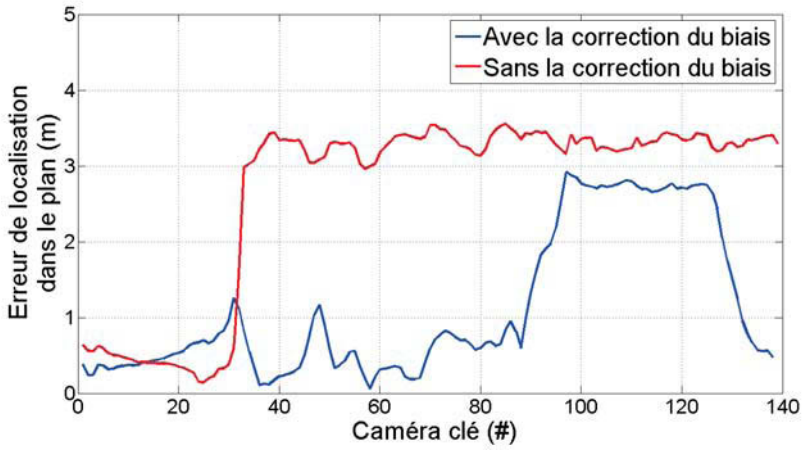
Concernant l'erreur de la localisation *dans le plan* obtenue après la fusion, les courbes représentées dans la figure 7.6(c) montrent que l'évolution de cette erreur est caractérisée par un comportement similaire à celui des données GPS (à l'exception des pics). Cette correction



(a)



(b)



(c)

FIGURE 7.6 – **Évaluation du processus de correction du biais du GPS.** (a) Vue de dessus des positions *dans le plan* des données GPS avant et après notre méthode de correction. (b) Évolution de l'imprécision des données GPS par rapport à la vérité terrain. (c) Évolution de l'erreur de la localisation *dans le plan* après la fusion.

permet une localisation plus précise. En effet, l'erreur médiane passe de 3.2635m à 0.6673m en appliquant notre processus de correction du biais du GPS. Comme nous l'avons mentionné ci-dessus, grâce à sa contrainte d'inégalité, le SLAM contraint au GPS et au MET permet de filtrer les données aberrantes notamment le pic observé dans la figure 7.6(b).

7.4.2 Évaluation sur les données réelles

L'objectif de cette expérimentation est d'évaluer la précision de la localisation de la fusion SLAM avec le MET et les données GPS corrigées grâce à notre processus de correction de biais.

7.4.2.1 Protocole expérimental et séquences utilisées

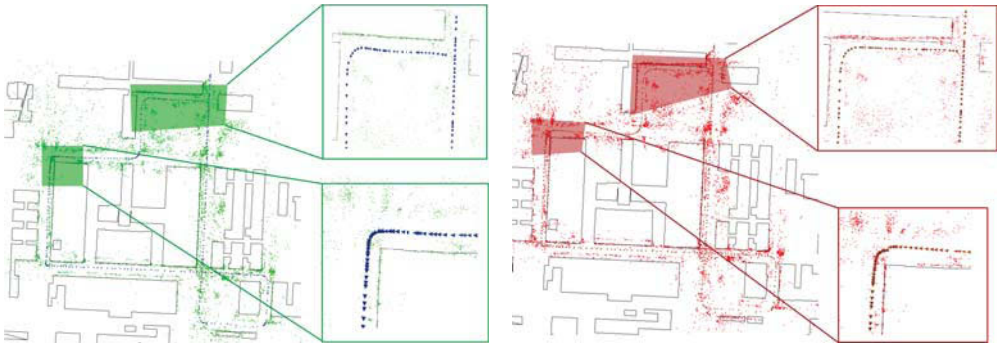
Étant donné que la vérité terrain n'est pas disponible pour les séquences de Versailles, notre évaluation quantitative se base sur la mesure de l'erreur de re-projection des modèles des bâtiments dans une sélection d'images des séquences utilisées. Nous utilisons alors la même technique introduite dans la section 6.4.3. Ainsi, nous labellisons manuellement les coins des bâtiments dans quelques images extraites des flux vidéo enregistrés dans le quartier de Versailles. Nous calculons par la suite l'erreur de re-projection entre les coins labellisés et la re-projection des coins des modèles des bâtiments. Pour évaluer qualitativement la précision de la localisation dans le plan, la vue de dessus de localisation obtenue par le SLAM contraint aux données GPS corrigées et au MET est comparée avec celle obtenue en exploitant les données GPS brutes. Par ailleurs, la précision de l'estimation des six degrés de liberté de la caméra est mise en évidence en re-projetant les modèles 3D des bâtiments sur des images extraites de la séquence utilisée. Dans la suite nous commençons par analyser les résultats dans les milieux péri-urbains. Les résultats obtenus en milieux urbains denses sont ensuite présentés.

En ce qui concerne les données réelles exploitées, nous utilisons les deux séquences enregistrées dans le quartier de Versailles (cas de milieu urbain dense) et la séquence enregistrée dans le quartier de Saclay (cas de milieu péri-urbain) décrites dans la section 6.4.1. Pour chacune des séquences utilisées, nous comparons la précision des localisations obtenues par un SLAM contraint aux données GPS et *la contrainte douce en altitude* (voir section 5.3) avec et sans correction du GPS.

7.4.2.2 Résultats

Cas d'un milieu péri-urbain : Cas de la séquence de Saclay Dans les milieux péri-urbains, les bâtiments sont moins présents. Par conséquent, les données GPS sont plus précises comme le montre la figure 7.7(b) où la reconstruction obtenue à l'issue de la fusion avec les données GPS brutes est assez précise. En effet, le nuage de points est globalement aligné avec les façades de bâtiments. Notre processus de correction du biais du GPS permet de préserver cette précision en l'absence des bâtiments. Si les contraintes fournies par les modèles des bâtiments sont suffisantes, notre processus permet également de pallier les éventuelles imprécisions du GPS (voir les zooms rouge de la figure 7.7(b) et les images de re-projection des modèles de bâtiments 7.8). En effet, appliquant la correction estimée dans ces zones, la précision s'est nettement améliorée (voir les zooms verts de la figure 7.7(a)) et les re-projections des modèles (figure 7.8).

Cas d'un milieu urbain dense. Dans les milieux urbains denses, le GPS est beaucoup moins précis comme le montrent les figures 7.9(b) et 7.10(b) où le nuage de points est souvent décalé



(a) (b)

FIGURE 7.7 – **Vue de dessus des localisations obtenues avec ou sans corrections du biais du GPS dans le quartier Saclay.** (a) Vue de dessus de la localisation obtenue avec un SLAM contraint aux données GPS corrigées et au MET. (b) Vue de dessus de la localisation obtenue avec un SLAM contraint aux données GPS brutes et au MET.

par rapport aux modèles des bâtiments. Dans la figure 7.12, nous représentons les positions de GPS avant et après la correction. Comme c'est le cas pour la séquence de synthèse, l'amélioration apportée par notre approche reste dépendante des contraintes disponibles. En effet, le long des lignes droites où peu de façades orthogonales sont observées, la correction estimée est limitée à la correction selon l'axe latéral de la route comme le montre les zooms vert de la figure 7.12. L'absence de la correction selon l'axe de déplacement de la caméra est compensée au niveau des virages qui apportent d'importantes contraintes pour l'estimation de la correction. Par ailleurs, appliquer la même correction sur les n dernières images clé peut entrainer quelques imprécisions locales observées principalement au niveau des zooms bleus de la figure 7.12. Ces imprécisions sont considérées comme des données aberrantes et donc seront filtrées, par la suite grâce, à la contrainte d'inégalité.

Concernant la localisation après la fusion du SLAM avec les données GPS corrigées, en comparant les figures 7.9(a) et 7.10(a) avec les figures 7.9(b) et 7.10(b), nous observons que le processus de correction du biais du GPS permet d'améliorer notablement la précision de la localisation *dans le plan*. En effet, après la correction des données GPS, le nuage de points reconstruit est aligné avec les empreintes des modèles des bâtiments. Ceci est également mis en évidence dans la figure 7.11 où les re-projections des modèles des bâtiments sont mieux recalés sur les façades des bâtiments après la correction. Les résultats quantitatifs du tableau 7.1 confirment ces observations. En effet, pour la première séquence de Versailles, la correction des données GPS a permis de réduire considérablement la moyenne des erreurs de re-projections obtenue en exploitant les mesures brutes du GPS. L'écart-type des erreurs mesurées a également baissé d'une façon notable en utilisant notre méthode de 11.6 pixels à 4.94 pixels. Par conséquent, en exploitant les informations fournies par les modèles des bâtiments, l'approche proposée dans ce chapitre permet de réduire l'incertitude du GPS dans les milieux urbains denses garantissant ainsi plus de précision à la localisation après la fusion. En évaluant quantitativement à travers la méthode de labellisation la reconstruction résultante, a posteriori, nous remarquons que la base d'amers géo-référencées obtenue par la présente méthode a une précision comparable à celle obtenue par notre solution de modélisation décrite dans le chapitre 6.

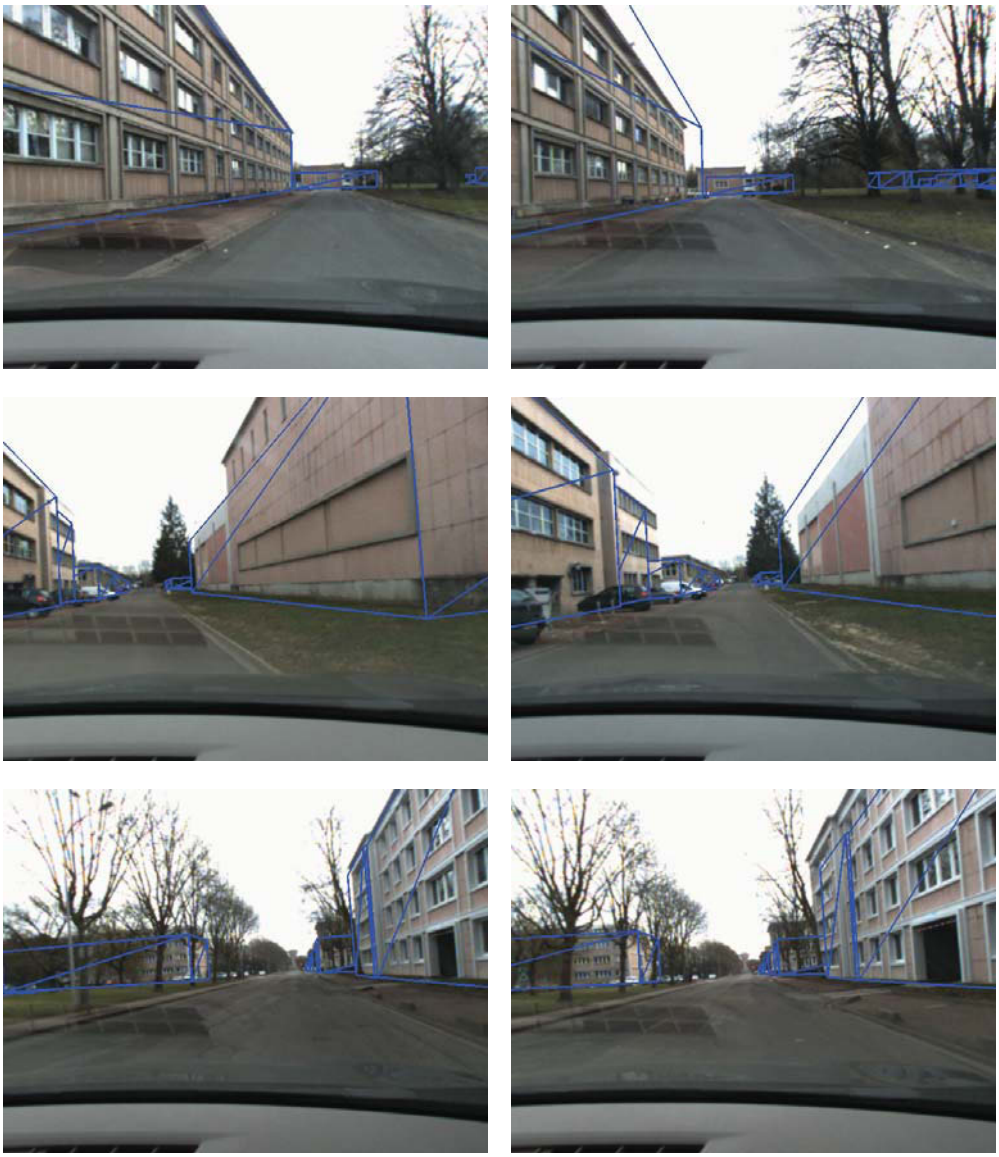
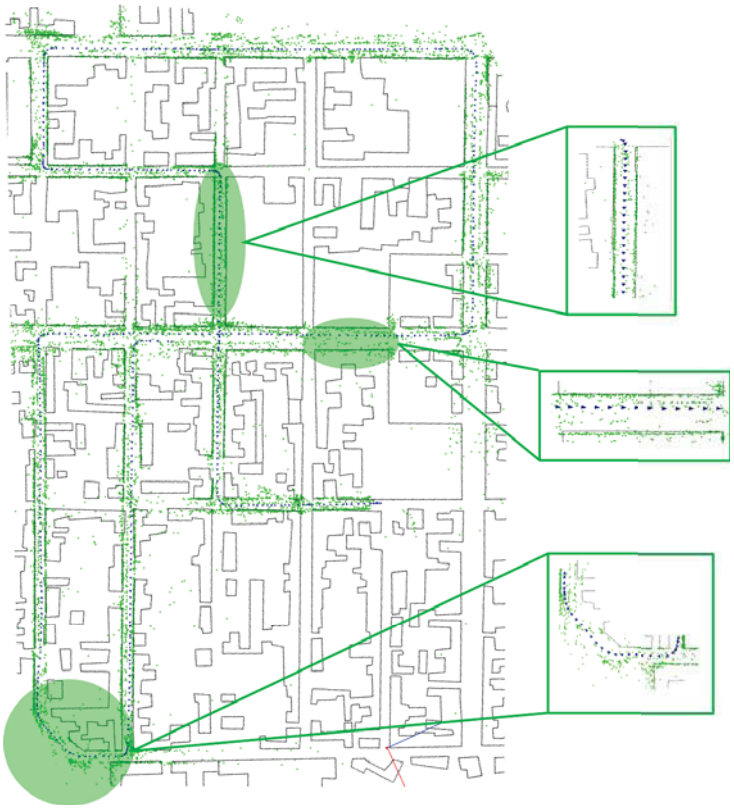
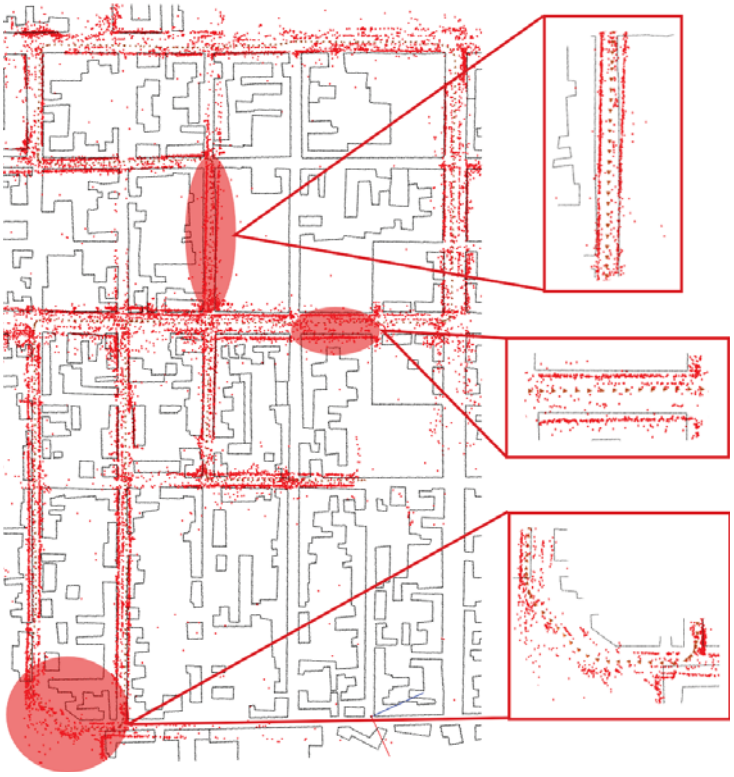


FIGURE 7.8 – **Exemples de re-projection des modèles des bâtiments sur des images extraites de la séquence de Saclay.** La colonne de gauche représente le résultat en utilisant notre processus de correction du biais. La colonne de droite représente le résultat obtenu par le SLAM contraint aux données GPS brutes et au MET.



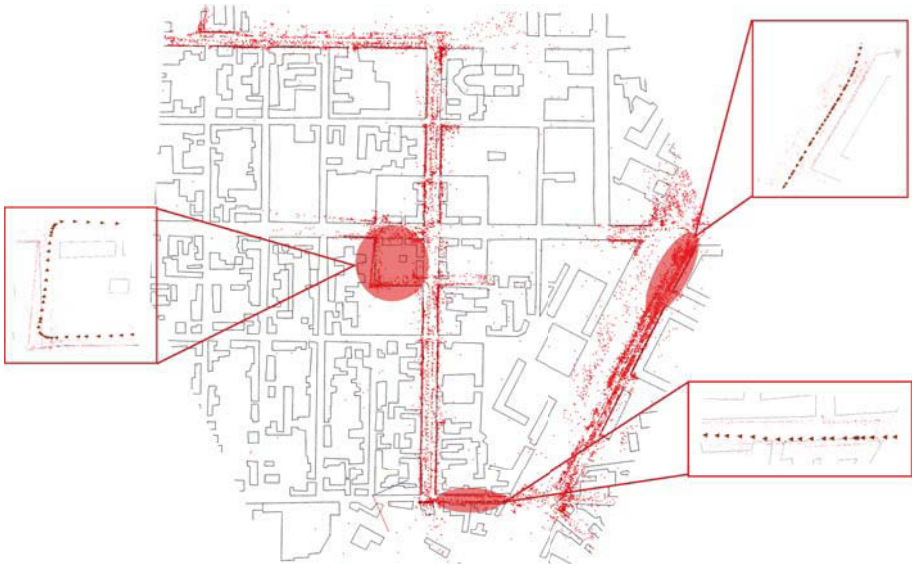
(a)



(b)

FIGURE 7.9 – **Vue de dessus des localisations obtenues avec ou sans corrections du biais du GPS dans le quartier de Versailles.** (a) Vue de dessus de la localisation obtenue avec un SLAM contraint aux données GPS corrigées et au MET. (b) Vue de dessus de la localisation obtenue avec un SLAM contraint aux données GPS brutes et au MET.

(a)



(b)

FIGURE 7.10 – **Vue de dessus des localisations obtenues avec ou sans correction du biais du GPS dans le quartier de Saclay.** (a) Vue de dessus de la localisation obtenue avec un SLAM contraint aux données GPS corrigées et au MET. (b) Vue de dessus de la localisation obtenue avec un SLAM contraint aux données GPS brutes et au MET.

(a) (b) (c) (d) (e) (f)

FIGURE 7.11 – **Exemples de re-projection des modèles des bâtiments sur des images extraites de la séquence de Versailles.** La colonne de gauche représente le résultat en utilisant notre processus de correction du biais. La colonne de droite représente le résultat obtenu par le SLAM contraint aux données GPS brutes et au MET. Les erreurs de re-projection associées : (a) 8.32 pixels, (b) 16.79 pixels, (c) 5.74 pixels, (d) 16.50 pixels, (e) 6.15 pixels et (f) 24.10 pixels.

