N° d'ordre : D.U : 2060
EDSPIC : 492

UNIVERSITÉ BLAISE PASCAL - CLERMONT-FERRAND II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée par :
Pierre LOTHE

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

en vue de l'obtention du grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : Vision pour la Robotique

Localisation et cartographie simultanées par vision monoculaire contraintes par un SIG :

Application à la géolocalisation d'un véhicule

Soutenue publiquement le 8 octobre 2010 devant le jury :

M. Steve BOURGEOIS	Examinateur
M. François CHAUMETTE	Rapporteur
M. Michel DHOME	Directeur de thèse
M. Nicolas PAPARODITIS	Président du jury
M. Patrick RIVES	Rapporteur
M. Eric ROYER	Examinateur
M. Peter STURM	Rapporteur

N° d'ordre : D.U : 2060
EDSPIC : 492

UNIVERSITÉ BLAISE PASCAL - CLERMONT-FERRAND II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée par :
Pierre LOTHE

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

en vue de l'obtention du grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : Vision pour la Robotique

Localisation et cartographie simultanées par vision monoculaire contraintes par un SIG :

Application à la géolocalisation d'un véhicule

Soutenue publiquement le 8 octobre 2010 devant le jury :

M. Steve BOURGEOIS	Examinateur
M. François CHAUMETTE	Rapporteur
M. Michel DHOME	Directeur de thèse
M. Nicolas PAPARODITIS	Président du jury
M. Patrick RIVES	Rapporteur
M. Eric ROYER	Examinateur
M. Peter STURM	Rapporteur

Remerciements

La rédaction des remerciements est un exercice difficile mais qui me semble néanmoins indispensable. En effet, si la soutenance et le mémoire mettent principalement en avant le doctorant, il m'apparait aujourd'hui comme une évidence que ce sont avant tout de nombreuses rencontres humaines et scientifiques qui ont transformé ces trois années en la formidable expérience que j'ai vécue. C'est pourquoi j'aimerais adresser, en quelques mots, mes plus chaleureux remerciements à toutes les personnes qui ont partagé cette expérience avec moi.

Tout d'abord, je tiens à remercier les personnes qui m'ont donné l'envie et qui m'ont amené à me lancer dans cette aventure. Je pense en premier lieu à Vincent Charvillat qui a su, durant mon cursus à l'ENSEEIHT, me transmettre sa passion pour la recherche et l'innovation. Merci également à François Gaspard et Michel Dhome pour la confiance qu'ils m'ont accordée en me proposant de réaliser cette thèse au sein du laboratoire LVIC du CEA LIST et en cotutelle avec l'équipe ComSee du LASMEA.

Avec le recul, je réalise que l'équipe d'encadrement influe indéniablement sur le déroulement de la thèse et la façon dont cette expérience est vécue par le doctorant. J'ai eu pour ma part la chance d'être encadré par des personnes qui vous incitent naturellement à donner le meilleur de vous-même. En premier, je remercie Steve Bourgeois, mon « super-encadrant » au quotidien au sein du LVIC, pour sa bonne humeur permanente, son opiniâtreté et pour avoir su être présent dans les moments heureux de la thèse comme dans les moments plus compliqués. Je remercie également Michel Dhome, mon directeur de thèse, et Eric Royer, mon co-encadrant au LASMEA, pour leurs innombrables conseils avisés et pour avoir su se rendre autant disponibles malgré leurs agendas surchargés et la distance qui nous séparait.