		Erreur de re-projection (pixels)			
		Moyenne	Écart type	Max	Min
Versailles	Données brutes	14.97	11.60	44.51	4.16
	Données corrigées	8.92	4.94	18.35	4.01

TABLE 7.1 – **Précision des localisations obtenues avec ou sans la correction du biais du GPS dans le quartier de Versailles : mesure de l'erreur de re-projection entre les coins labellisés et les coins des modèles de bâtiments re-projetés.**

Limitation. Nous avons montré dans la section précédente que la correction des données GPS permet d'avoir une localisation plus précise dans les milieux urbains. Toutefois, l'amélioration apportée par notre approche reste dépendante des contraintes disponibles. Ainsi, certains retards de correction sont notés (principalement dans les lignes droites). Même si ces retards sont rattrapés a posteriori (au niveau des virages), ces derniers réduisent la précision de la localisation instantanée. Ceci sera détaillé dans le chapitre 8.

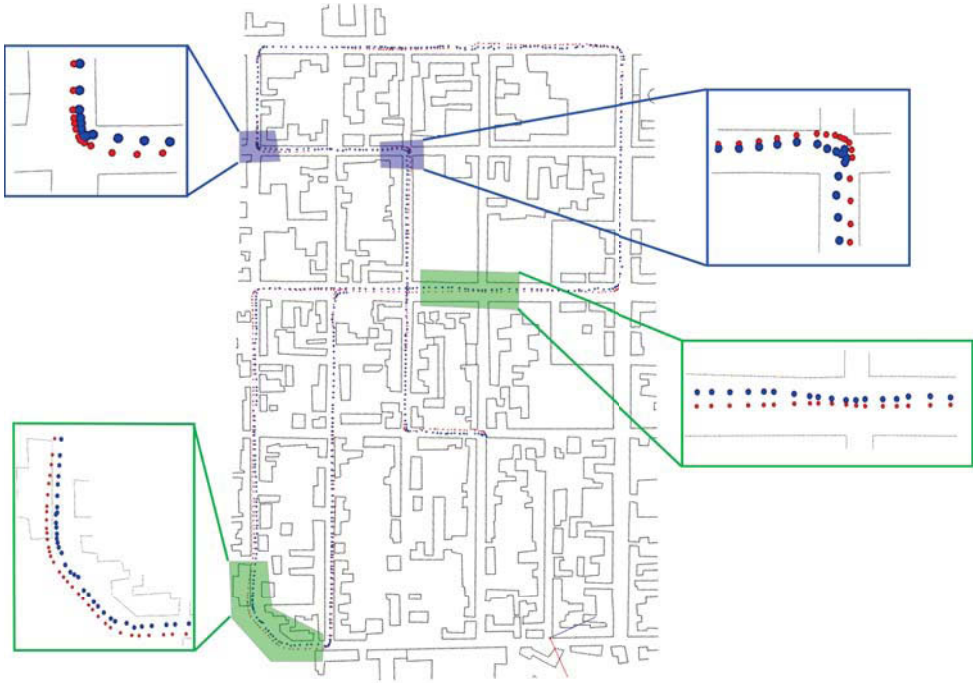


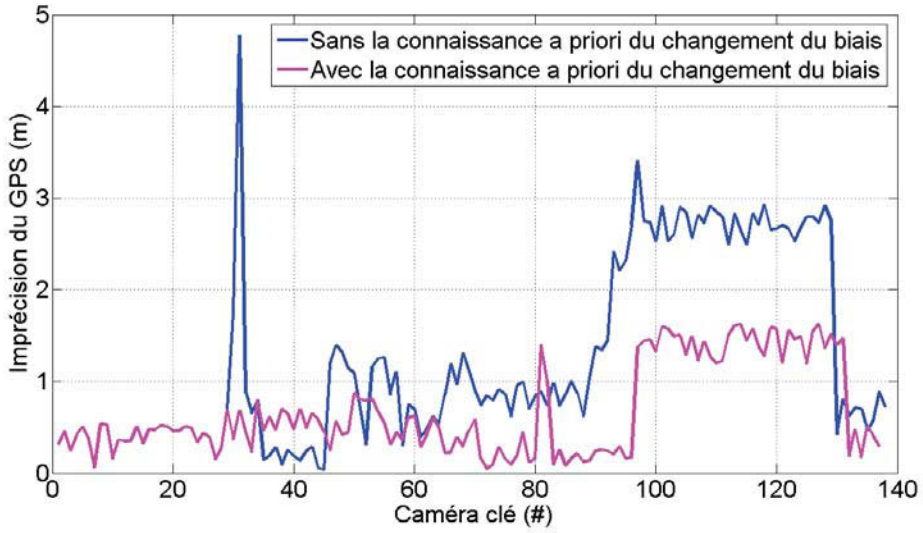
FIGURE 7.12 – **Vue de dessus des positions *dans le plan* des données GPS avant et après notre méthode de correction.** En rouge, les données GPS brutes. En bleu, les données GPS corrigées.

7.5 Conclusion et perspectives

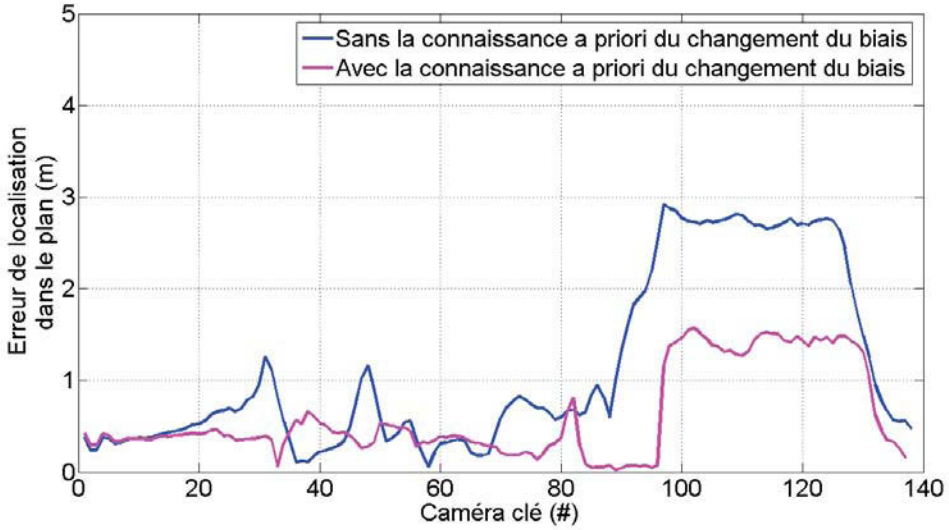
Dans ce chapitre, nous avons proposé une solution pour fusionner en ligne les contraintes multi-vues avec celles fournies par le GPS, le MET et les modèles des bâtiments. Tandis que le MET contraint les degrés de liberté *hors plan*, les données GPS permettent de contraindre la position *dans le plan* de la caméra. Les bâtiments quant à eux sont exploités pour corriger

localement les imprécisions du GPS, d'où la notion du GPS différentiel basé sur les modèles 3D des bâtiments. La correction apportée aux données GPS permet alors d'obtenir plus de précision au SLAM contraint au GPS et au MET dans les milieux urbains denses. Les expérimentations sur des données synthétiques et réelles confirment l'amélioration de la localisation dans le plan. Malgré ces résultats prometteurs, certaines limitations restent notables. Premièrement, la précision de la correction reste limitée aux contraintes disponibles notamment dans les lignes droites. Deuxièmement, un certain retard de l'estimation de la correction est notable quand le biais change.

Pour faire face au premier problème, il est possible d'enrichir la reconstruction SLAM en modifiant le critère de décision de création d'images clés. En effet, augmenter le nombre d'images clés créés dans certaines zones notamment, à proximité des croisements, permet de mieux construire les façades orthogonales et donc d'apporter plus de contraintes pour l'estimation de la transformation T_1. En ce qui concerne la deuxième limitation, une solution possible est d'estimer un même biais uniquement pour l'ensemble des données GPS affectées d'un biais identique. Pour ceci, il est important d'identifier les changements de biais. Certaines études comme celle proposée par Chausse et al. (2005), affirment qu'un tel changement est lié en partie au changement des constellations satellite à l'origine de chaque donnée GPS. Étant donné que nous ne disposons pas de ces informations sur les données réelles, nous avons mené une expérience sur la séquence de synthèse décrite dans la section 6.4.1 où nous disposons de la connaissance a priori des changements du biais. Notre algorithme a été légèrement modifié pour prendre en compte cette connaissance a priori. Ainsi, dès que le biais change, une nouvelle correction est estimée et est appliquée uniquement sur les données GPS ayant le même biais. Pour évaluer cette nouvelle approche de correction de biais, nous analysons l'évolution de l'erreur *dans le plan* de la localisation issue de la fusion de SLAM avec le MET et les données GPS corrigées et celle des données GPS corrigées. Comme le montrent les courbes de la figure 7.13, nous remarquons que cette information supplémentaire permet d'améliorer l'estimation de la correction.



(a)



(b)

FIGURE 7.13 – **Évaluation sur les données de synthèse du processus de correction du biais du GPS avec (violet) et sans (bleu) la connaissance a priori des changements de biais.** (a) Évolution de l'imprécision des données GPS après la correction avec (violet) et sans (bleu) la connaissance a priori des changements de biais. (b) L'évolution de l'erreur de la localisation dans le plan avec (violet) et sans (bleu) la connaissance a priori des changements de biais.

CHAPITRE 8

Applications de Réalité Augmentée en milieu urbain

Dans les deux précédents chapitres nous avons proposé deux solutions pour fusionner les contraintes multi-vues avec les données GPS et les contraintes géométriques apportées par le MET et les modèles 3D des bâtiments. Tandis que la première solution est réalisée hors ligne, la deuxième approche de fusion s'effectue en ligne. Dans ce chapitre nous proposons un algorithme original qui exploite ces deux approches de fusion afin de fournir une application d'aide à la navigation à travers la Réalité Augmentée.

8.1 Introduction

Les systèmes d'aide à la navigation actuellement embarqués dans les véhicules offrent généralement une restitution de l'information de guidage sous la forme d'un tracé sur un plan de type carte routière, que celui-ci soit 2D ou 3D. Cette représentation étant différente de la perception que le conducteur du véhicule a de son environnement, ce dernier doit effectuer un effort pour transposer cette représentation symbolique vers sa perception du monde réel. Cet effort cognitif est d'autant plus important que l'environnement dans lequel le véhicule évolue est complexe. Ceci peut s'avérer gênant en zone urbaine dense où le système routier est généralement plus complexe et où la charge cognitive du conducteur devrait avant tout se focaliser sur les autres usagers de la route.

Face à ce problème, la réalité augmentée semble être une solution pertinente puisqu'elle permet de présenter l'information de navigation selon une modalité similaire à la perception que le conducteur a de l'environnement. La trajectoire à suivre étant affichée sur une image correspondant à la vision qu'a le conducteur de l'environnement, ce dernier n'a plus besoin d'effectuer cet effort de transposition entre virtuel et réel.

Cependant, pour qu'un tel système soit exploitable, ce dernier doit vérifier un ensemble de contraintes applicatives :

- ▷ Qualité de service : les informations virtuelles doivent s'aligner de manière précise avec l'environnement réel, sous peine de donner une mauvaise indication (eg. indiquer une mauvaise route) ou d'induire auprès du conducteur un effort cognitif important pour compenser cette imprécision. En particulier, ceci implique que la localisation soit sur six

degrés de libertés et pas uniquement deux degrés comme c'est le cas pour les systèmes GPS courant.

- ▷ Continuité de service : le système doit offrir une localisation à tout instant, quel que soit l'environnement.
- ▷ Facilité de déploiement : le coût de mise en place et d'exploitation du système doit être le plus réduit possible. En particulier, le matériel nécessaire pour son exploitation par l'utilisateur final doit être disponible et de faible coût.

Ainsi, d'une part, les approches proposant de se localiser par rapport à une base d'amers visuels (*e.g.* Dong et al. (2009), Irschara et al. (2009), Li et al. (2012)) ou de modèles 3D de ville texturés (*e.g.* Soheilian et al. (2013)) ne vérifient pas l'ensemble de ces contraintes. En effet, la création de ces bases d'amers reposent soit sur une flotte de véhicules dédiée (*e.g.* Soheilian et al. (2013)) embarquant généralement un matériel couteux (LIDAR, multiples caméras, etc), soit sur l'exploitation de bases d'images issues d'internet (*e.g.* Agarwal et al. (2011), Frahm et al. (2010)). Dans le premier cas, si le modèle obtenu sera d'une grande précision, ce dernier ne représentera l'apparence de la scène que pour une condition d'illumination donnée et ne pourra pas facilement être mis à jour en cas de changement dans l'environnement. Le système risque donc d'être soit peu robuste, soit extrêmement couteux en terme d'exploitation (flotte de véhicule mettant continuellement à jour le modèle). Dans le second cas (exploitation de bases d'images internet), si le modèle est obtenu à moindre coût (pas de flotte de véhicule), ce dernier sera principalement issu de point de vue « piéton » et non « véhicule ». La pertinence du modèle obtenu sera donc limitée vis-à-vis de l'application, et la continuité de service réduite (zones non cartographiées).

D'autre part, les approches n'utilisant pas de bases d'amers souffrent quant à elles généralement d'un défaut de qualité ou de robustesse. Ainsi, les méthodes consistant à fusionner le GPS avec d'autres capteurs (odomètre, angle de braquage, détection de ligne blanche, etc.) offrent généralement une estimation de seulement 3 degrés de liberté (position dans le plan du sol et cap, *e.g.* Chausse et al. (2005) *etc.*). Les méthodes basées sur la contrainte d'un SLAM avec un modèle 3D de ville, tels que proposées par Lothe et al. (2010) ou dans le chapitre 4 de ce mémoire, souffrent généralement d'un problème de robustesse à l'issue d'une longue ligne droite (accumulation d'erreur trop importante). Par ailleurs, les méthodes basées sur la fusion de la vision avec les données GPS, telle celle proposée dans le chapitre 5 sont sujettes à une erreur de localisation de l'ordre de celle du GPS employé. Enfin, la méthode de GPS différentiel basé sur les modèles 3D des bâtiments présentée dans le chapitre précédent offre à la fois une facilité de déploiement et une continuité de service mais présente une précision de localisation plus faible que les méthodes basées sur les bases d'amers.

L'objectif de ce chapitre est donc de proposer une solution exploitant une base d'amer géo-référencée mais offrant à la fois la qualité de service, la continuité de service et la facilité de déploiement nécessaire à une application de réalité augmentée pour l'aide à la navigation. Après avoir présenté le principe de l'approche (section 8.2), nous présenterons le fonctionnement de notre solution lorsque le véhicule navigue dans une zone pour laquelle une base d'amers exploitable est disponible (section 8.3). Nous présenterons ensuite la manière dont fonctionne notre solution lorsque le véhicule évolue dans une zone où aucune base d'amers n'est disponible, ainsi que la manière dont cette base peut être mise à jour (section 8.4). Enfin, nous présenterons comment notre solution gère la transition entre une zone pourvue d'une base d'amers valide et une zone dépourvue (section 8.5). Ce chapitre fera alors l'objet d'une évaluation expérimentale (section 8.6) ainsi que d'une discussion (section 8.7)

8.2 Approche proposée

Comme nous l'avons indiqué précédemment, le premier défaut des approches exploitant une base d'amers réside en l'absence de continuité de service lorsque le véhicule entre dans une zone où la base d'amers est absente ou incompatible avec les conditions d'illumination courante. Pour résoudre ce problème, nous proposons de passer dans un mode dégradé, reposant sur notre solution qui fusionne le SLAM avec le MET et le GPS différentiel basé sur les modèles 3D des bâtiments présentée dans le chapitre précédent, lorsque le véhicule entre dans une telle zone. A défaut de maintenir continuellement le même niveau de qualité de service, cette approche permet de maintenir sa continuité.

Le second défaut des approches exploitant une base d'amers est lié à la difficulté de la création et du maintien de cette base ainsi que le coût associé. Pour résoudre ce problème, nous proposons d'utiliser une approche collaborative pour la création de la base en question. En effet, une approche collaborative présente l'avantage de transformer l'ensemble des utilisateurs en une flotte de véhicules dédiés à la création de la base. Néanmoins, pour y parvenir, le processus utilisé doit exploiter les capteurs disponibles et ne pas requérir des transmissions de données trop volumineuses. De plus, cette modélisation doit pouvoir être réalisée tout en offrant un service de localisation. Comme nous le verrons, la méthode présentée dans le chapitre 6 permet de répondre à l'ensemble de ces critères.

8.3 Navigation en zone disposant d'une base d'amers

La majorité des méthodes de re-localisation existantes est basée uniquement sur des algorithmes de reconnaissance de point de vue (*e.g.* Dong et al. (2009), Arth et al. (2009), Tong et al. (2012), *etc.*). Le principe de ces méthodes consiste à mettre en correspondance les amers 2D de l'image courante (les points d'intérêt détectés) avec les amers 3D de la base géo-référencée préalablement construite. A partir de cette mise en correspondance, la pose actuelle de la caméra peut être estimée. Toutefois, ces solutions restent très sensibles aux conditions d'illumination et aux changements de point de vue. De plus ces solutions ne peuvent pas garantir une estimation de pose précise à chaque image puisque la base d'amers exploitée ne modélise pas forcément toute la scène observée. Pour apporter plus de robustesse face à cette limitation, nous choisissons d'améliorer la méthode proposée par Gay-Bellile et al. (2010) qui fusionne l'algorithme de reconnaissance de point de vue (*e.g* Ballas et al. (2012)) avec l'algorithme du SLAM visuel. Ce dernier apporte une certaine cohérence temporelle qui permet d'assurer la continuité de la localisation même dans le cas où l'algorithme de reconnaissance de point de vue échoue.

Le principe de cette méthode est le suivant. Le flux vidéo courant (c'est-à-dire enregistré en ligne par la caméra embarquée sur le véhicule) est traité en temps-réel par le SLAM visuel Mouragnon et al. (2006). Afin d'éviter la dérive inhérente à ce type de méthodes, l'idée consiste à corriger au fur et à mesure la reconstruction SLAM à partir des données géo-référencées de la base d'amers. Pour cela, un traitement spécifique est réalisé à chaque nouvelle image clé. Deux poses sont ainsi calculées pour cette image :

- ▷ La pose fournie par le module SLAM, cette pose étant erronée à cause de la dérive du facteur d'échelle.
- ▷ La pose calculée à partir de la base d'amers préalablement reconstruite. Cette pose est supposée correcte et précise. Nous verrons par la suite que plusieurs filtres sont utilisés

en pratique pour s'assurer que cette pose est correcte.

La transformation entre ces deux poses (figure 8.1(a)) définit complètement la dérive du SLAM : l'erreur de rotation, de translation ainsi que le facteur d'échelle (estimé par exemple soit à partir du nuage de points reconstruit ou par le rapport du déplacement entre deux positions consécutives de la caméra estimées par le module SLAM et celles estimées par le processus de relocalisation). Cette similitude est alors utilisée pour corriger la reconstruction SLAM. Tout d'abord, l'ensemble de la reconstruction est déplacée de telle sorte que la caméra clé courante ait la pose définie par les informations géo-référencées (figure 8.1(b)). Le facteur d'échelle est alors utilisé (figure 8.1(c)) pour corriger la norme du déplacement des n dernières poses de caméras clés considérées dans l'ajustement de faisceaux local. Ainsi, le facteur d'échelle transmis à la suite de la reconstruction est correct.

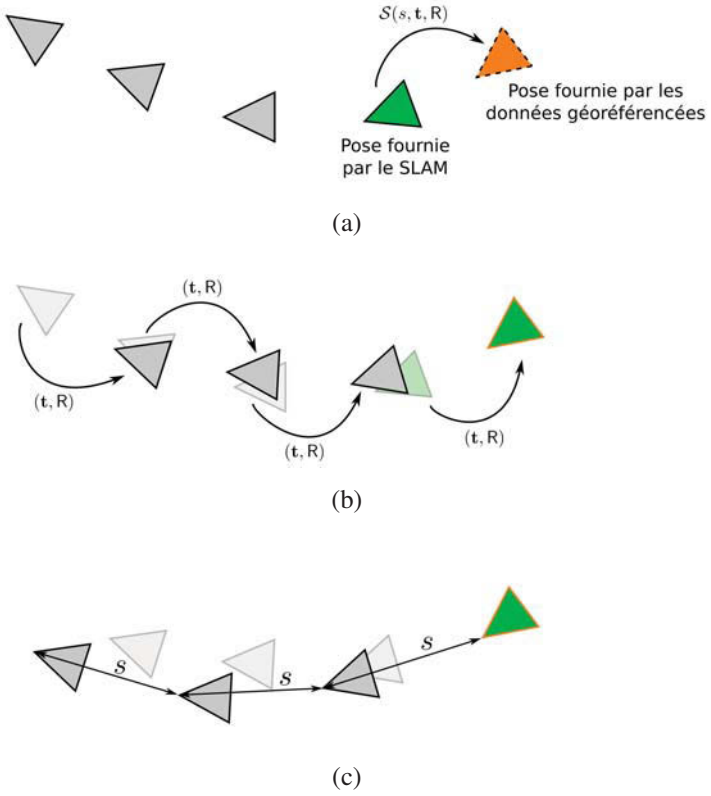


FIGURE 8.1 – **Couplage du SLAM avec des données géo-référencées (Image extraite du mémoire de Pierre Lothe 2010).** La reconstruction SLAM est corrigée en utilisant les données géo-référencées (a) : une transformation euclidienne est appliquée pour corriger la pose de la caméra clé courante (b) et le facteur d'échelle est alors corrigé (c).

Notons que le processus complet décrit dans cette section ne fonctionne que lorsque la pose calculée à partir des données géo-référencées est correcte. Or, il est possible que celle-ci soit erronée. Pour éviter cela, plusieurs filtres ont été mis en place : vérification des contraintes épipolaires, du ratio d'inliers conservés pour le calcul de la pose, de la bonne répartition de ces points dans l'image, *etc.* Ces différents filtres permettent de maximiser les chances d'obtenir une pose non-aberrante. Dès lors, deux cas de figure sont possibles : si la pose passe les filtres, la reconstruction SLAM est corrigée ; dans le cas contraire, aucune correction n'est appliquée et le processus de SLAM continue.

Néanmoins, à cause des erreurs de calcul qui peuvent survenir, l'estimation du facteur d'échelle en utilisant cette méthode reste peu précise. Or une propagation d'un facteur d'échelle erroné entraine des importantes imprécisions au niveau de l'estimation de la pose de la caméra. Pour pallier ces problèmes, nous avons choisi de remplacer l'utilisation du SLAM visuel proposé par Mouragnon et al. (2006) par l'approche introduite par Eudes et al. (2010a) qui fusionne le SLAM visuel avec la sortie d'un odomètre qui fournit avec plus de précision la norme de déplacement de la caméra. Le facteur d'échelle étant bien contraint, seules l'orientation et la translation de la caméra sont estimées par le processus de la relocalisation. Par ailleurs pour améliorer la précision du processus de la mise en correspondance, une comparaison exhaustive dans la base d'amers en délimitant la zone de recherche à une région de rayon de quelques mètres autour de la position fournie par le GPS est utilisée.

8.4 Navigation en zone dépourvue de base d'amers et mise à jour de la base

Considérons à présent le cas de figure où le véhicule se situe dans une zone pour laquelle aucune base d'amers n'est valide, ceci en raison de l'absence complète de base d'amers ou de la présence de base incompatible avec les conditions d'illumination. Deux tâches sont alors à réaliser : localiser le véhicule et mettre à jour la base d'amers. Pour y parvenir, nous proposons de combiner les méthodes présentées dans les précédents chapitres :

- ▷ Le véhicule est localisé à l'aide d'un SLAM contraint au MET et au GPS différentiel basé sur les modèles 3D des bâtiments si le véhicule est en milieu urbain (chapitre 7), ou avec une simple fusion du SLAM avec le GPS et MET si le véhicule est en zone rurale (chapitre 5).
- ▷ La reconstruction de l'environnement obtenue par le processus de localisation est envoyée à un serveur qui réalise l'étape hors ligne du processus de géo-localisation d'une reconstruction SLAM présentée dans le chapitre 6.

Tout d'abord, on notera que cette solution vérifie bien les critères évoqués dans la section 8.2. En effet, la localisation est bien réalisée en parallèle de la modélisation avec des capteurs bas coût et largement répandus (GPS standard, caméra VGA). De plus, la quantité de données à transférer au serveur est réduite puisque seule la reconstruction SLAM (point 3D, descripteurs et images clés) est à transmettre.

Aussi, on notera que le processus de localisation offre une qualité de service légèrement dégradée par rapport à la localisation basée sur la base d'amers. Néanmoins, la continuité de service est maintenue, et la fréquence d'apparition de ce cas de figure devrait diminuer avec le temps. En effet, l'approche collaborative permet d'obtenir une mise à jour fréquente de la base (*i.e.* chaque fois qu'un utilisateur passe dans une telle zone). Ceci permet donc d'avoir une couverture géographique importante, mais aussi une couverture des zones en terme de condition d'illumination (*i.e.* les différents véhicules passant sur une même zone à des heures et dates différentes).

Enfin, s'agissant du temps de traitement au niveau du serveur distant, on notera que les expériences sur le raffinement de la reconstruction présentées dans le chapitre 6 ont présenté des temps de traitement de l'ordre de 2 minutes pour 2400m. Si ce temps, sur une implémentation non GPU de l'algorithme, peut sembler déjà relativement raisonnable, on notera que ce temps est réduit dans les cas présents puisque l'utilisation d'un SLAM contraint au MET et au GPS

différentiel basé sur les modèles 3D des bâtiments fournit une initialisation plus proche de la solution que celle utilisée lors de cette expérimentation.

8.5 Transition entre zone avec base d'amers vers une zone sans base d'amers valide

Pour améliorer les performances du système, on notera que lors du passage du mode « basé amers » vers le mode « sans base d'amers », le biais affectant les données GPS peut être estimé à l'aide des dernières images clés obtenues à l'aide de la base d'amers. En effet, la précision de la localisation des images clés lorsqu'elles ont été obtenues à l'aide de la base d'amers permet d'estimer le biais du GPS directement à partir de la différence entre la donnée GPS brute et la position de l'image clé. Le biais ainsi estimé est alors utilisé comme initialisation dans le processus de SLAM contraint au MET et au GPS différentiel basé sur les modèles 3D des bâtiments.

8.6 Évaluation expérimentale

Cette section a pour objectif de démontrer la qualité de la localisation accessible par le système présenté dans ce chapitre. L'application cible étant la Réalité Augmentée, cette appréciation doit être évaluée en terme d'erreur de re-projection. Néanmoins, ne disposant pas de vérité terrain permettant d'évaluer précisément cette erreur, nous proposons ici une simple évaluation par appréciation visuelle. Dans un premier temps, nous proposons une expérience permettant de comparer sur une même séquence l'erreur de re-projection associée à une localisation en ligne dans une zone dépourvue de base d'amers, ce qui correspond au cas de figure le moins favorable pour notre méthode, avec celle obtenue à partir de la base d'amers issue de la méthode hors ligne du chapitre 6. Dans un second temps, nous présentons quelques images de résultats obtenues à partir de notre application de Réalité Augmentée basée sur la méthode introduite dans ce chapitre, ceci aussi bien dans le cas d'une navigation pour lequel une base d'amers est disponible que dans le cas contraire.

8.6.1 Comparaison de la précision de la navigation en zone disposant d'une base d'amers avec la navigation en zone dépourvue d'une base d'amers

Cette expérience a pour objectif d'évaluer la qualité de la localisation perçue par l'utilisateur final dans le pire des cas de notre méthode, à savoir la localisation en zone dépourvue de base d'amers. Ne disposant pas de véritable vérité terrain, nous proposons d'utiliser les poses issues de la méthode hors ligne présentée dans le chapitre 6 au titre de résultat de référence.

Protocole expérimental. Pour évaluer la précision de la localisation, nous proposons d'apprécier visuellement la re-projection des modèles 3D des bâtiments sur des images extraites d'une séquence enregistrée dans le quartier de Versailles (2400m, milieu urbain dense).

Résultats. La figure 8.2 présente une série d'images résultats. Les différentes re-projections des modèles 3D mettent en évidence deux types d'imprécisions d'origines différentes au niveau de la localisation en ligne basé sur un SLAM contraint au MET et au GPS différentiel basé sur les modèles 3D des bâtiments. Le premier type d'imprécisions a pour origine la fusion SLAM avec le GPS et le MET. En effet, tandis que le MET permet de contraindre implicitement l'angle tangage en ligne droite, l'angle roulis quant à lui se trouve peu contraint dans ce cas de figure, comme le montre la figure 8.2(a).

La source du deuxième type d'imprécisions est spécifique à la méthode de correction du biais du GPS. En effet, un certain retard de correction est notable. Ceci est dû soit au manque de contraintes fournies par les modèles 3D des bâtiments comme c'est le cas pour la figure 8.2(c) ou au manque de points reconstruits dans des zones d'intérêt notamment les façades orthogonales observées comme c'est le cas dans la figure 8.2(e). Malgré ces imprécisions, nous remarquons que quand les contraintes fournies par le MET (pour bien contraindre l'orientation) et les bâtiments (pour mieux estimer la correction) disponibles sont suffisantes, la précision atteinte par la fusion en ligne des contraintes est équivalente à celle obtenue à partir de la base d'amers comme le montre la figure 8.2(g).

8.6.2 Aide à la navigation en Réalité Augmentée

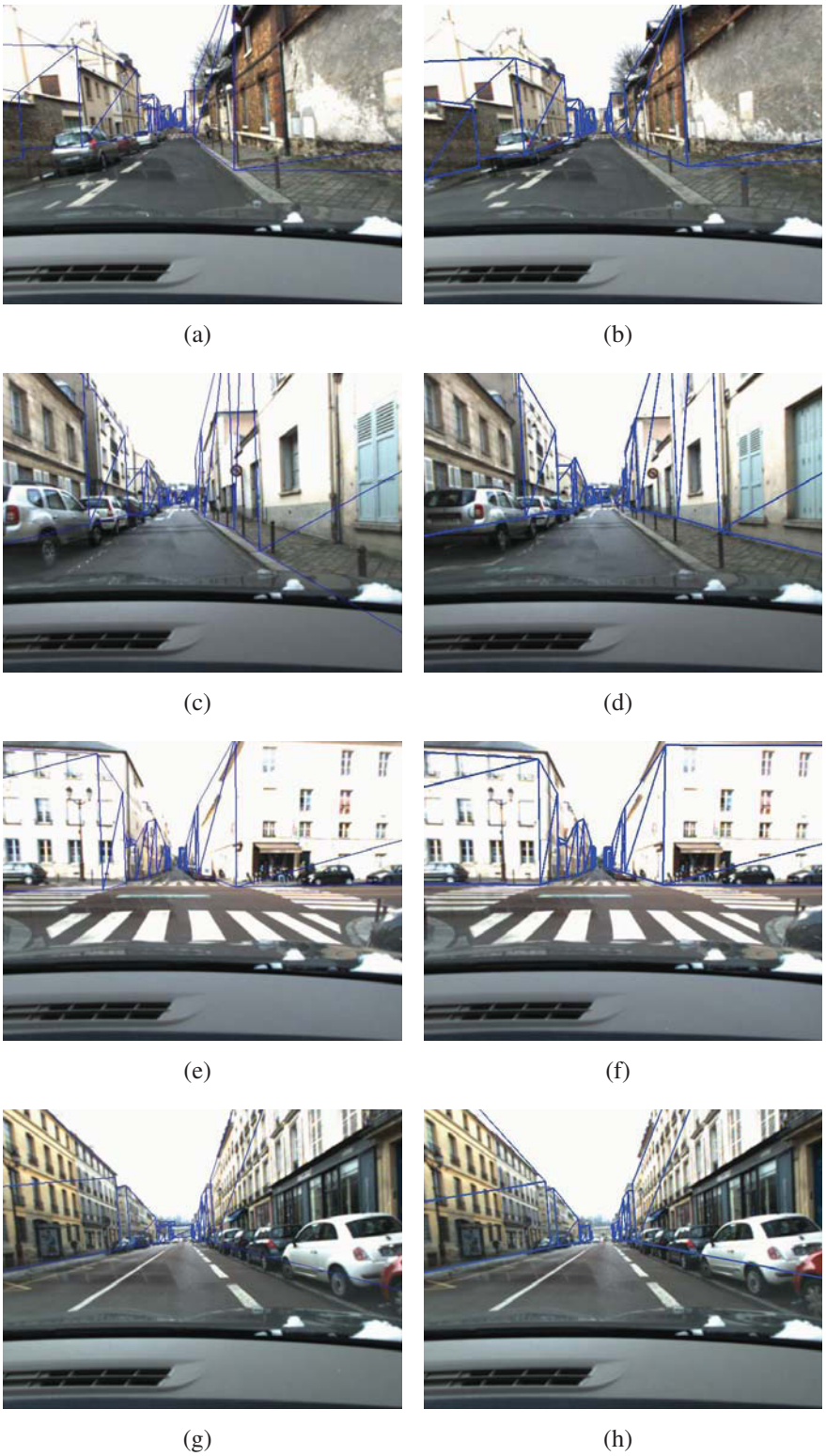
Cette expérience a pour objectif d'illustrer la qualité et le service pouvant être fourni par notre solution de localisation en ligne complète présentée dans ce chapitre.

Protocole expérimental. Pour évaluer la qualité de l'expérience de Réalité Augmentée ainsi que le service pouvant être offert, nous proposons d'apprécier visuellement les résultats obtenus par notre application d'aide à la navigation en Réalité Augmentée basée sur la méthode de localisation présentée dans ce chapitre. Cette évaluation est réalisée sur deux séquences réelles, et pour les différents cas de figures observables, à savoir la navigation en zone disposant d'une base d'amers et le cas inverse. La première séquence est enregistrée dans le quartier de Versailles (2400m) qui représente un exemple de milieu urbain dense. La deuxième séquence est enregistrée dans le quartier de Saclay (1200m) et qui représente un exemple de milieu péri-urbain. Comme le montre la figure 8.3, ces séquences présentent des conditions d'illuminations différentes de celles caractérisant les séquences utilisées lors de la création des bases d'amers.

Résultats Les figures 8.5 et 8.4 montrent que les localisations obtenues sont suffisamment précises pour assurer des applications convaincantes de Réalité Augmentée en insérant soit des informations routières (panneaux routier, passage piétons), des information de navigation (route à suivre), les noms des rues, ou en re-projetant des bâtiments d'intérêt. Ces résultats témoignent également de la grande précision atteinte par nos deux solutions de fusions de contraintes.

8.7 Discussion

La méthode présentée dans ce chapitre se distingue de la majorité des approches existantes par différents aspects. Tout d'abord, elle repose sur une modélisation collaborative, ce qui rend la méthode déployable à moindre coût. Ensuite, elle offre un niveau de qualité variable, celui-ci variant en fonction que la zone dispose d'une base d'amers compatible avec les conditions d'illumination courante ou non. Cependant, cette variabilité de la qualité de localisation devrait



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 8.2 – **Comparaison entre la localisation instantanée de la fusion en ligne (chapitre 7) et celle obtenue avec la fusion hors ligne (chapitre 6) : Re-projection des modèles 3D sur des images extraites de la séquence de Versailles.** La colonne de gauche représente les résultats obtenus à l'issu de la localisation en ligne. La colonne de gauche représente les résultats obtenus à l'issu de la localisation a posteriori.

FIGURE 8.3 – **Illustrations de séquences d'apprentissage et de test utilisées.** La première ligne correspond à des illustrations des séquences d'apprentissage et de test dans le quartier de Saclay. La deuxième ligne correspond à des illustrations des séquences d'apprentissage et de test dans le quartier de Versailles.

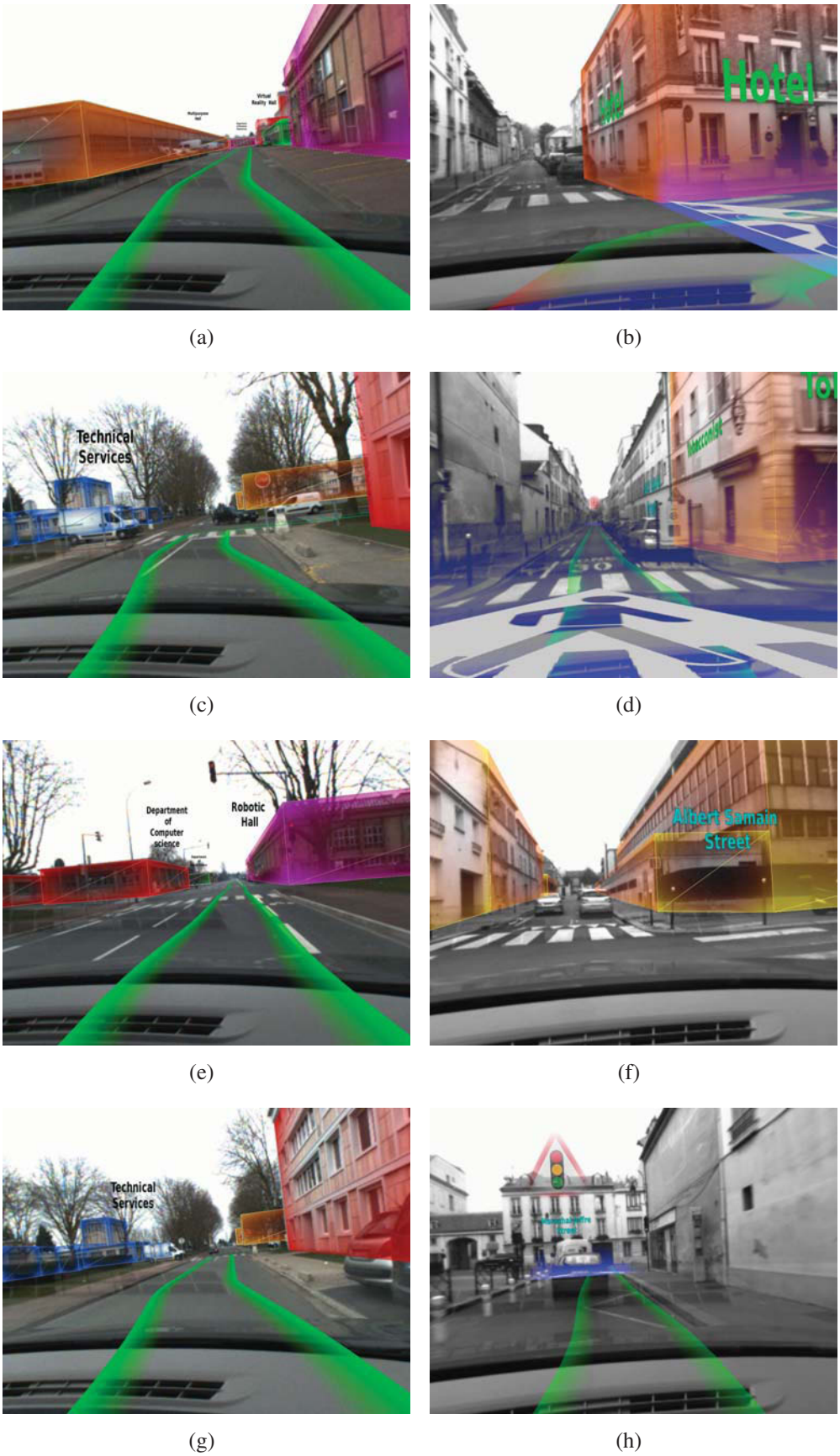
(a) (b)

FIGURE 8.4 – **Des exemples d'applications de Réalité Augmentée en utilisant la fusion en ligne des contraintes (chapitre 7) :** projection des modèles des bâtiments et la trajectoire du véhicule.

se stabiliser au fur et à mesure de l'utilisation du système. En effet, au cours du temps, l'exploitation du système par les utilisateurs va permettre d'obtenir une base d'amers couvrant à la fois l'ensemble des zones géographiques mais aussi l'ensemble des conditions d'illumination. La fréquence d'apparition du cas de figure correspondant à une navigation en zone ne disposant pas d'une base d'amers valide devrait donc diminuer au fil du temps. On notera aussi que cette fréquence baissera d'autant plus vite sur les axes les plus fréquentés par les utilisateurs.