Au sein du laboratoire LVIC, j'ai eu la chance de travailler avec de nombreuses personnes qui m'ont, chacune à leur façon, accompagné au cours de ces trois années. S'il m'est impossible de lister l'ensemble des membres du laboratoire, je les remercie tous pour les nombreuses discussions que nous avons eues, pour les heures passées à résoudre des problèmes scientifiques et informatiques, pour les parties de foot endiablées et pour la bonne ambiance générale qui a rendu le quotidien tellement agréable. Au sein de l'équipe, quelques personnes ont eu un rôle particulier. Aussi, je remercie Fabien Dekeyser et Sylvie Naudet-Collette pour avoir suivi mes travaux au cours de leur période respective. Merci à Hanna Martinsson, Pierre-Emmanuel Viel, Laurent Lucat et Stevens Lion pour avoir partagé mon bureau et donc par la même occasion mes humeurs. Merci aussi à Vincent Gay-Bellile, véritable bibliographie vivante, pour sa disponibilité et pour ses nombreux conseils. Merci également à Alexandre Eudes et Julien Michot pour

les discussions enflammées mais très agréables que nous avons pu avoir autour de nos travaux de recherche. Enfin, j'aimerais remercier nos secrétaires, Frédéric Descreaux, Elodie Duret et Annie Straboni pour avoir su m'accompagner dans les nombreuses démarches administratives durant mes années au CEA et ce, toujours avec le sourire.

La rédaction du mémoire puis la soutenance viennent alors clore ces trois années de travail. Je tiens ici à remercier chaleureusement l'ensemble du jury de thèse qui a su transformer cette épreuve tant redoutée en une journée particulièrement agréable. Je remercie en particulier François Chaumette, Patrick Rives et Peter Sturm pour avoir accepté de rapporter mes travaux et pour les échanges que nous avons pu avoir au cours de cette journée. Je remercie également Nicolas Paparoditis pour avoir présidé ma soutenance et pour les conseils et encouragements qu'il m'a prodigués au cours de différents congrès.

J'aimerais maintenant adresser des remerciements plus personnels mais non moins essentiels. En effet, si la thèse est une aventure professionnelle, elle s'intègre évidemment au sein d'un parcours personnel. Je remercie tout d'abord mes parents, Catherine et Bruno, pour avoir accepté tous mes choix, pour avoir toujours tout fait pour m'épauler mais aussi pour l'amour avec lequel ils m'entourent depuis maintenant vingt-six ans. Je tiens aussi à remercier ma soeur, Agnès, pour son soutien et la grande complicité qui nous a toujours unis.

J'aimerais également remercier les parents d'Aline, Nadine et Michel, ainsi que sa soeur, Odile, qui m'ont soutenu en particulier pendant ces trois dernières années et qui ont su accepter les contraintes imposées par la thèse. Enfin, mes plus profonds remerciements vont vers Aline, ma chère et tendre, pour le soutien et le réconfort qu'elle m'a apportés, pour la patience et la compréhension dont elle a fait part durant ces trois dernières années et plus encore pour le bonheur que j'ai à vivre à ses côtés depuis notre rencontre.

Résumé

Les travaux réalisés au cours de cette thèse s'inscrivent dans les problématiques de localisation d'un véhicule par vision. Nous nous plaçons en particulier dans le cas de parcours sur de longues distances, c'est à dire plusieurs kilomètres. Les méthodes actuelles de localisation et cartographie simultanées souffrent de problèmes de dérives qui les rendent difficilement exploitables après plusieurs centaines de mètres. Nous proposons dans ce mémoire de pallier ces limites en exploitant une connaissance *a priori* sur la géométrie de l'environnement parcouru. Cette information est extraite d'un Système d'Information Géographique. En particulier, les travaux réalisés se basent sur les modèles 3D des bâtiments des villes et sur une carte de la route.

Dans la première partie de ce mémoire, nous proposons une approche permettant de corriger hors ligne une reconstruction SLAM en exploitant la connaissance d'un modèle 3D simple de l'environnement. Cette correction s'applique en deux étapes. En premier lieu, un recalage non-rigide entre le nuage de points reconstruit et le modèle 3D est effectué de sorte à retrouver la cohérence globale de la reconstruction. Dans le but de raffiner le nuage de points obtenu, un ajustement de faisceaux contraint par le SIG est alors effectué sur l'ensemble de la reconstruction. La particularité de cet ajustement de faisceaux est qu'il prend implicitement en compte les contraintes géométriques apportées par le modèle 3D. La reconstruction ainsi corrigée est alors utilisée en tant que base de données pour la relocalisation en ligne d'une caméra mobile. La précision de relocalisation obtenue est en particulier suffisante pour les applications de réalité augmentée.