Un autre avantage de notre méthode par rapport aux autres méthodes exploitant une base d'amers vient du fait qu'elle offre une résistance à la perte de connexion internet. En effet, si la base d'amers est obtenue à l'aide d'une connexion internet mobile, l'indisponibilité de ce type de connexion entraine l'échec des méthodes classiques exploitant la base en question. Dans le cas de notre méthode, une perte de connexion internet revient simplement à naviguer en zone dépourvue d'une base d'amers valide.

L'utilisation d'une approche collaborative engendre néanmoins un certain nombre de problèmes qu'il faudra résoudre à terme. En particulier, la taille de la base d'amers risque de croître



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 8.5 – **Des exemples d'applications de Réalité Augmentée en exploitant des bases d'amers créées à travers la fusion hors ligne des contraintes (chapitre 6) :** projection des modèles des bâtiments, insertion des informations routières et la trajectoire du véhicule.

rapidement. Il sera donc nécessaire d'étudier la robustesse de la méthode d'indexation à une telle montée en charge. Une procédure de « maintenance » visant à supprimer les données redondantes ou inutiles est aussi à envisager. Celle-ci pourrait se baser sur des statistiques sur l'utilisation des amers de la base par exemple.

Conclusion

Travaux réalisés

L'objectif principal de cette thèse était de proposer une solution de géo-localisation d'un véhicule en milieu urbain qui soit à la fois peu coûteuse, facile à déployer et offrant des performances compatibles avec des applications de Réalité Augmentée.

Si de nombreux travaux traitant la problématique de la localisation de véhicule existent, certains d'entre eux reposent sur l'exploitation de capteurs coûteux (*e.g.* LIDAR) ou complexes à intégrer (*e.g.* caméra catadioptriques). Les solutions reposant sur des capteurs grand public, quant à elles, n'offrent généralement pas la précision ou la robustesse requise pour une application de Réalité Augmentée. En effet, la plupart de ces solutions estime une localisation selon trois degrés de libertés uniquement (*i.e.* longitude, latitude et cap). Ceci est certes suffisant pour une localisation dans le plan 2D de déplacement mais reste incompatible avec une application de Réalité Augmentée qui requiert une localisation selon les six degrés de liberté de la caméra. Les rares solutions respectant ce critère (*i.e.* une localisation selon les six degrés de libertés de la caméra) offrent souvent une faible précision, un manque de robustesse ou encore requièrent un processus de déploiement complexe, coûteux ou difficile à maintenir dans le temps.

Pour parvenir à une solution acceptable vis-à-vis de ces différents critères, nous avons choisi de retenir l'approche de SLAM visuel contraint basé image clé. Cette récente approche présente l'avantage d'offrir une localisation à six degrés de liberté. Des premières solutions ont démontré la possibilité d'obtenir une localisation robuste mais peu précise (*e.g.* Lhuillier (2012)) ou précise mais peu robuste (*e.g.* Lothe et al. (2009), Tamaazousti et al. (2011)) à l'aide de capteurs grand public et de données disponibles à grande échelle. La principale contribution de cette thèse fut donc de proposer une solution pour combiner ces approches afin de bénéficier de leurs avantages respectifs. Le principal défi auquel les travaux menés ont dû faire face fut la difficulté à intégrer au sein d'un ajustement de faisceaux des contraintes issues de données et capteurs hétérogènes (*i.e.* grande variabilité des précisions des mesures en fonction du capteur ou de la donnée utilisés) sans en perturber la convergence. En particulier, nous nous sommes principalement intéressés à l'exploitation d'une unique caméra, d'un GPS standard, des modèles d'élévation de terrain et des modèles 3D des bâtiments d'un système d'information géographique. Ce choix se justifie par le fait que ces capteurs et données sont bas coût et disponibles (*e.g.* sur un smarphone).

Nos principales contributions furent donc :

- ▷ **Une solution de SLAM en ligne contraint à des modèles 3D des bâtiments et un MET**. Nous avons proposé de modifier l'ajustement de faisceaux contraint aux modèles 3D des bâtiments introduite par Tamaazousti et al. (2011) de manière y intégrer une contrainte supplémentaire sur l'altitude de la caméra. Ainsi formulé, le problème de localisation est moins sujet aux dérives au niveau des degrés de libertés peu ou pas contraints dont souffrait la solution de Tamaazousti et al. (2011) lorsqu'elle était appliqué au contexte urbain. Pour accroître encore plus la robustesse de ce processus, une nouvelle méthode d'identification des points 3D reconstruits correspondant aux façades des bâtiments a été introduite. Cependant, la solution complète souffre d'un manque de précision et de robustesse lorsque l'environnement est pauvre en bâtiments (*i.e.* milieu péri-urbain et rural).
- ▷ **Une solution de SLAM en ligne contraint simultanément à un MET et à un GPS**. Nous avons proposé de modifier la contrainte au GPS introduite par Lhuillier (2012) afin d'y intégrer une contrainte sur l'altitude de la caméra, cette dernière étant déterminée à l'aide d'un modèle d'élévation de terrain. Nous avons montré que cette solution permettait non seulement d'améliorer la dérive en altitude dont souffrait la solution de Lhuillier (2012) mais aussi d'améliorer l'estimation des degrés de liberté *hors plan* restant (*i.e.* angles tangage et roulis). Cette solution reste cependant limitée en termes de précision *dans le plan* en raison du biais dont souffrent les données GPS.
- ▷ **Une solution de SfM hors ligne exploitant les contraintes GPS, MET et bâtiments**. Nous avons proposé d'intégrer les contraintes GPS, MET et bâtiment 3D au sein d'un même processus de SfM. Pour éviter les problèmes de convergence que pourraient introduire la présence d'un biais dans les données GPS vis-à-vis des contraintes liées aux modèles 3D des bâtiments, ces deux contraintes sont exploitées successivement et non simultanément. Cette solution entièrement automatique permet d'obtenir une reconstruction précise de l'environnement et de la trajectoire de la caméra selon ses six degrés de liberté tout en offrant une bonne robustesse. Cette solution offre également l'intérêt de réduire le temps de traitement de processus de création de bases d'amers géo-référencées puisqu'une grande partie des calculs peut être réalisée simultanément à l'acquisition des données.
- ▷ **Une solution de SLAM en ligne exploitant les contraintes GPS, MET et bâtiments**. Afin d'accroître la précision du SLAM en ligne contraint au GPS et MET, nous avons proposé d'exploiter la connaissance des modèles 3D des bâtiments pour estimer en ligne le biais affectant les données GPS. Cette estimation permet non seulement de corriger les données GPS passées mais aussi de prédire la correction à appliquer aux prochaines données GPS. Grâce à ces corrections, le processus de localisation bénéficie de données GPS plus précises et offre donc une précision de localisation accrue. Bien que moins précise que la solution de SfM hors ligne, cette solution permet d'atteindre en ligne des précisions acceptables pour des applications de Réalité Augmentée.
- ▷ **Un système collaboratif de Réalité Augmentée pour le véhicule**. Nous avons proposé de combiner les deux précédentes contributions au sein d'un système d'aide à la navigation exploitant soit une base d'amers pré-existante si celle-ci est disponible, soit notre méthode de localisation SLAM GPS/MET/Bâtiments 3D en ligne. L'intérêt de cette approche réside dans le fait que les utilisateurs finaux participent à la création des bases d'amers tout en bénéficiant du système d'aide à la navigation mis en place. En plus de la facilité du déploiement, ce système assure la continuité du service puisque la méthode de SLAM GPS/MET/Bâtiments 3D en ligne prend le relais lorsque la base d'amers n'est

pas existante.

Perspectives

Les différentes solutions proposées pour intégrer au SLAM des contraintes issues d'un GPS standard, un MET ou les modèles 3D des bâtiments ont permis d'améliorer grandement la localisation du SLAM classique. Les résultats obtenus sont encourageants et confirment l'intérêt de telles approches comme alternative d'une localisation basée uniquement sur les systèmes GPS. Les études et expérimentations réalisées nous ont également permis de mettre en évidence certaines perspectives directes de nos travaux.

Amélioration de la localisation

Pour accroitre la précision de la localisation, il est possible d'apporter des améliorations au niveau de certaines contraintes utilisées mais aussi intégrer des nouvelles contraintes fournies par des capteurs disponibles et bas coûts.

Contrainte aux bâtiments

Meilleure différenciation des points 3D reconstruits. Dans ce mémoire nous avons proposé une amélioration de la méthode de segmentation du nuage de points (*i.e.* point du modèle/ point d'environnement) proposée par Tamaazousti et al. (2011) tout en essayant de respecter la contrainte temps réel. Malgré l'amélioration de la qualité de la segmentation, la méthode proposée reste sujette à quelques imprécisions. Pour améliorer la classification des points 3D, nous pouvons, par exemple, utiliser une méthode d'apprentissage afin de détecter dans les images les éléments généralement observés dans un milieu urbain (arbres, véhicules garés sur le bas-côté, routes, bâtiments, *etc.*).

Densifier la reconstruction pour mieux contraindre. La précision des approches basées sur les contraintes aux bâtiment dépend principalement des informations géométriques disponibles. En effet, agissant sur la reconstruction, les contraintes aux bâtiments dépendent des façades visibles et la distribution du nuage de points dans l'espace. Pour apporter le plus de contraintes possibles, il est envisageable de densifier le nuage de points reconstruit au niveau des zones d'intérêt telles qu'à proximité des intersections. Introduire plus de contraintes sur la reconstruction permet d'une part d'améliorer la convergence d'un ajustement de faisceaux incluant les contraintes aux bâtiments et d'autre part d'apporter plus de robustesse au processus de correction du biais du GPS.

Contrainte au GPS

Afin d'améliorer la précision de la localisation des approches basées sur l'exploitation des données GPS, nous avons proposé un processus de correction du biais affectant ces données. Si l'approche actuelle de correction du biais permet d'avoir une précision de localisation en ligne jusqu'ici inaccessible avec un GPS standard, cette précision reste inférieure à celle obtenue a posteriori à travers la fusion hors ligne des contraintes fournie par le GPS, le MET et les modèles 3D des bâtiments. Pour accroitre cette précision, il est intéressant d'étudier la possibilité

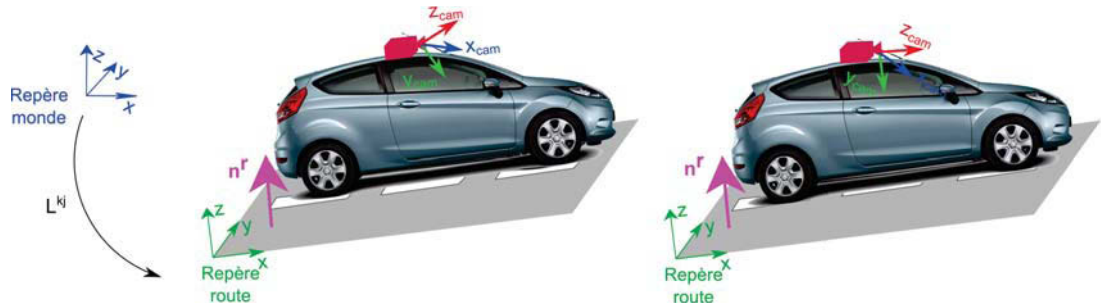


FIGURE 8.6 – **Principe de la contrainte en orientation** : encourager l'axes $\mathbf{y}_{cam}$ à avoir la même direction et le sens opposé à la normale à la route $\mathbf{n}^r$.

d'estimer le biais du GPS au sein d'un ajustement de faisceaux intégrant les contraintes aux bâtiments. Il est également intéressant de pouvoir identifier les données GPS partageant un même biais afin d'avoir un problème mieux contraint et donc obtenir une estimation plus précise et plus sûre.

Intégration de nouvelles contraintes

Une des limites actuelles de nos solutions est le manque de précision au niveau des angles roulis et tangages à cause de l'absence d'une contrainte explicite affectant ces paramètres. Pour faire face à cette limitation, il est possible d'introduire une nouvelle contrainte en orientation dans l'ajustement de faisceaux. Ceci peut être réalisé en adoptant l'hypothèse que les angles roulis et tangage sont considérés constant par rapport au plan de la route vu que la caméra est rigidement liée au véhicule. Étant donné que la représentation du MET utilisé est simplifiée, il est difficile de contraindre efficacement l'orientation de la caméra spécialement ses angles roulis et tangage en utilisant uniquement ces modèles. Toutefois, cette information peut être fournie par une centrale inertielle. En effet, il est possible de déduire à partir de la centrale inertielle la normale à la route $\mathbf{n}^r$ à chaque image clé. A partir de cette donnée, nous pouvons formuler la contrainte en orientation comme suit. En effet, comme le montre la figure 8.6, contraindre l'orientation revient à faire tendre le vecteur $\mathbf{y_{cam}}$ vers l'inverse de la normale à la route $\mathbf{n}^r$.

Ainsi un terme d'accorche aux données supplémentaire peut être introduit dans la fonction de coût avec la contrainte d'inégalité. Pour chaque pose de caméra, ce terme est donné par $c_{orientation} = \|\mathcal{R}\begin{pmatrix}0\\1\\0\end{pmatrix} - \mathbf{n}^r\|^2$. L'ajout de ce terme impliquerait une étude du poids qu'il faut lui attribuer.

En plus de la contrainte en orientation à travers les données d'une centrale inertielle, il est possible également d'améliorer la précision de l'estimation du facteur d'échelle, en contraignant le déplacement relatif de la caméra en exploitant les données fournies par une odomètre.

Amélioration de l'application finale

Plusieurs pistes d'amélioration sont envisageables afin de rendre nos solutions plus déployables :

- ▷ Il est envisageable d'étudier la possibilité d'exploiter des caméras infra rouge pour assurer le fonctionnement de nuit de nos solutions.

- ▷ Il est intéressant d'améliorer les performances en termes de temps de calcul de nos ajustements de faisceaux plus précisément celui basé sur la contrainte d'inégalité. Ce dernier nécessite un nombre important d'images clés (actuellement 40) afin d'assurer une bonne fusion des contraintes multi-vues avec celles du GPS. Une réduction des temps de calcul permettrait l'exploitation de nos solutions sur un simple smartphone par exemple.
- ▷ En ce qui concerne la gestion des bases d'amers collaboratives, il est important de traiter le problème lié à l'augmentation rapide de la taille des bases.

ANNEXE A

Ajustement de faisceaux contraint

Dans cette annexe, nous détaillerons les calculs de dérivées analytiques associées aux fonctions de coût utilisées au cours de nos travaux. Pour ceci, nous nous intéresserons en premier lieu à la fonction de coût standard de l'ajustement de faisceaux. Par la suite, nous détaillerons les fonctions de coût intégrant des contraintes supplémentaires : les contraintes SIG (i.e. la contrainte des modèles des bâtiments ainsi que la contrainte dure en altitude) et la contrainte GPS fusionnée à la contrainte douce et dure en altitude.

A.1 Ajustement de faisceaux basé sur la fonction de coût standard

Comme nous l'avons détaillé dans la section 1.3.2.5, la fonction de coût utilisée dans l'ajustement de faisceaux consiste à minimiser l'erreur de re-projection standard. Rappelons que la projection du point 3D $\tilde{\mathcal{Q}} = (X, Y, Z, 1)^T$ en un point 2D de coordonnées homogènes $\tilde{\mathbf{p}} = (p_1, p_2, p_3)^T$ par la caméra $\mathcal{C}$, définie par sa rotation $\mathcal{R}$, sa translation $\mathbf{t}$ et sa matrice de paramètres intrinsèques K, s'écrit :

$$\begin{pmatrix} p_1/p_3 \\ p_2/p_3 \end{pmatrix} = \pi(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}}). \tag{A.1}$$

Ainsi l'erreur de re-projection pour l'observation 2D $\mathbf{q} = (q_1, q_2)^T$ associé au point 3D $\tilde{\mathcal{Q}}$ est définie par :

$$\mathbf{f} = \begin{pmatrix} q_1 - p_1/p_3 \\ q_2 - p_2/p_3 \end{pmatrix} = \mathbf{q} - \pi(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}}). \tag{A.2}$$

La dérivée de $\mathbf{f}$ par rapport aux paramètres de caméra et les paramètres du point 3D est donnée par

$$d\left(\mathbf{f}\right) = \left(\frac{\partial \mathbf{f}}{\partial \mathcal{R}} | \frac{\partial \mathbf{f}}{\partial \mathbf{t}} | \frac{\partial \mathbf{f}}{\partial \mathcal{Q}}\right) = d\left(\pi\right) \times d\left(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}}\right). \tag{A.3}$$

π représente la matrice qui permet de passer des coordonnées homogènes aux coordonnées non homogènes d'une observation 2D. Sa dérivée est donnée par :

$$d(\pi) = \begin{pmatrix} \frac{1}{p_3} & 0 & \frac{-p_1}{p_3^2} \\ 0 & \frac{1}{p_3} & \frac{-p_2}{p_3^2} \end{pmatrix}. \tag{A.4}$$

Il reste maintenant à calculer les dérivées de $\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}}$ par rapport aux paramètres de la caméra $(\mathcal{R}|\mathbf{t})$ et les paramètres du point 3D $\mathcal{Q}$.

Dérivées par rapport aux paramètres de la caméra.

▷ Dérivées par rapport aux paramètres de la rotation. Ces dernières sont calculées en utilisant un repère relatif au voisinage de la rotation courante $\mathcal{R}_0$, telle que $\mathcal{R}_0(\omega) = \mathcal{R}(\omega)\mathcal{R}_0$, avec $\omega = (\omega_1, \omega_2, \omega_3)^T$ contient les faibles angles d'Euler autour de $\mathcal{R}_0$ et :

$$\mathcal{R}(\omega) = \begin{pmatrix} cos(\omega_1) & -sin(\omega_2) & 0 \\ sin(\omega_1) & cos(\omega_2) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos(\omega_1) & 0 & sin(\omega_2) \\ 0 & 1 & 0 \\ -sin(\omega_2) & 0 & cos(\omega_2) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\omega_1) & -sin(\omega_2) \\ 0 & sin(\omega_1) & cos(\omega_2) \end{pmatrix}. \tag{A.5}$$

Au voisinage de $\mathcal{R}_0$, $\mathcal{R}(\omega)$ peut s'écrire sous la forme suivante :

$$\mathcal{R}(\omega) = \mathsf{I}_{3\times3} + [\omega]_\times + o(\omega) \approx \begin{pmatrix} 1 & -\omega_3 & \omega_2 \\ \omega_3 & 1 & -\omega_1 \\ -\omega_2 & \omega_1 & 1 \end{pmatrix}. \tag{A.6}$$

Ainsi, à chaque itération de l'ajustement de faisceaux, les dérivées par rapport aux angles euler α, β et γ correspondant à la matrice de rotation $\mathcal{R}$ dans le repère global sont remplacées par les dérivées par rapport aux angles ω_1, ω_2 et ω_3 dans le repère local lié à la matrice de rotation $\mathcal{R}_0$. Par conséquent :

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \omega} &= d(\pi) \times \frac{\partial}{\partial \omega} \left(\mathsf{K}\mathcal{R}_0^T(\omega) \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}} \right) \\ &= d(\pi) \times \mathsf{K} \times \frac{\partial}{\partial \omega} \left(\left(\mathcal{R}(\omega)\mathcal{R}_0\right)^T \right) \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}} \\ &= d(\pi) \times \mathsf{K}\mathcal{R}_0^T \times -\frac{\partial}{\partial \omega} \left(\mathcal{R}^T(\omega) \right) \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}} \end{aligned} \tag{A.7}$$

Avec l'hypothèse des faibles angles, chaque dérivée peut s'écrire comme suit :

$$\frac{\partial \mathbf{f}}{\partial \omega} = d(\pi) \times \mathsf{K}\mathcal{R}_0^T \times -\frac{\partial \left[\omega\right]_\times}{\partial \omega_j} \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}}, \tag{A.8}$$

où les $\frac{\partial [\omega]_\times}{\partial \omega_j}$, $j \in \{1, 2, 3\}$ sont :

$$\frac{\partial \left[\omega\right]_\times}{\partial \omega_1} \approx \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{A.9}$$

$$\frac{\partial \left[\omega\right]_\times}{\partial \omega_2} \approx \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \tag{A.10}$$

$$\frac{\partial \left[\omega\right]_{\times}}{\partial \omega_3} \approx \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{A.11}$$

▷ Dérivées par rapport aux paramètres de la translation. Elles sont données par :

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \mathbf{t}} &= d(\pi) \times \frac{\partial}{\partial \mathbf{t}} \left(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t} \right] \tilde{\mathcal{Q}} \right) \\ &= d(\pi) \times \mathsf{K}\mathcal{R}^T \times \frac{\partial}{\partial \mathbf{t}} \left(\left[\mathsf{0}_{3\times 3} | - \mathbf{t} \right] \tilde{\mathcal{Q}} \right) \\ &= -1 \times d(\pi) \times \mathsf{K}\mathcal{R}^T. \end{aligned} \tag{A.12}$$

Dérivées par rapport aux paramètres des points 3D. Ces dernières s'obtiennent immédiatement en dérivant par rapport à $\mathcal{Q}$:

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \mathcal{Q}} &= d(\pi) \times \frac{\partial}{\partial \mathcal{Q}} \left(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t} \right] \mathcal{Q} \right) \\ &= d(\pi) \times \mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t} \right] \end{aligned} \tag{A.13}$$

Construction du système creux. La fonction de coût associée à l'erreur de re-projection standard est calculée comme suit :

$$f\left(\{\mathcal{R}_j, \mathbf{t}_j\}_{j=1}^{N}, \{\mathcal{Q}_i\}_{i=1}^{M} \right) = \sum_{i=1}^{i=M} \sum_{j \in \mathcal{D}_i} \|\mathbf{f}_{i,j}\|^2, \tag{A.14}$$

Avec $\mathbf{f}_{i,j}$ est l'erreur de re-projection correspondante au $i^{ème}$ point 3D observé par la $j^{ème}$ caméra.

Dans ce qui suit, nous souhaitons optimiser cette fonction de coût en utilisant l'algorithme de Levenberg Marquardt exploitant la nature creuse de la Jacobienne et la Hessienne.

Le calcul des dérivées de chaque $\mathbf{f}_{i,j}$ permet de construire la matrice Jacobienne J et ainsi de construire le système des équations normales. Cependant, l'évaluation complète de la matrice Jacobienne n'est pas indispensable pour la résolution du système. Il est possible de calculer directement la Hessienne $\mathsf{H} \approx \mathsf{J}^T\mathsf{J}$ ainsi que le vecteur gradient $\mathbf{g} = \mathsf{J}^T\mathbf{r} = (\mathbf{g}_{caméra}^T, \mathbf{g}_{point}^T)$, avec $\mathbf{r}$ est le vecteur des résidus concaténant les erreurs de re-projection $\mathbf{f}_{i,j}$. Le système à résoudre est représenté dans la figure A.1. Les matrices U et V sont diagonales par blocs où chaque bloc a une dimension de 6×6 et 3×3 respectivement. Quant à la matrice W, elle contient des blocs de dimension 6×3. Ces différentes matrices sont construites d'une façon incrémentale. Au départ, chacun des blocs $\mathsf{U}_j, \mathsf{V}_i$ et W_{ij} est initialisé à zéro. Ensuite, ils sont mis à jour pour chaque observation (i, j) :

▷
$$\mathsf{U}_j = \mathsf{U}_j + \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{R}_j} | \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{t}_j} \right)^T \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{R}_j} | \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{t}_j} \right). \tag{A.15}$$

▷
$$\mathsf{V}_i = \mathsf{V}_i + \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{Q}_i} \right)^T \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{Q}_i} \right). \tag{A.16}$$

▷
$$\mathsf{W}_{i,j} = \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{R}_j} | \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{t}_j} \right)^T \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{Q}_i} \right). \tag{A.17}$$

▷

$$\mathbf{g}_{caméra}[j] = \mathbf{g}_{caméra}[j] + \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{R}_j} | \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{t}_j}\right)^T \times \mathbf{f}_{i,j}. \tag{A.18}$$

▷

$$\mathbf{g}_{point}[i] = \mathbf{g}_{point}[i] + \left(\frac{\partial \mathbf{f}_{i,j}}{\partial \mathcal{Q}_i}\right)^T \times \mathbf{f}_{i,j}. \tag{A.19}$$

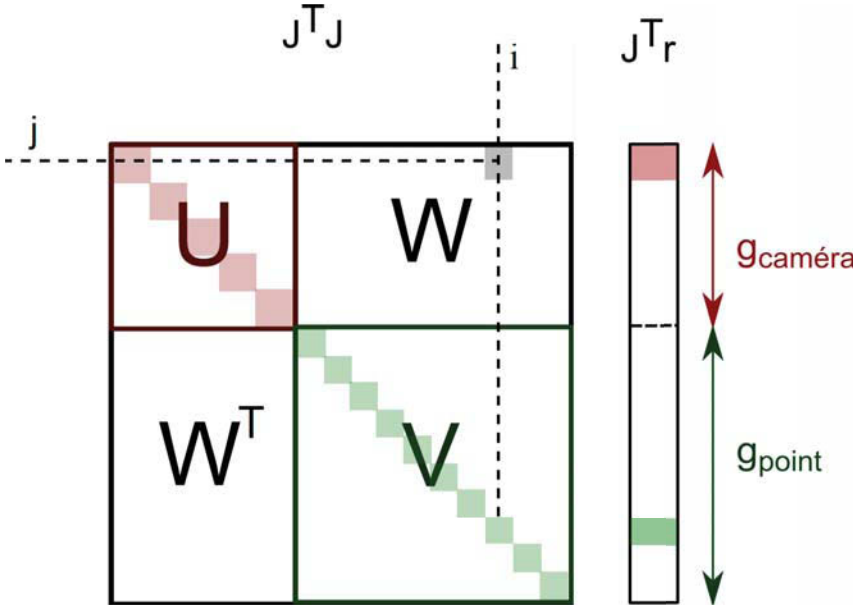


FIGURE A.1 – **Structure de la matrice Hessienne de l'ajustement de faisceaux.**

Le système obtenu a ainsi la forme suivante :

$$\begin{gathered} \mathsf{H}\delta = \mathbf{g} \\ \begin{pmatrix} \mathsf{U} & \mathsf{W} \\ \mathsf{W}^T & \mathsf{V} \end{pmatrix} \begin{pmatrix} \delta_{caméra} \\ \delta_{point} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_{caméra} \\ \mathbf{g}_{point} \end{pmatrix}. \end{gathered} \tag{A.20}$$

A l'aide du complément de Schur, ce système peut être résolu en deux étapes :

1. Le calcul des incréments δ_{camera} à appliquer aux paramètres extrinsèques de la caméra par résolution du système linéaire suivant :

$$\left(\mathsf{U} - \mathsf{W}\mathsf{V}^{-1}\mathsf{W}^T\right)\delta_{camera} = \mathbf{g}_{camera} - \mathsf{W}\mathsf{V}^{-1}\mathbf{g}_{point}. \tag{A.21}$$

2. Le calcul direct des incréments δ_{point} applicables aux points 3D :

$$\delta_{point} = \mathsf{V}^{-1}\left(\mathbf{g}_{point} - \mathsf{W}^T\delta_{camera}\right). \tag{A.22}$$

Pour l'ajustement de faisceaux local, les paramètres optimisés sont : N poses de caméra et M points 3D visibles dans les N images correspondantes. Cela signifie que les matrice U, V et W sont réduites. Elles sont respectivement de taille $6N \times 6N$, $3M \times 3M$ et $6N \times 3M$.

Notons qu'il est possible d'utiliser un estimateur robuste au niveau de la fonction de coût afin d'assurer plus de robustesse face aux données aberrantes. Son utilisation ne change pas la structure du système. Seule une matrice de poids apparait au niveau de la Hessienne et du vecteur gradient. Par conséquent, le même principe est utilisé pour la résolution. Afin de ne pas surcharger les notions dans la suite de cette annexe, nous présenterons les fonctions de coût sans estimateur robuste.

A.2 Ajustement de faisceaux basé sur une fonction de coût avec une contrainte dure

Les modèles SIG sont exempts de dérive et de données aberrantes. Ces avantages permettent d'introduire les informations fournies par ces modèles dans la fonction de coût à travers des contraintes dures. Ce genre de contraintes implique la réduction de degrés de liberté à optimiser au cours de l'ajustement de faisceaux. Nous détaillerons dans la suite les modifications à apporter au niveau des dérivées de la fonction de coût tout d'abord pour prendre en compte la contrainte liée aux modèles des bâtiments ensuite celle liée au MET.

A.2.1 Fonction de coût avec la contrainte des modèles 3D bâtiments

Tamaazousti et al. (2011) ont proposé une fonction de coût bi-objective qui prend en compte à la fois les contraintes multi-vues et les contraintes géométriques fournies pas les modèles des bâtiments (voir section 3.2). Pour ceci, le nuage de points reconstruit est segmenté en un premier ensemble $\mathcal{M}$ contenant les points 3D associés aux modèles des bâtiments (représentant les façades des bâtiments) et un second ensemble $\mathcal{E}$ contenant les points 3D restants (représentant la partie inconnue de l'environnement : arbres, voitures garée, panneaux routier...). Pour ce deuxième ensemble la fonction de coût utilisée représente la fonction de coût standard dont les dérivées sont détaillées dans la section précédente. Quant au premier ensemble (contenant les points 3D associés aux modèles des bâtiments), l'erreur de re-projection est donnée par :

$$\mathbf{g} = \begin{pmatrix} q_1 - p_1/p_3 \\ q_2 - p_2/p_3 \end{pmatrix} = \mathbf{q} - \pi(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \left(\tilde{\mathsf{M}}^h\right)^{-1} \tilde{\mathcal{Q}}^h), \tag{A.23}$$

avec $\tilde{\mathcal{Q}}^h = (X^h, Y^h, 0, 1)^T$. Ce nouveau point est défini tel que $\tilde{\mathcal{Q}}^h = \tilde{\mathsf{M}}^h \tilde{\mathcal{Q}}$, où $\tilde{\mathsf{M}}^h$ est la matrice de passage homogène du repère monde au repère de la façade $\mathbf{\Pi}^h$ du bâtiment (voir les détails de la définition du repère en question dans la figure 3.4). Au cours de l'optimisation, seuls les 2 degrés de liberté (X^h, Y^h) sont raffinés dans le nouveau repère. Ceci garantit l'appartenance du point $\mathcal{Q}$ à sa façade.

Dans ce qui suit nous détaillerons les dérivées de $\mathbf{g}$ par rapport aux paramètres de la caméra $(\mathcal{R}|\mathbf{t})$ et les paramètres du point 3D $\mathcal{Q}^h$.

Dérivées par rapport aux paramètres de la caméra. En suivant les mêmes étapes et en adoptant les mêmes hypothèses détaillées précédemment, nous obtenons des dérivées par rapport aux paramètres de la caméra similaires à ceux de la fonction de coût standard. En effet, les dérivées par rapport aux paramètres de la rotation sont données par :

$$\begin{aligned} \frac{\partial \mathbf{g}}{\partial \omega} &= d(\pi) \times \mathsf{K}\mathcal{R}_0^T \times -\frac{\partial \left[\omega\right]_\times}{\partial \omega_j} \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \left(\tilde{\mathsf{M}}^h\right)^{-1} \tilde{\mathcal{Q}}^h \\ &= d(\pi) \times \mathsf{K}\mathcal{R}_0^T \times -\frac{\partial \left[\omega\right]_\times}{\partial \omega_j} \left[\mathsf{I}_{3\times3}\right| - \mathbf{t}] \, \tilde{\mathcal{Q}}. \end{aligned} \tag{A.24}$$

Concernant les dérivées par rapport aux paramètres de la translation, nous obtenons :

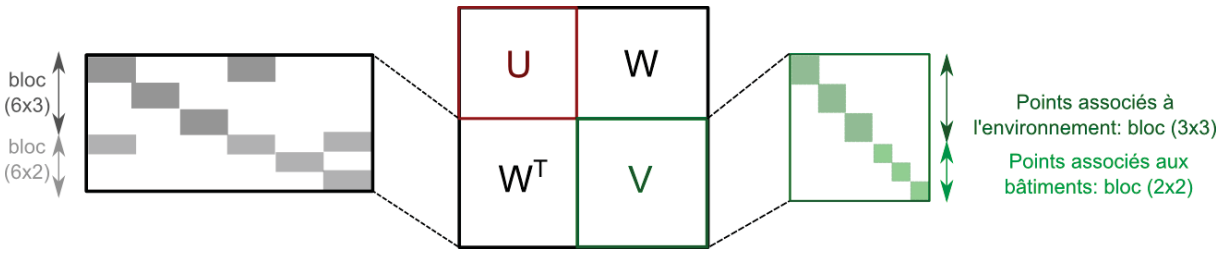


FIGURE A.2 – **Structure de la Hessienne de l'ajustement de faisceaux intégrant la contrainte des modèles des bâtiments**.

$$\begin{aligned}\frac{\partial \mathbf{g}}{\partial \mathbf{t}} &= d(\pi) \times \frac{\partial}{\partial \mathbf{t}} \left(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times 3} \right| - \mathbf{t} \right] (\tilde{\mathsf{M}}^h)^{-1} \tilde{\mathcal{Q}}^h \Big) \\ &= d(\pi) \times \mathsf{K}\mathcal{R}^T \times \frac{\partial}{\partial \mathbf{t}} \left(\left[\mathsf{0}_{3\times 3} \right| - \mathbf{t} \right] \tilde{\mathcal{Q}} \Big) \\ &= -1 \times d(\pi) \times \mathsf{K}\mathcal{R}^T. \end{aligned} \tag{A.25}$$

Dérivées par rapport aux paramètres des points 3D. Étant donné que la paramétrisation des points 3D associés aux bâtiments a changé, certaines modifications sont notables au niveau des dérivées par rapport aux paramètres des points 3D $\mathcal{Q}^h$:

$$\begin{aligned}\frac{\partial \mathbf{g}}{\partial \mathcal{Q}^h} &= d(\pi) \times \frac{\partial}{\partial \mathcal{Q}^h} \left(\mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times 3} \right| - \mathbf{t} \right] (\tilde{\mathsf{M}}^h)^{-1} \tilde{\mathcal{Q}}^h \Big) \\ &= d(\pi) \times \mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times 3} \right| - \mathbf{t} \right] (\tilde{\mathsf{M}}^h)^{-1} \end{aligned} \tag{A.26}$$

Nous rappelons que le paramètre Z^h n'est pas optimisé. Ainsi $\frac{\partial \mathbf{g}}{\partial Z^h} = 0$.

Construction du système creux. La fonction de coût bi-objective qui prend en compte à la fois les contraintes multi-vues et les contraintes géométriques fournies pas les modèles des bâtiments est donc donnée par :

$$g\left(\{\mathcal{R}_j, \mathbf{t}_j\}_{j=1}^N, \{\mathcal{Q}_i\}_{i\in\mathcal{E}} \left\{ \mathcal{Q}_i^{h_i} \right\}_{i\in\mathcal{M}} \right) = \sum_{i\in\mathcal{E}} \sum_{j\in\mathcal{D}_i} \|\mathbf{f}_{i,j}\|^2 + \sum_{i\in\mathcal{M}} \sum_{j\in\mathcal{D}_i} \|\mathbf{g}_{i,j}\|^2, \tag{A.27}$$

avec $\mathbf{g}_{i,j}$ est l'erreur de re-projection, définie dans l'équation A.23, du $i^{ème}$ point 3D observé par la $j^{ème}$ caméra.

Introduire une contrainte dure dans la fonction de coût en réduisant le nombre de degrés de liberté à optimiser permet de conserver la structure creuse du système à résoudre. La Hessienne $\mathsf{H}_g \approx \mathsf{J}_g^T \mathsf{J}_g$, où J_g est la Jacobienne associée à la fonction de coût g, a exactement la même structure que celle obtenue pour la fonction de coût standard. Les seules modifications à apporter sont au niveau des blocs V_i et $\mathsf{W}_{i,j}$. En effet, pour les points associés aux modèles des bâtiments, ces blocs sont de dimensions (2×2) et (6×2) au lieu de (3×3) et (6×3) comme le schématise la figure A.2.

A.2.2 Fonction de coût avec la contrainte du MET

Pour introduire la contrainte dure en altitude, une nouvelle paramétrisation de la caméra a été introduite dans la section 4.1. Cette paramétrisation se base sur l'hypothèse que la caméra, une fois embarquée dans le véhicule, garde une altitude fixe par rapport au plan de la route. Ceci a permis de réduire le nombre de degrés de liberté à optimiser pour chaque pose de la caméra. Nous rappelons que l'erreur de re-projection avec la nouvelle paramétrisation de la caméra est donnée par :

$$\mathbf{l} = \begin{pmatrix} q_1 - p_1/p_3 \\ q_2 - p_2/p_3 \end{pmatrix} = \mathbf{q} - \pi \left(\mathsf{K}(\mathcal{R}^k)^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k \tilde{\mathcal{Q}} \right), \tag{A.28}$$

avec $\mathcal{R}^k$ et $\mathbf{t}^k$ représentent les paramètres extrinsèques de la caméra exprimés dans le repère de plan de route correspondant. $\tilde{\mathsf{L}}^k$ est la matrice de passage du repère monde au repère du plan de la route $\mathbf{\Lambda}^k$ (voir figure 4.1).

Le même principe introduit pour les points 3D associés aux modèles est appliqué pour la contrainte dure apportée par le MET. En effet, l'optimisation se déroule dans le nouveau repère où l'altitude de la caméra est constante.

Dans ce qui suit nous détaillerons les dérivées de $\mathbf{l}$ par rapport aux paramètres de la caméra $(\mathcal{R}^k|\mathbf{t}^k)$ dans le repère de la route et les paramètres du point 3D $\mathcal{Q}$.