Dans la deuxième partie de ce mémoire, nous détaillons une solution permettant de corriger en ligne la reconstruction SLAM. Pour cela, les contraintes géométriques apportées par le SIG sont exploitées au fur et à mesure de la trajectoire du véhicule. Nous montrons tout d'abord que la connaissance de la position relative de la caméra par rapport à la route permet de corriger de façon robuste la dérive de facteur d'échelle. De plus, lorsque les contraintes géométriques sont suffisantes, la reconstruction SLAM réalisée jusqu'à l'instant courant est recalée sur le SIG. Cela permet de corriger ponctuellement la dérive observée sur la position courante de la caméra. Le processus complet permet dès lors de localiser le véhicule avec une précision semblable à celle d'un système GPS sur des trajectoires de plusieurs kilomètres.

Les deux méthodes proposées ont été testées à la fois sur des séquences de synthèse et réelles. Des résultats qualitatifs et quantitatifs sont présentés tout au long de ce mémoire.

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

Abstract

This thesis deals with the vision based geolocalisation of a vehicle. In particular, the problem of localisation on large sequences, *i.e.* several kilometers, is studied. In this context, state of the art Simultaneous Localisation and Mapping systems suffer from drift. In consequence, existing SLAM methods can not provide accurate localisation of the camera after several hundred meters. Thus, we propose in this thesis to avoid the drift phenomenon by exploiting a simple knowledge about the geometry of the environment. This information is provided by a Geographical Information System. In particular, our work is based on coarse 3D city models and road maps.

In the first part, we propose an offline two steps correction of SLAM reconstructions based on a 3D city model of the area. First, the reconstructed 3D point cloud and this 3D city model are aligned through a non-rigid transformation. This step allows the SLAM reconstruction to regain its global consistency. Then, a bundle adjustment constrained with the GIS is applied on the entire reconstruction to refine its geometry. The innovation of this bundle adjustment is that it takes into account the geometrical constraints provided by the 3D city model in a single term. The obtained 3D point cloud can then be considered as a feature landmark database. Finally, this database is used to localise a moving camera in real-time. In pratice, the precision of the obtained localisation is sufficient for augmented reality applications.

In the second part of this manuscript, we present a solution which makes possible the online correction of a SLAM reconstruction. The GIS geometrical constraints are exploited over the vehicle trajectory. First, we show that the scale factor drift can be robustly corrected thanks to the knowledge of the ground plane equation. Furthermore, the current SLAM reconstruction is fitted onto the GIS when the geometrical constraints are sufficient. It punctually ensures the correction of the current camera position. The entire process allows the geolocalisation of a vehicle on several kilometers. The obtained precision is close to GPS.

The two proposed solutions have been validated of both synthetic and real sequences. Quantitative and qualitative experiments are presented over this manuscript.

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

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
$\tilde{\mathcal{Q}}$	Coordonnées homogènes du point 3D
$\mathbf{d}$	Droite de l'espace
$\mathbf{\Pi}$	Plan de l'espace
R	Matrice de rotation
$\mathbf{t}$	Vecteur de translation
$\mathcal{S}$	Transformation géométrique 3D
S	Matrice de transformation associée à $\mathcal{S}$