Dérivées par rapport aux paramètres de la caméra. En changeant la paramétrisation de la caméra, les dérivées par rapport aux paramètres de sa pose subissent certaines modifications. Concernant les dérivées par rapport aux paramètres de la rotation, elles sont calculées en utilisant un repère relatif au voisinage de la rotation courante dans le repère de la route $\mathcal{R}_0^k$, telle que $\mathcal{R}_0^k(\omega^k) = \mathcal{R}(\omega^k)\mathcal{R}_0^k$, avec $\omega^k = (\omega_1^k, \omega_2^k, \omega_3^k)^T$ contient les faibles angles d'Euler autour de $\mathcal{R}_0^k$ et :

$$\begin{aligned} \frac{\partial \mathbf{l}}{\partial \omega^k} &= d(\pi) \times \mathsf{K}(\mathcal{R}_0^k)^T \times -\frac{\partial \left[\omega^k\right]_\times}{\partial \omega_j^k} \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k \tilde{\mathcal{Q}} \\ &= d(\pi) \times \mathsf{K}(\mathcal{R}_0^k)^T \times -\frac{\partial \left[\omega^k\right]_\times}{\partial \omega_j^k} \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k \tilde{\mathcal{Q}}. \end{aligned} \tag{A.29}$$

Concernant les dérivées par rapport aux paramètres de la translation, nous obtenons :

$$\begin{aligned} \frac{\partial \mathbf{l}}{\partial \mathbf{t}^k} &= d(\pi) \times \frac{\partial}{\partial \mathbf{t}^k} \left(\mathsf{K}(\mathcal{R}^k)^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k \tilde{\mathcal{Q}} \right) \\ &= d(\pi) \times \mathsf{K}(\mathcal{R}^k)^T \times \frac{\partial}{\partial \mathbf{t}^k} \left(\left[\mathsf{0}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k \tilde{\mathcal{Q}} \right) \\ &= -1 \times d(\pi) \times \mathsf{K}(\mathcal{R}^k)^T. \end{aligned} \tag{A.30}$$

Notons que $\frac{\partial \mathbf{l}}{\partial (t^k)_z} = 0$.

Dérivées par rapport aux paramètres des points 3D. Les dérivées par rapport aux paramètres des points 3D restent inchangées en les comparant avec la fonction de coût standard.

$$\begin{aligned}\frac{\partial \mathbf{l}}{\partial \mathcal{Q}} &= d(\pi) \times \frac{\partial}{\partial \mathcal{Q}} \left(\mathsf{K}(\mathcal{R}^k)^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k \tilde{\mathcal{Q}} \right) \\ &= d(\pi) \times \mathsf{K}\mathcal{R}^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t} \right]\end{aligned} \tag{A.31}$$

Construction du système creux. La fonction de coût intégrant la contrainte dure en altitude s'écrit :

$$l\left(\left\{ \mathcal{R}_j^{k_j}, \mathbf{t}_j^{k_j} \right\}_{j=1}^{N}, \{\mathcal{Q}_i\}_{i=1}^{M} \right) = \sum_{i=1}^{M} \sum_{j \in \mathcal{D}_i} \|\mathbf{l}_{i,j}\|^2, \tag{A.32}$$

avec $\mathbf{l}_{i,j}$ est l'erreur de re-projection donnée par l'équation A.28. Comme c'est le cas pour la contrainte dure au niveau des points 3D, introduire une contrainte dure en altitude dans la fonction de coût permet de conserver la structure creuse du système à résoudre. Ainsi, la Hessienne $\mathsf{H}_l \approx \mathsf{J}_l^T \mathsf{J}_l$ (J_l étant la Jacobienne associée à la fonction de coût l) a exactement la même structure que celle obtenue pour la fonction de coût standard. Les seules modifications notables sont au niveau des blocs U_j et $\mathsf{W}_{i,j}$. En effet, ces blocs sont de dimensions respectives (5×5) et (5×3) au lieu de (6×6) et (6×3).

A.2.3 Fonction de coût avec la contrainte des modèles 3D des bâtiments et la contrainte du MET

Il est possible de fusionner les deux contraintes présentées précédemment dans une même fonction de coût (voir section 4.2). Nous supposons alors que la caméra est contrainte en permanence au MET. Le nuage de points quant à lui est segmenté en un ensemble de points associés aux modèles des bâtiments $\mathcal{M}$ et l'ensemble $\mathcal{E}$ de points associés au reste de l'environnement. Pour ce dernier ensemble, la fonction de coût optimisée est la fonction l détaillée dans la section A.2.2. En ce qui concerne l'ensemble de point $\mathcal{M}$, l'erreur de re-projection associée est définie par :

$$\mathbf{lg} = \begin{pmatrix} q_1 - p_1/p_3 \\ q_2 - p_2/p_3 \end{pmatrix} = \mathbf{q} - \pi \left(\mathsf{K}(\mathcal{R}^k)^T \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k (\tilde{\mathsf{M}}^h)^{-1} \tilde{\mathcal{Q}}^h \right). \tag{A.33}$$

Les dérivées de $\mathbf{lg}$ par rapport aux paramètres de la caméra $(\mathcal{R}^k|\mathbf{t}^k)$ dans le repère de la route et les paramètres du point 3D $\mathcal{Q}^h$ dans le repère du plan de la façade sont détaillées ci-dessous.

Dérivées par rapport aux paramètres de la caméra.

▷ Par rapport à la rotation :

$$\begin{aligned}\frac{\partial \mathbf{lg}}{\partial \omega^k} &= d(\pi) \times \mathsf{K}(\mathcal{R}_0^k)^T \times - \frac{\partial \left[\omega^k \right]_\times}{\partial \omega_j^k} \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k (\tilde{\mathsf{M}}^h)^{-1} \tilde{\mathcal{Q}}^h \\ &= d(\pi) \times \mathsf{K}(\mathcal{R}_0^k)^T \times - \frac{\partial \left[\omega^k \right]_\times}{\partial \omega_j^k} \left[\mathsf{I}_{3\times 3} | - \mathbf{t}^k \right] \tilde{\mathsf{L}}^k \tilde{\mathcal{Q}}.\end{aligned} \tag{A.34}$$

▷ Par rapport à la translation :

$$\begin{aligned}\frac{\partial \mathbf{lg}}{\partial \mathbf{t}^k} &= d(\pi) \times \frac{\partial}{\partial \mathbf{t}^k}\left(\mathsf{K}(\mathcal{R}^k)^T\left[\mathsf{I}_{3\times3}| - \mathbf{t}^k\right]\tilde{\mathsf{L}}^k(\tilde{\mathsf{M}}^h)^{-1}\tilde{\mathcal{Q}}^h\right)\\ &= d(\pi) \times \mathsf{K}(\mathcal{R}^k)^T \times \frac{\partial}{\partial \mathbf{t}^k}\left(\left[\mathsf{0}_{3\times3}| - \mathbf{t}^k\right]\tilde{\mathsf{L}}^k\tilde{\mathcal{Q}}\right)\\ &= -1 \times d(\pi) \times \mathsf{K}(\mathcal{R}^k)^T.\end{aligned} \tag{A.35}$$

Dérivées par rapport aux paramètres des points 3D.

$$\begin{aligned}\frac{\partial \mathbf{lg}}{\partial \mathcal{Q}^h} &= d(\pi) \times \frac{\partial}{\partial \mathcal{Q}^h}\left(\mathsf{K}(\mathcal{R}^k)^T\left[\mathsf{I}_{3\times3}| - \mathbf{t}^k\right]\tilde{\mathsf{L}}^k(\tilde{\mathsf{M}}^h)^{-1}\tilde{\mathcal{Q}}^h\right)\\ &= d(\pi) \times \mathsf{K}\mathcal{R}^T\left[\mathsf{I}_{3\times3}| - \mathbf{t}\right](\tilde{\mathsf{M}}^h)^{-1}.\end{aligned} \tag{A.36}$$

Construction du système creux. La fonction de coût intégrant la contrainte dure en altitude et celle des bâtiment s'écrit :

$$lg\left(\left\{\mathcal{R}_j^{k_j}, \mathbf{t}_j^{k_j}\right\}_{j=1}^{N}, \{\mathcal{Q}_i\}_{i\in\mathcal{E}}\left\{\mathcal{Q}_i^{h_i}\right\}_{i\in\mathcal{M}}\right) = \sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{D}_i}\|\mathbf{l}_{i,j}\|^2 + \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{D}_i}\|\mathbf{lg}_{i,j}\|^2, \tag{A.37}$$

avec $\mathbf{lg}_{i,j}$ est l'erreur de re-projection donnée par l'équation A.33.

En fusionnant les deux contraintes dures dans la fonction de coût, la structure de la Hessienne $\mathsf{H}_{lg} \approx \mathsf{J}_{lg}^T\mathsf{J}_{lg}$, J_{lg} étant la Jacobienne associée à lg, reste inchangée par rapport à celle obtenue avec la fonction de coût standard. Seules les dimensions des blocs U_j, V_i et $\mathsf{W}_{i,j}$ changent. En effet, les blocs U_j sont de dimensions (5×5). Pour les points associés aux modèles des bâtiments, les blocs V_i et $\mathsf{W}_{i,j}$ sont de dimensions (2×2) et (5×2) respectivement. La structure de H_{lg} est représentée dans la figure A.3.

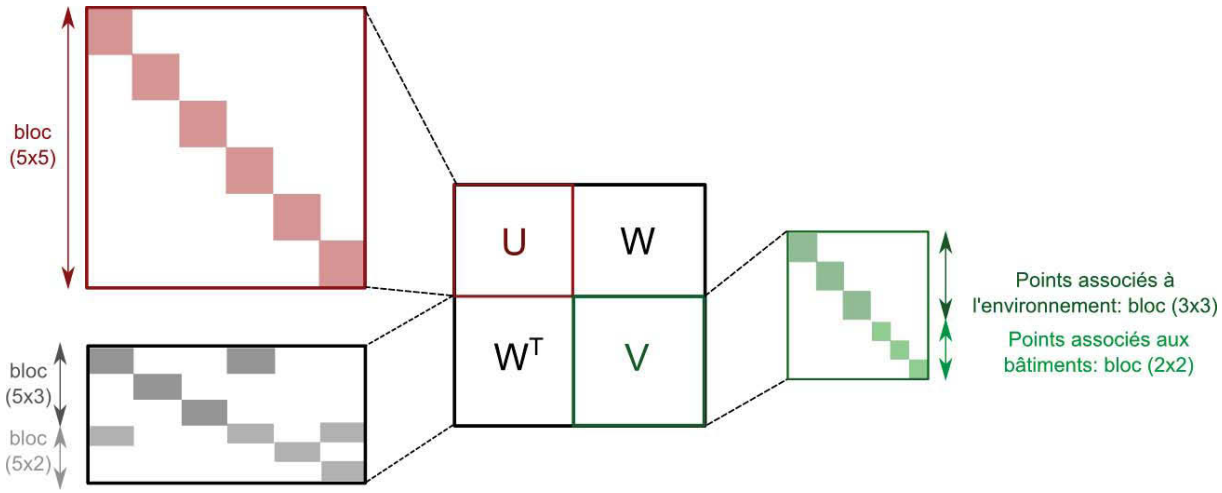


FIGURE A.3 – **Structure de la Hessienne de l'ajustement de faisceaux intégrant la contrainte des modèles des bâtiments et la contrainte dure en altitude.**

A.3 Ajustement de faisceaux basé sur une fonction de coût avec une contrainte d'inégalité

Dans cette section, nous présenterons dans un premier temps la fonction de coût avec la contrainte d'inégalité introduite par Lhuillier (2012) afin d'intégrer les données GPS. Ensuite nous détaillerons les fonctions de coût que nous avons proposées afin d'intégrer efficacement la contrainte en altitude fournie par le MET.

A.3.1 Fonction de coût contrainte aux données GPS

Contrairement aux données SIG, les mesures fournies par le GPS sont à la fois incertaines et peuvent contenir des données aberrantes. Pour garantir plus de robustesse face aux données aberrantes, Lhuillier (2012) a proposé un ajustement de faisceaux avec une contrainte d'inégalité qui permet de prendre en compte la fusion que si les mesures du GPS ne perturbent pas significativement la géométrie multi-vues (voir section 3.3).

Dans ce qui suit nous noterons $\kappa = \left(\left\{\alpha_j, \beta_j, \gamma_j, \mathbf{t}_j^T\right\}_{j=1}^{N}, \left\{X_i, Y_i, Z_i\right\}_{i=1}^{M}\right)^T$ le vecteur des paramètres à optimiser. Nous rappelons que $(\alpha_j, \beta_j, \gamma_j)$ sont les angles euler correspondants à la rotation $\mathcal{R}_j$ de la $j^{ème}$ caméra, $\mathbf{t}_j$ est sa translation. Enfin (X_i, Y_i, Z_i) sont les coordonnée du $i^{ème}$ point 3D. Toutes les données GPS sont stockées dans le vecteur $\mathbf{v} = (\mathbf{v}_1^T, .., \mathbf{v}_N^T)^T$, avec $\mathbf{v}_j = (x_j^{gps}, y_j^{gps})^T$.

La fonction de coût intégrant une contrainte d'inégalité $f_I(\kappa)$ est donnée par :

$$f_I(\kappa) = \frac{\omega}{e_t - f(\kappa)} + \|\mathsf{M}\kappa - \mathbf{v}\|^2, \tag{A.38}$$

avec

$$f(\kappa) = \|\mathbf{r}\|^2 = \sum_{i=1}^{i=M} \sum_{j \in \mathcal{D}_i} \|\mathbf{f}_{i,j}\|^2 \tag{A.39}$$

et $\mathsf{M} = (\mathsf{D}_{2N\times 6N} | 0_{2N\times 3M})$ tel que $\mathsf{D}_{2N\times 6N}$ est une matrice diagonale par bloc où chaque bloc de taille (2×6) est donné par

$$\mathsf{D}_j = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{A.40}$$

La matrice M permet de récupérer les positions dans le plan de la caméra.

Afin de minimiser la fonction de coût en question il est indispensable de déterminer la nouvelle matrice Hessienne H_I et le vecteur gradient $\mathbf{g}_I$ associés à $f_I(\kappa)$.

Calcul du gradient $\mathbf{g}_I$.

$$\begin{aligned}\mathbf{g}_I &= \frac{\partial f_I}{\partial \kappa} \\ &= \frac{\partial}{\partial \kappa}\left(\frac{\omega}{e_t - \|\mathbf{r}\|^2} + \|\mathsf{M}\kappa - \mathbf{v}\|^2\right) \\ &= \frac{2\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^2}\left(\frac{\partial \mathbf{r}}{\partial \kappa}\mathbf{r}\right) + 2\mathsf{M}^T\left(\mathsf{M}\kappa - \mathbf{v}\right) \\ &= \frac{2\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^2}\left(\mathsf{J}^T\mathbf{r}\right) + 2\mathsf{M}^T\left(\mathsf{M}\kappa - \mathbf{v}\right) \\ &= \frac{2\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^2}\mathbf{g} + 2\mathsf{M}^T\left(\mathsf{M}\kappa - \mathbf{v}\right)\end{aligned} \tag{A.41}$$

Calcul de la Hessienne H_I.

$$\begin{aligned}\mathsf{H}_I &= \frac{\partial^2 f_I}{\partial \kappa^2} \\ &= \frac{\partial}{\partial \kappa}\left(\frac{2\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^2}\left(\frac{\partial \mathbf{r}}{\partial \kappa}\mathbf{r}\right) + 2\mathsf{M}^T\left(\mathsf{M}\kappa - \mathbf{v}\right)\right) \\ &= \frac{2\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^3}\left[4\mathbf{g}\mathbf{g}^T + \left(e_t - \|\mathbf{r}\|^2\right)\mathsf{J}^T\mathsf{J}\right] + 2\mathsf{M}^T\mathsf{M}.\end{aligned} \tag{A.42}$$

Étant donné que le terme $\mathbf{g}\mathbf{g}^T$ n'est pas éparse, la Hessienne H_I a plutôt une structure dense contrairement à celle de H. Ainsi, le système ne peut pas être résolu de la même façon que précédemment (voir section A.1). Par conséquent, Lhuillier (2012) propose d'introduire la matrice $\bar{\mathsf{H}}$ et le vecteur $\bar{\mathbf{g}}$ tel que :

$$\mathsf{H}_I + \lambda diag(\mathsf{H}_I) = \bar{\mathsf{H}} + \bar{\mathbf{g}}\bar{\mathbf{g}}^T, \tag{A.43}$$

avec

$$\bar{\mathbf{g}} = \sqrt{\frac{8\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^3}}\mathbf{g} \tag{A.44}$$

et $\bar{\mathsf{H}}$ a la même structure creuse que H. Elle est donnée par :

$$\begin{aligned}\bar{\mathsf{H}} &= \frac{2\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^2}\mathsf{J}^T\mathsf{J} + 2\mathsf{M}^T\mathsf{M} + \lambda diag(\mathsf{H}_I) \\ &= \frac{2\omega}{\left(e_t - \|\mathbf{r}\|^2\right)^2}\mathsf{H} + 2\mathsf{M}^T\mathsf{M} + \lambda diag(\mathsf{H}_I)\end{aligned} \tag{A.45}$$

Par conséquent, le vecteur δ peut être calculé à partir de $\bar{\mathsf{H}}$ et $\bar{\mathbf{g}}$ comme suit :

$$\begin{aligned}\delta &= -\left(\bar{\mathsf{H}} + \bar{\mathbf{g}}\bar{\mathbf{g}}^T\right)^{-1}\mathbf{g}_I \\ &= -\left(\mathsf{I} - \frac{\bar{\mathsf{H}}^{-1}\bar{\mathbf{g}}\bar{\mathbf{g}}^T}{1 + \bar{\mathbf{g}}^T\bar{\mathsf{H}}^{-1}\bar{\mathbf{g}}^T}\right)\bar{\mathsf{H}}^{-1}\mathbf{g}_I \\ &= \mathbf{a} - \frac{\bar{\mathbf{g}}^T\mathbf{a}}{1 + \bar{\mathbf{g}}^T\mathbf{b}}\mathbf{b},\end{aligned} \tag{A.46}$$

avec $\mathbf{a} = -\bar{\mathsf{H}}^{-1}\mathbf{g}_I$ et $\mathbf{b} = \bar{\mathsf{H}}^{-1}\bar{\mathbf{g}}$. Ainsi, résoudre le système initial revient à estimer $\mathbf{a}$ et $\mathbf{b}$ tel que $\bar{\mathsf{H}}(\mathbf{a}, \mathbf{b}) = (-\mathbf{g}_I, \bar{\mathbf{g}})$. Ce système linéaire étant creux ($\bar{\mathsf{H}}$ est creuse), il peut être résolu de la même façon que le système construit dans la section A.1.

L'algorithme 8 résume les différentes étapes de l'algorithme de la minimisation de la fonction de coût f_I. Cet algorithme prend en entrée :

- ▷ L'erreur de re-projection $f = \|\mathbf{r}\|^2$.
- ▷ Les mesures du GPS $\mathbf{v}$.
- ▷ Le vecteur des paramètres à optimiser κ et qui minimise l'erreur de re-projection.
- ▷ Le nombre d'itérations maximal It_{max}.
- ▷ Le seuil e_t.

La sortie de l'algorithme est le vecteur κ tel que $f(\kappa) < e_t$ et $f_I(\kappa)$ a la plus petite valeur possible.

A.3.2 Intégration de la contrainte dure en altitude

Dans le chapitre 5, nous avons proposé deux façons pour introduire la contrainte en altitude dans l'ajustement de faisceaux avec la contrainte d'inégalité. La première façon consiste à introduire l'information de l'altitude au niveau du terme de la vision sous forme d'une contrainte dure. Ainsi le fonction de coût à optimiser est la suivante :

$$f_{1_I}(\eta) = \frac{\omega}{e_t - l(\eta)} + \|\mathsf{M}_1\eta - \vartheta\|^2, \tag{A.47}$$

avec $\eta = \left(\{\alpha_j^{k_j}, \beta_j^{k_j}, \gamma_j^{k_j}, (\mathbf{t}_j^{k_j})_x, (\mathbf{t}_j^{k_j})_y\}_{j=1}^N, (X_i, Y_i, Z_i)_{i=1}^M\right)^T$, où $(\alpha_j^{k_j}, \beta_j^{k_j}, \gamma_j^{k_j}, (\mathbf{t}_j^{k_j})_x, (\mathbf{t}_j^{k_j})_y)$ sont les paramètres de la pose de la caméra à optimiser exprimés dans le repère de plan de la route Λ^{k_j}. L'altitude de la caméra $(\mathbf{t}_j^{k_j})_z$ dans le repère de route n'est pas optimisée. $\vartheta = ((\mathbf{v}_1^{k_1})^T, .., (\mathbf{v}_N^{k_N})^T)^T$ où $\mathbf{v}_j^{k_j}$ sont les $j^{ème}$ données GPS exprimée dans le plan de la route associé à la caméra j.

$\mathsf{M}_1 = (\mathsf{D}_{1_{2N\times 5N}}|0_{2N\times 3M})$ tel que $\mathsf{D}_{1_{2N\times 5N}}$ est une matrice diagonale par bloc où chaque bloc de taille (2×5)

$$\mathsf{D}_{1_j} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{A.48}$$

Dans ce cas, le gradient est donné par :

$$\mathbf{g}_{1_I} = \frac{2\omega}{(e_t - l(\eta))^2}\mathbf{g}_l + 2\mathsf{M}_1{}^T(\mathsf{M}_1\eta - \vartheta) \tag{A.49}$$

et la Hessienne s'écrit comme suit :

$$\mathsf{H}_{1_I} = \frac{2\omega}{(e_t - l(\eta))^3}\left[4\mathbf{g}_l\mathbf{g}_l^T + (e_t - l(\eta))\,\mathsf{J}_l^T\mathsf{J}_l\right] + 2\mathsf{M}_1{}^T\mathsf{M}_1, \tag{A.50}$$

avec J_l et $\mathbf{g}_l$ respectivement la Jacobienne et le vecteur gradient associés à la fonction de coût de l'ajustement de faisceaux contraint aux MET defini en section A.2.2.

Pour minimiser cette nouvelle fonction de coût, le même algorithme détaillé dans la section A.3.1 est utilisé. Toutefois, à présent, l'optimisation des paramètres de la caméra se déroule plutôt dans le plan de la route où l'altitude est supposée fixe et donc non optimisée. Ainsi, comme nous l'avons expliqué dans la section A.2.2, les seules modifications notables concernant la structure du système à résoudre consistent à réduire les dimensions des blocs U_j et $\mathsf{W}_{i,j}$ qui sont désormais des blocs de dimensions (5×5) et (5×3) respectivement.

$err = f_I(\kappa)$;
$UpdateD = 1$;
$\lambda = 0.001$;
for $(It = 0; It < It_{max}; It++)$ **do**
  if $(UpdateD)$ **then**
    $UpdateD = 0$;
    $\mathbf{g} = \mathsf{J}^T\mathbf{r}$;
    $\mathsf{H} = \mathsf{J}^T\mathsf{J}$;
    $\mathbf{g}_I = \frac{2\omega}{(e_t - \|\mathbf{r}\|^2)^2}\mathbf{g} + 2\mathsf{M}^T(\mathsf{M}\kappa - \mathbf{v})$;
    $\mathsf{H}_1 = \frac{2\omega}{(e_t - \|\mathbf{r}\|^2)^2}\mathsf{H} + 2\mathsf{M}^T\mathsf{M}$;
    $\bar{\mathbf{g}} = \sqrt{\frac{8\omega}{(e_t - \|\mathbf{r}\|^2)^3}}\mathbf{g}$;
  end
  $\bar{\mathsf{H}} = \mathsf{H}_1 + \lambda diag(\mathsf{H}_1 + \bar{\mathbf{g}}\bar{\mathbf{g}}^T)$;
  Résoudre $\bar{\mathsf{H}}(\mathbf{a}, \mathbf{b}) = (-\mathbf{g}_I, \bar{\mathbf{g}})$;
  $\delta = \mathbf{a} - \frac{\bar{\mathbf{g}}^T\mathbf{a}}{1 + \bar{\mathbf{g}}^T\mathbf{b}}\mathbf{b}$
  if $(f(\kappa + \delta) \geq e_t)$ **then**
    $\lambda = 10\lambda$;
    continue ;
  end
  $err' = f_I(\kappa + \delta)$;
  if $(err' < err)$ **then**
    $\kappa = \kappa + \delta$;
    if $(0.999err < err')$ **then**
      break ;
    end
    $err = err'$;
    $UpdateD = 1$;
    $\lambda = \frac{\lambda}{10}$;
  end
  $\lambda = 10\lambda$;
end

Algorithme 8 : Étapes de l'optimisation pour l'ajustement de faisceaux avec la contrainte d'inégalité.

A.3.3 Intégration de la contrainte douce en altitude

A.3.3.1 Fusion des contraintes GPS avec la contrainte douce en altitude

Pour introduire l'information d'altitude dans la fonction de coût avec la contrainte GPS, une deuxième approche a été proposée (voir chapitre 5). Celle-ci intègre plutôt une contrainte douce en exploitant le second terme de la fonction de coût avec la contrainte d'inégalité. La fonction de coût résultante est la suivante :

$$f_{2_I}(\kappa) = \frac{\omega}{e_t - f(\kappa)} + \|(\mathsf{M}_2\kappa + \mathbf{m}_2) - \mathbf{w}\|^2, \tag{A.51}$$

tel que

- $\mathsf{M}_2 = (\mathsf{D}_{2_{3N\times 6N}}|0_{3N\times 3M})$ tel que $\mathsf{D}_{2_{3N\times 6N}}$ est une matrice diagonale par bloc où chaque bloc de taille (3×6) $\mathsf{D}_{2_j} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathsf{L}^{k_j}(3,1) & \mathsf{L}^{k_j}(3,2) & \mathsf{L}^{k_j}(3,3) \end{pmatrix}$. Nous rappelons que L^{k_j} est la matrice de passage du repère monde au repère de plan de la route $\mathbf{\Lambda}^{k_j}$

- $\mathbf{m}_2 = (\mathbf{n}_1^T ... \mathbf{n}_N^T)^T$ où $\mathbf{n}_j = \begin{pmatrix} 0 \\ 0 \\ \mathsf{L}^{k_j}(3,4) \end{pmatrix}$.

- $\mathbf{w} = (\mathbf{u}_1^T ... \mathbf{u}_N^T)^T$ où $\mathbf{u}_j = \begin{pmatrix} x_j^{gps} \\ y_j^{gps} \\ h \end{pmatrix}$, les données GPS sont exprimées dans le repère monde et h est l'altitude réelle de la caméra par rapport au plan de la route.

ce qui donne : $\mathsf{M}_2\kappa + \mathbf{m}_2 = ((\mathbf{t}_1)_x, (\mathbf{t}_1)_y, (\mathbf{t}_1^{k_1})_z, ..., (\mathbf{t}_N)_x, (\mathbf{t}_N)_y, (\mathbf{t}_N^{k_N})_z)^T$.

En adoptant cette approche, la structure du système à résoudre reste inchangé. Toutefois, quelques modifications sont notables dans le calcul du gradient $\mathbf{g}_{2_I}$ et la Hessienne H_{2_I}. En effet, $\mathbf{g}_{2_I}$ devient :

$$\mathbf{g}_{2_I} = \frac{2\omega}{(e_t - f(\kappa))^2}\mathbf{g} + 2\mathsf{M}_2{}^T(\mathsf{M}_2\kappa + \mathbf{m}_2 - \mathbf{w}) \tag{A.52}$$

et H_{2_I} est donnée par :

$$\mathsf{H}_{2_I} = \frac{2\omega}{(e_t - f(\kappa))^3}\left[4\mathbf{g}\mathbf{g}^T + (e_t - f(\kappa))\,\mathsf{J}^T\mathsf{J}\right] + 2\mathsf{M}_2{}^T\mathsf{M}_2. \tag{A.53}$$

A.3.3.2 Fusion des contraintes des modèles des bâtiments avec la contrainte douce en altitude

La contrainte douce en altitude peut également être fusionnée avec les contraintes géométriques apportées par les modèles 3D des bâtiments (chapitre 6). La fonction de coût résultante s'écrit de la manière suivante :

$$f_{3_I}(\varsigma) = \frac{\omega}{e_t - g(\varsigma)} + \|\mathsf{M}_3\varsigma + \mathbf{m} - \mathbf{h}\|^2, \tag{A.54}$$

avec $\varsigma = \left(\left\{\alpha_j, \beta_j, \gamma_j, \mathbf{t}_j^T\right\}_{j=1}^{N}, \{X_i, Y_i, Z_i\}_{i\in\mathcal{E}}, \left\{X_i^{h_i}, Y_i^{h_i}\right\}_{i\in\mathcal{M}}\right)^T$ où $\left\{X_i^{h_i}, Y_i^{h_i}\right\}$ sont les coordonnées non nulles du point $\mathcal{Q}_i^{h_i}$ dans le repère du plan du bâtiments $\mathbf{\Pi}^{h_i}$.

$\mathsf{M_3} = (\mathsf{D}_{3_{N\times 6N}}|\mathsf{0}_{N\times S_1}|\mathsf{0}_{N\times S_2})$, avec $S_1 = 3 \times \text{card}(\mathcal{E})$, $S_2 = 2 \times \text{card}(\mathcal{M})$ et tel que $\mathsf{D}_{3_{N\times 6N}}$ est une matrice diagonale par bloc où chaque bloc de taille (1×6) $\mathbf{d}_{3_j} = \begin{pmatrix}0 & 0 & 0 & \mathsf{L}^{k_j}(3,1) & \mathsf{L}^{k_j}(3,2) & \mathsf{L}^{k_j}(3,3)\end{pmatrix}$. Finalement, $\mathbf{m} = (\mathsf{L}^{k_1}(3,4)...\mathsf{L}^{k_N}(3,4))^T$.

La minimisation de cette fonction de coût se déroule de la même façon que précédemment. Elle s'appuie sur le calcul de

$$\mathbf{g}_{3_I} = \frac{2\omega}{\left(e_t - g\left(\varsigma\right)\right)^2}\mathbf{g}_g + 2\mathsf{M_3}^T\left(\mathsf{M_3}\varsigma + \mathbf{m} - \mathbf{h}\right) \tag{A.55}$$

et

$$\mathsf{H}_{3_I} = \frac{2\omega}{\left(e_t - g\left(\varsigma\right)\right)^3}\left[4\mathbf{g}_g\mathbf{g}_g^T + \left(e_t - g\left(\varsigma\right)\right)\mathsf{J}_g^T\mathsf{J}_g\right] + 2\mathsf{M_3}^T\mathsf{M_3}, \tag{A.56}$$

avec J_g et $\mathbf{g}_g$ respectivement la Jacobienne et le vecteur gradient associés à la fonction de coût de l'ajustement de faisceaux contraint aux plans défini dans la section A.2.1.

Bibliographie

S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications Association for Computing Machinery*, 54(10) : 105–112, 2011.

A. Angeli, S. Doncieux, J.A. Meyer, and D. Filliat. Visual topological slam and global localization. In *International Conference on Robotics and Automation*, 2009.

C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. In *International Symposium on Mixed and Augmented Reality*, 2009.

N. Ballas, B. Labbe, H. Le Borgne, and A. Shabou. Semantic indexing and instance search. In *Text Retrieval Conference, Video Retrieval Evaluation Workshop*, 2012.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. In *European Conference on Computer Vision*, 2006.

N. Bioret, G. Moreau, and M. Servieres. Geolocalisation en milieu urbain par appariement entre une collection d'images et un SIG 2D. *Ingenierie des systemes d'information*, 14(5) : 107–131, 2009.

G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, 2008a.

G.J Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, 2008b.

C. Cappelle, M. E. B. El Najjar, D. Pomorski, and F. Charpillet. Intelligent Geolocalization in Urban Areas Using Global Positioning Systems, Three-Dimensional Geographic Information Systems, and Vision. *Journal of Intelligent Transportation Systems*, 14(1) :3–12, 2010.

J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardos. The spmap : A probabilistic framework for simultaneous localization and map building. *Transactions on Robotics and Automation*, 15(5) :948–953, 1999.

J.A. Castellanos, J. Neira, and J.D. Tardós. Multisensor fusion for simultaneous localization and map building. *Transactions on Robotics and Automation*, 17(6) :908–914, 2001.

T.J. Cham, A. Ciptadi, W.C. Tan, M.T. Pham, and L.T. Chia. Estimating Camera Pose from a Single Urban Ground-View Omnidirectional Image and a 2D Building Outline Map. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

F. Chausse, J. Laneurit, and R. Chapuis. Localisation d'un vehicule sur une carte routiere precise. *Traitement de signal*, 23 :159–177, 2005.

J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-point ransac for ekf-based structure from motion. In *International Conference on Intelligent Robots and Systems*, 2009.

A.I. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *International Conference on Robotics and Automation*, 2007.

M. Cummins and P. Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proceedings of Robotics : Science and Systems*, 2009.

A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM : Real-time single camera SLAM. *Pattern Analysis and Machine Intelligence*, 26(6) :1052–1067, 2007.

M. Dawood, C. Cappelle, M. El Najjar El Badaoui, M. Khalil, and D. Pomorski. Vehicle geo-localization based on imm-ukf data fusion using a gps receiver, a video camera and a 3d city model. In *Intelligent Vehicles Symposium*, 2011.

G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *Transactions on Robotics and Automation*, 17(3) :229–241, 2001.

Z. Dong, G. Zhang, J. Jia, and H. Bao. Keyframe-based real-time camera tracking. In *International Conference on Computer Vision*, 2009.

E. Eade and T. Drummond. Scalable monocular slam. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

A.F. Elaksher, J.S. Bethel, and E.M. Mikhail. Reconstructing 3d building wireframes from multiple images. In *International Society for Photogrammetry and Remote Sensing*, 2002.

A. Eudes and M. Lhuillier. Error propagations for local bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

A. Eudes, M. Lhuiller, S. Naudet-Collette, and M. Dhome. Fast odometry integration in local bundle adjustment-based visual slam. In *International Conference on Pattern Recognition*, 2010a.

A. Eudes, S. Naudet-Collette, M. Lhuillier, and M. Dhome. Weighted local bundle adjustment and application to odometry and visual slam fusion. In *British Machine Vision Conference*, 2010b.

O. Faugeras. *Three-dimensional computer vision : a geometric viewpoint*. Massachusetts Institute of Technology Press, 1993.

J. M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *European Conference on Computer Vision*, 2010.

F. Fraundorfer, L. Heng, D. Honegger, G.H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor mav. In *International Conference on Intelligent Robots and Systems*, 2012.

D. Gamage and T. Drummond. Reduced dimensionality extended kalman filter for slam. In *British Machine Vision Conference*, 2013.

V. Gay-Bellile, P. Lothe, S. Bourgeois, E. Royer, and S. Naudet-Collette. Augmented reality in large environments : Application to aided navigation in urban context. In *International Symposium on Mixed and Augmented Reality*, 2010.

R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3) :331–356, 1994.

C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

R. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence*, 19(6) :580–593, 1997.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004.

K. Hideyuki, T. Takafumi, S. Tomokazu, and Y.Naokazu. Extrinsic camera parameter estimation using video images and gps considering gps positioning accuracy. In *International Conference on Pattern Recognition*, 2010.

P. Huber. *Robust Statistics*. Wiley, 1981.

A. Irschara, C. Zach, J.M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(Series D) :35–45, 1960.

R. S. Kaminsky, N. Snavely, S. M. Seitz, and R. Szeliski. Alignment of 3d point clouds to overhead images. In *IEEE Workshop on Internet Vision*, 2009.

Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera : robust ego-motion estimation and ground-layer detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

J. Kichun, C. Keounyup, and S. Myoungho. Gps-bias correction for precise localization of autonomous vehicles. In *Intelligent Vehicles Symposium*, 2013.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007.

K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In *International Symposium on Robotics Research*, 2007.

D. Larnaout, S. Bourgeois, V. Gay-Bellile, and M. Dhome. Towards bundle adjustment with gis constraints for online geo-localization of a vehicle in urban center. In *3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012.

D. Larnaout, V. Gay-Bellile, S. Bourgeois, and M. Dhome. Vehicle 6-dof localization based on slam constrained by gps and digital elevation model information. In *International Conference on Image Processing*, 2013a.

D. Larnaout, V. Gay-Bellile, S. Bourgeois, B. Labbe, and M. Dhome. Fast and automatic city-scale environment modeling for an accurate 6dof vehicle localization. In *International Symposium on Mixed and Augmented Reality*, 2013b.

D. Larnaout, V. Gay-Bellile, S. Bourgeois, B. Labbe, and M. Dhome. Driving in an augmented-city : From fast and automatic large scale environment modeling to on-line 6dof vehicle localization. In *International Conference on Virtual Reality Continuum and Its Applications*, 2013c.

J.-M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration ? In *European Conference on Computer Vision*, 1998.

T. Lemaire, C. Berger, I. Jung, and S. Lacroix. Vision-based slam : Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3) :343–364, 2007.

J. Leonard and P. Newman. Consistent, convergent, and constant-time slam. In *International Joint Conference on Artificial intelligence*, 2003.

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.

K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2) :164–168, 1944.

M. Lhuillier. Incremental fusion of structure-from-motion and gps using constrained bundle adjustments. *Pattern Analysis and Machine Intelligence*, 34(12) :2489–2495, 2012.

Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *European Conference on Computer Vision*, 2012.

B. Liang and N. Pears. Visual navigation using planar homographies. In *International Conference on Robotics and Automation*, 2002.

H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012a.

J. Lim, J.M. Frahml, and M. Pollefeys. Online environment mapping using metric-topological maps. *International Journal of Robotic Research*, 31(12) :1394–1408, 2012b.

P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Towards geographical referencing of monocular slam reconstruction using 3d city models : Application to real-time accurate vision-based localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *Pattern Analysis and Machine Intelligence*, 22(6) :610–622, 2000.

E. Malis and E. Marchand. Experiments with robust estimation techniques in real-time robot vision. In *International Conference on Intelligent Robots and Systems*, 2006.

J. Michot, A. Bartoli, and F. Gaspard. Bi-objective bundle adjustment with application to multi-sensor slam. In *3D Data Processing, Visualization and Transmission*, 2010.

K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, 2002.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, 2005.

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

D. Nister. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence*, 26(6) :756–777, 2004.

D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

M. Recky, A. Wendel, and F. Leberl. Facade segmentation in a multi-view scenario. In *3D Imaging, Modeling, Processing, Visualization and Transmission*, 2011.

E. Royer, M. Lhuillier, M. Dhome, and T. Chateau. Localization in urban environments : Monocular vision compared to a differential gps sensor. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling*, 2001.

O. Saurer, F. Fraundorfer, and M. Pollefeys. Omnitour : Semi-automatic generation of interactive virtual tours from omnidirectional video. In *3D Data Processing, Visualization and Transmission*, 2010.

D. Scaramuzza and R. Siegwart. Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5) :1015–1026, 2008.

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 *International Conference on Computer Vision*, 2009a.

D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *International Conference on Robotics and Automation*, 2009b.

D. Schleicher, L.M. Bergasa, M. Ocana, R. Barea, and E. Lopez. Real-time hierarchical gps aided visual slam on urban environments. In *International Conference on Robotics and Automation*, 2009.

C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000.

G. Silveira, E. Malis, and P. Rives. An efficient direct approach to visual slam. *IEEE Transactions on Robotics*, 24(5) :969 –979, 2008.

G. Simon. Tracking-by-synthesis using point features and pyramidal blurring. In *International Symposium on Mixed and Augmented Reality*, 2011.

N. Simond and P. Rives. Trajectography of an uncalibrated stereo rig in urban environments. In *International Conference on Intelligent Robots and Systems*, 2004.

B. Soheilian, O. Tournaire, N. Paparoditis, B. Vallet, and J.P. Papelard. Generation of an integrated 3d city model with visual landmarks for autonomous navigation in dense urban areas. In *Intelligent Vehicles Symposium*, 2013.

G. Sourimant, L. Morin, and K. Bouatouch. Gps, gis and video fusion for urban modeling. In *Computer Graphics International*, 2007.

Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Visual slam : Why filter ? *Image Vision Computing*, 30(2) :65–77, 2012.

C. Strecha, T. Pylvanainen, and P. Fua. Dynamic and scalable large scale image image reconstruction. In *3D Data Processing, Visualization and Transmission*, 2010.

M. Tamaazousti, V. Gay-Bellile, S. Naudet-Collette, S. Bourgeois, and M. Dhome. Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

A. Taneja, L. Ballan, and M. Pollefeys. Registration of spherical panoramic images with cadastral 3d models. In *3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012.

J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *International Conference on Intelligent Robots and Systems*, 2008.

J.D. Tardos, J. Neira, P.M. Newman, and J.J. Leonard. Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research*, 21 (4) :311–330, 2002.

G. Tong, Z. Wu, N. Weng, and W. Hou. An omni-directional vslam based on spherical camera model and 3d modeling. In *World Congress on Intelligent Control and Automation*, 2012.

P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3) :271–300, 1997.

B. Triggs. Autocalibration from planar scenes. In *European Conference on Computer Vision*, 1998.

C. Wang, T. Wang, J. Liang, Y. Chen, Y. Zhang, and C. Wang. Monocular visual slam for small uavs in gps-denied environments. In *IEEE International Conference on Robotics and Biomimetics*, 2012.

C.P. Wang, K. Wilson, and N. Snavely. Accurate georegistration of point clouds using geographic data. In *International Conference on 3D Vision*, 2013.

H. Wang, K. Yuan, W. Zou, and Q. Zhou. Visual odometry based on locally planar ground assumption. In *International Conference on Information Acquisition*, 2005.

Table des figures

1 Exemples de voitures autonomes 2
2 Exemples de systèmes d'aide à la navigation via la Réalité Augmentée 2

1.1 Projection perspective 9
1.2 Définition des angles roulis, tangage et lacet 10
1.3 Exemples de M-estimateurs 15
1.4 Géométrie épipolaire 16
1.5 Homographies 2D 18
1.6 Triangulation de points 3D 20
1.7 Erreur de reprojection 22
1.8 Structure de la matrice Hessienne de l'ajustement de faisceaux approchée $\mathsf{J}^T\mathsf{J}$. 24
1.9 Schéma du fonctionnement du SLAM de Mouragnon et al. (2006) 24
1.10 Les étapes de la méthodes de Mouragnon et al. (2006) 27
1.11 Exemples de reconstruction obtenues avec la méthode de Mouragnon et al. (2006) 28
1.12 Modélisation du bruit du GPS 29
1.13 Exemples de SIG 31
1.14 Modèles SIG utilisés 31

2.1 Différents graphes schématisant le problème du SLAM 37
2.2 Schéma de la fusion proposée par Wang et al. (2012) 40
2.3 Recalage d'une reconstruction SLAM sur un SIG 3D 42
2.4 Localisation basée vision exploitant des modèles 3D texturés 44

3.1 Illustrations des limitations de la méthode de segmentation 49
3.2 Illustrations des limitations de la méthode d'association point/plan 50
3.3 Fonction de coût proposée par Lothe et al. (2009) 51
3.4 Matrice de changement de repère M^{h_i} 52
3.5 Comparaison entre les résultats obtenus, hors ligne, avec l'ajustement de faisceaux global proposé par Tamaazousti et al. (2011) et celui introduit par Lothe et al. (2009) 56
3.6 Re-projection des modèles 3D des bâtiments sur un ensembles d'images extraite de la séquence de Versailles. 57

3.7 Résultat de la localisation en ligne obtenue avec l'approche de Tamaazousti et al. (2011) . . . 58
3.8 Illustrations de la séquence de synthèse utilisée. . . . 62
3.9 Évaluation de l'algorithme de Lhuillier (2012) sur la séquence de synthèse : évolution de l'erreur de la localisation *dans le plan* . . . 63
3.10 Résultat de la localisation en ligne du SLAM avec l'ajustement de faisceaux avec la contrainte d'inégalité . . . 63
3.11 Résultat de la localisation en ligne du SLAM avec l'ajustement de faisceaux avec la contrainte d'inégalité dans un milieu urbain dense . . . 64

4.1 Contrainte en altitude fournie par le MET . . . 73
4.2 Exemple de distribution de distance . . . 80
4.3 Distribution de distance pour chacune des façades observée . . . 81
4.4 Classification des plans des modèles 3D des bâtiments selon leurs secteurs angulaires . . . 81
4.5 Illustrations de la première séquence de synthèse . . . 83
4.6 Illustrations de la deuxième séquence de synthèse . . . 84
4.7 Apport de *la contrainte dure en altitude* : Cas d'une route plate . . . 87
4.8 Apport de *la contrainte dure en altitude* : Cas d'une route bosselée . . . 88
4.9 Évaluation de la méthode de segmentation proposée . . . 90
4.10 Évolution du seuil du rejet du M-estimateur . . . 91
4.11 Illustration de deuxième séquence de synthèse . . . 92
4.12 Robustesse face aux incertitudes des modèles 3D des bâtiments . . . 93
4.13 Séquences utilisées dans la ville de Versailles . . . 94
4.14 Localisation dans un milieu urbain en utilisant un SLAM contraint au MET et aux modèles 3D des bâtiments . . . 96

5.1 Définition du point G_j . . . 99
5.2 Schéma général du fonctionnement du processus d'optimisation incluant la contrainte GPS et la *contrainte dure en altitude* . . . 101
5.3 Schéma général du fonctionnement du processus d'optimisation incluant la contrainte GPS et la *contrainte douce en altitude* . . . 102
5.4 Illustrations de la séquence de synthèse utilisée . . . 103
5.5 Évaluation de la robustesse face aux imperfections du MET . . . 104
5.6 Localisation dans le contexte péri-urbain . . . 107
5.7 Résultat de la localisation en ligne du SLAM contraint aux données GPS et au MET dans un milieu urbain dense . . . 116

6.1 Illustrations de la séquence de synthèse utilisée. . . . 124
6.2 Illustrations des séquences réelles utilisées et les modèles SIG associés . . . 125
6.3 Concaténation de deux bases de données différentes dans le quartier de Versailles 126
6.4 Validation de notre processus de création de bases d'amers géo-référencées . . . 129
6.5 Exemples de re-projection des modèles des bâtiments . . . 130
6.6 Comparaison avec la méthode de Lothe et al. (2009) en utilisant la séquence de synthèse . . . 131
6.7 Comparaison avec la méthode de Lothe et al. (2009) sur des séquences réelles . 132
6.8 Exemples de re-projection des modèles des bâtiments sur des images extraites de la première séquence de Versailles . . . 133

7.1 Principe de fonctionnement du GPS différentiel . . . 136
7.2 Principe général de l'approche proposée . . . 137
7.3 Schéma global de la solution proposée . . . 138
7.4 Repère utilisé pour le calcul de la transformation T_1 . . . 139
7.5 Nombre de degrés de liberté de la transformation rigide T_1 en fonction des contraintes disponibles . . . 140
7.6 Évaluation du processus de correction du biais du GPS . . . 145
7.7 Vue de dessus des localisations obtenues avec ou sans corrections du biais du GPS dans le quartier Saclay . . . 147
7.8 Exemples de re-projection des modèles des bâtiments sur des images extraites de la séquence de Saclay . . . 148
7.9 Vue de dessus des localisations obtenues avec ou sans corrections du biais du GPS dans le quartier de Versailles . . . 149
7.10 Vue de dessus des localisations obtenues avec ou sans correction du biais du GPS dans le quartier de Saclay . . . 150
7.11 Exemples de re-projection des modèles des bâtiments sur des images extraites de la séquence de Versailles . . . 151
7.12 Vue de dessus des positions *dans le plan* des données GPS avant et après notre méthode de correction . . . 152
7.13 Évaluation sur les données de synthèse du processus de correction du biais du GPS avec la connaissance a priori des changements de biais . . . 154

8.1 Couplage du SLAM avec des données géo-référencées . . . 158
8.2 Comparaison entre la localisation instantanée de la fusion en ligne et celle obtenue avec la fusion hors ligne . . . 162
8.3 Illustrations de séquences d'apprentissage et de test utilisées . . . 163
8.4 Des exemples d'applications de Réalité Augmentée en utilisant la fusion en ligne des contraintes. . . . 163
8.5 Des exemples d'applications de Réalité Augmentée en exploitant des bases d'amers créées à travers la fusion hors ligne des contraintes. . . . 164
8.6 Principe de la contrainte en orientation . . . 170

A.1 Structure de la matrice Hessienne de l'ajustement de faisceaux . . . 176
A.2 Structure de la Hessienne de l'ajustement de faisceaux intégrant la contrainte des modèles des bâtiments . . . 178
A.3 Structure de la Hessienne de l'ajustement de faisceaux intégrant la contrainte des modèles des bâtiments et la contrainte dure en altitude . . . 181

Liste des tableaux

4.1 Distance médiane séparant les points 3D et leurs façades dans chaque secteur angulaire observé par la caméra courante 80

5.1 Tableau récapitulatif des erreurs de localisation obtenues sur la séquence de synthèse 105
5.2 Tableau récapitulatif des erreurs de localisation *dans le plan* pour la séquence réelle enregistrée en Saint Quentin en Yveline 106

6.1 Comparaison de notre approche de création de base d'amers avec l'approche de Lothe et al. (2009) 127

7.1 Précision des localisations obtenues avec ou sans la correction du biais du GPS dans le quartier de Versailles 152

Liste des Algorithmes

1 Différentes étapes du processus d'optimisation du SLAM contraint aux modèles des bâtiments proposé par Tamaazousti et al. (2011) 55
2 Ajustement de faisceaux avec contrainte d'inégalité 61
3 Différentes étapes de l'ajustement de faisceaux *avec une contrainte dure en altitude*. 74
4 Différentes étapes de l'ajustement de faisceaux contraint au MET et aux modèles 3D des bâtiments 77
5 L'algorithme de segmentation du nuage de points proposé 82
6 Processus d'optimisation avec la contrainte d'inégalité intégrant les informations du MET et des modèles 3D des bâtiments 123
7 Différentes étapes du processus d'optimisation associé au SLAM contraint au MET et aux données du GPS corrigées 143
8 Étapes d'optimisation pour l'ajustement de faisceaux avec la contrainte d'inégalité 185

Table des matières

Introduction **1**

1 Notions de base et données utilisées **7**
1.1 Caméras perspectives et géométrie associée . . . 7
1.1.1 Géométrie projective . . . 7
1.1.2 Représentation d'une caméra perspective . . . 8
1.1.2.1 Projection perspective . . . 8
1.1.2.2 Notion de rétroprojection . . . 11
1.2 Optimisation numérique . . . 12
1.2.1 Moindres carrés . . . 12
1.2.2 Méthodes de résolution linéaires . . . 12
1.2.3 Méthodes de résolution non-linéaires . . . 13
1.2.4 Optimisation robuste . . . 14
1.3 Localisation et reconstruction 3D par vision . . . 15
1.3.1 Géométrie épipolaire . . . 16
1.3.2 Calcul de la géométrie de l'environnement . . . 18
1.3.2.1 Poses de caméras et déplacement relatif . . . 19
1.3.2.2 Calcul du déplacement relatif par associations 2D/2D . . . 19
1.3.2.3 Calcul de la structure de l'environnement . . . 20
1.3.2.4 Calcul de pose par associations 2D/3D . . . 21
1.3.2.5 Erreur de reprojection et ajustement de faisceaux . . . 21
1.4 Algorithmes et données utilisés . . . 23
1.4.1 Algorithme de localisation et cartographie simultanées . . . 24
1.4.1.1 Traitements 2D . . . 25
1.4.1.2 Traitements 3D . . . 26
1.4.2 Les données d'entrée . . . 26
1.4.2.1 Les données GPS . . . 26
1.4.2.2 Les modèles 3D urbains . . . 30

2 Etat de l'art **33**
2.1 Localisation basée vision sans a priori . . . 33

2.1.1 Odométrie visuelle . . . 34
2.1.2 SLAM visuel . . . 34
2.1.2.1 SLAM par filtrage . . . 35
2.1.2.2 SLAM basé image clé . . . 35
2.1.2.3 Comparaison entre SLAM par filtrage et SLAM basé image clé 36
2.1.3 Limites de ce type de méthodes . . . 38
2.1.3.1 Localisation non géo-référencée . . . 38
2.1.3.2 Dérives sur les longues trajectoires . . . 38
2.2 Localisation basée vision avec ajout d'informations additionnelles . . . 39
2.2.1 Intégration des capteurs supplémentaires . . . 39
2.2.2 Exploitation de la connaissance a priori de l'environnement . . . 41
2.2.2.1 Utilisation des modèles géométriques . . . 41
2.2.2.2 Utilisation des modèles photo-géométriques . . . 43
2.3 Bilan . . . 45

3 État de l'art : Ajustement de faisceaux contraint 47
3.1 Introduction . . . 47
3.2 SLAM contraint à un modèle géométrique de la ville : Contraindre la reconstruction . . . 48
3.2.1 Établissement de la contrainte aux bâtiments : Segmentation du nuage de points et association point/plan . . . 48
3.2.2 Formalisation de la contrainte apportée par les modèles 3D des bâtiments 50
3.2.3 Optimisation robuste vis à vis des contraintes utilisées . . . 53
3.2.4 Limitations . . . 54
3.3 SLAM contraint aux données GPS : Contraindre la trajectoire . . . 59
3.3.1 Établissement de la contrainte aux données GPS : association image clé/mesure GPS . . . 59
3.3.2 Formalisation de la contrainte apportée par les données GPS . . . 59
3.3.3 Optimisation vis à vis des contraintes utilisées . . . 60
3.3.4 Limitations . . . 61
3.4 Bilan . . . 62

I Intégration des contraintes fournies par le MET dans un SLAM contraint pour une géo-localisation plus précise sur les 6DoF de la caméra 65

Présentation des méthodes proposées 69

4 SLAM contraint au MET et aux modèles 3D des bâtiments pour une localisation en ligne 71
4.1 SLAM contraint au MET . . . 71
4.1.1 Établissement de la contrainte en altitude . . . 72
4.1.2 Formalisation de la contrainte en altitude apportée par le MET . . . 73
4.1.3 Optimisation robuste vis-à-vis des contraintes utilisées . . . 74
4.2 SLAM contraint au MET et aux modèles 3D des bâtiments . . . 75
4.3 Segmentation du nuage de points . . . 78
4.4 Évaluation expérimentale . . . 82

4.4.1 Évaluation de la contrainte dure en altitude . . . 82
4.4.2 Évaluation de la méthode de segmentation . . . 86
4.4.3 Évaluation de la robustesse face aux incertitudes des modèles 3D des bâtiments . . . 89
4.4.4 Évaluation de notre algorithme dans des conditions réelles . . . 91
4.5 Conclusion et perspectives . . . 95

5 SLAM contraint aux données GPS et au MET 97
5.1 Introduction . . . 97
5.2 Fusion de la contrainte aux données GPS avec *une contrainte dure en altitude* . 98
5.3 Fusion de la contrainte aux données GPS avec *une contrainte douce en altitude* 100
5.4 Évaluation expérimentale . . . 102
5.4.1 Évaluation de la robustesse des solutions proposées . . . 102
5.4.2 Localisation dans un milieu péri-urbain . . . 106
5.5 Conclusion et perspectives . . . 108

Bilan 109

II Fusion des contraintes apportées par le GPS, le MET et les modèles des bâtiments pour une localisation précise d'une caméra dans un milieu urbain : Application à la Réalité Augmentée 111

Présentation des méthodes proposées 115

6 Fusion hors ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Création d'une base d'amers géo-référencée précise 117
6.1 Introduction . . . 117
6.2 Positionnement et principe de notre approche . . . 118
6.3 Création d'une base d'amers 3D géo-référencée . . . 119
6.3.1 Base d'amers géo-référencée initiale . . . 119
6.3.2 Raffinement de la base d'amers géo-référencée initiale . . . 120
6.3.2.1 Recalage dans le plan utilisant les modèles 3D des bâtiments 121
6.3.2.2 Fusion des contraintes des modèles 3D des bâtiments et du MET . . . 121
6.4 Évaluation expérimentale . . . 122
6.4.1 Séquences de tests utilisées . . . 122
6.4.2 Évaluation de l'ensemble de notre solution pour la création d'une base d'amers géo-référencés . . . 124
6.4.3 Comparaison avec l'approche de Lothe et al. (2009) . . . 126
6.5 Conclusion et perspectives . . . 128

7 Fusion en ligne des contraintes fournies par le GPS, MET et les modèles 3D des bâtiments : Correction du biais du GPS 135
7.1 Positionnement et principe de notre approche . . . 135
7.2 GPS différentiel basé sur les modèles 3D des bâtiments . . . 137
7.2.1 Présentation générale . . . 137

7.2.2 Estimation de l'erreur *dans le plan* de la fusion du SLAM avec le GPS et le MET . . . 138
7.2.2.1 Estimation de l'erreur *dans le plan* de la fusion dans le cas simple . . . 138
7.2.2.2 Gestions des degrés de liberté mal contraints . . . 140
7.2.3 Correction du biais du GPS . . . 140
7.3 SLAM contraint au MET et aux données du GPS corrigées . . . 142
7.4 Évaluation expérimentale . . . 143
7.4.1 Évaluation sur les données de synthèse . . . 144
7.4.2 Évaluation sur les données réelles . . . 146
7.4.2.1 Protocole expérimental et séquences utilisées . . . 146
7.4.2.2 Résultats . . . 146
7.5 Conclusion et perspectives . . . 152

8 Applications de Réalité Augmentée en milieu urbain 155
8.1 Introduction . . . 155
8.2 Approche proposée . . . 157
8.3 Navigation en zone disposant d'une base d'amers . . . 157
8.4 Navigation en zone dépourvue de base d'amers et mise à jour de la base . . . 159
8.5 Transition entre zone avec base d'amers vers une zone sans base d'amers valide 160
8.6 Évaluation expérimentale . . . 160
8.6.1 Comparaison de la précision de la navigation en zone disposant d'une base d'amers avec la navigation en zone dépourvue d'une base d'amers 160
8.6.2 Aide à la navigation en Réalité Augmentée . . . 161
8.7 Discussion . . . 161

Conclusion 167

Annexes 171

A Ajustement de faisceaux contraint 173
A.1 Ajustement de faisceaux basé sur la fonction de coût standard . . . 173
A.2 Ajustement de faisceaux basé sur une fonction de coût avec une contrainte dure 177
A.2.1 Fonction de coût avec la contrainte des modèles 3D bâtiments . . . 177
A.2.2 Fonction de coût avec la contrainte du MET . . . 179
A.2.3 Fonction de coût avec la contrainte des modèles 3D des bâtiments et la contrainte du MET . . . 180
A.3 Ajustement de faisceaux basé sur une fonction de coût avec une contrainte d'inégalité . . . 182
A.3.1 Fonction de coût contrainte aux données GPS . . . 182
A.3.2 Intégration de la contrainte dure en altitude . . . 184
A.3.3 Intégration de la contrainte douce en altitude . . . 186
A.3.3.1 Fusion des contraintes GPS avec la contrainte douce en altitude 186
A.3.3.2 Fusion des contraintes des modèles des bâtiments avec la contrainte douce en altitude . . . 186

Bibliographie 188

Table des figures **197**

Liste des tableaux **199**

Liste des algorithmes **201**

Table des matières **207**