Caméras et reconstruction 3D

$\mathcal{C}$	Caméra
$\mathcal{I}$	Image capturée par la caméra
$\tilde{\mathsf{P}}$	Matrice de projection
K	Matrice de calibrage
C^E	Paramètres extrinsèques
F	Matrice fondamentale
E	Matrice essentielle
$\mathcal{C}_j$	j^e caméra reconstruite
$\mathcal{Q}^i$	i^e point 3D reconstruit
$\mathbf{q}^i_j$	Observation du i^e point 3D par la j^e caméra
$\mathcal{B}$	Fragment de reconstruction 3D

Acronymes

ICP	Iterative Closest Point
MAD	Median Absolute Deviation (Valeur absolue des écarts à la médiane)
SfM	Structure from Motion (Stéréo-mouvement)
SIG	Système d'Information Géographique
SLAM	Simultaneous Localization and Mapping (Localisation et cartographie simultanées)
ST-CBA	Single-Term Constrained Bundle Adjustment (Ajustement de faisceaux contraint à un seul terme)

Sommaire

Introduction **1**

1 Etat de l'art **7**
1.1 Localisation monoculaire en environnement inconnu 7
1.2 Environnements partiellement connus : exploitation des Systèmes d'Information Géographique 11

2 Notions de base et données utilisées **17**
2.1 Géométrie projective 17
2.2 Caméras perspectives et géométrie associée 18
2.3 Géométrie multi-vue 22
2.4 Cas d'une scène plane 28
2.5 Optimisation numérique 30
2.6 Algorithmes et données utilisés 35

I Création d'une base d'amers visuels et relocalisation d'une caméra mobile **43**

Présentation de la méthode **47**

3 ICP non-rigide **51**
3.1 Méthodes d'alignement 3D 51
3.2 Espace de transformations utilisé 52
3.3 Recherche de l'alignement optimal 56
3.4 Discussion 59

4 Ajustements de faisceaux contraints par un SIG **61**
4.1 Méthodes classiques et limites 61
4.2 Approche proposée : ST-CBA 63

5 Résultats expérimentaux **69**
5.1 Evaluation quantitative sur des données de synthèse . . . 69
5.2 Evaluation qualitative sur des données réelles . . . 77
5.3 Discussion . . . 82

6 Relocalisation et réalité augmentée **87**
6.1 Présentation de l'application visée . . . 87
6.2 Localisation absolue dans la base d'amers . . . 87
6.3 Vers une application d'aide à la navigation . . . 89
6.4 Discussion . . . 92

II Vers la correction en ligne d'une reconstruction SLAM **95**

Présentation de la méthode **99**

7 Méthode de correction du facteur d'échelle **103**
7.1 Etat de l'art . . . 103
7.2 Contraintes disponibles et positionnement de nos travaux . . . 105
7.3 Outils nécessaires à l'estimation du facteur d'échelle . . . 106
7.4 Méthodes d'estimation du facteur d'échelle proposées . . . 115
7.5 Validation expérimentale de l'estimation du facteur d'échelle . . . 118
7.6 Intégration du facteur d'échelle dans la méthode SLAM . . . 124
7.7 Résultats expérimentaux . . . 129
7.8 Discussion . . . 133

8 Méthode de correction de l'accumulation d'erreur **135**
8.1 Objectif de l'étude . . . 135
8.2 Alignement et contraintes géométriques exploitables . . . 136
8.3 Estimation de la dérive à l'aide d'un modèle 3D de ville . . . 138
8.4 Estimation de la dérive à l'aide d'une carte de la route . . . 144
8.5 Intégration de la nouvelle information . . . 149
8.6 Résultats expérimentaux . . . 149
8.7 Discussion . . . 156

Conclusion **157**

Annexes **159**

A Ajustement de faisceaux par combinaison linéaire des fonctions de coût **161**
A.1 Approches existantes . . . 161
A.2 Utilisation dans notre cadre d'étude . . . 162
A.3 Résultats expérimentaux . . . 164

B Séquences vidéos utilisées **169**
B.1 Séquence Synthèse 1 . . . 170
B.2 Séquence Synthèse 2 . . . 171
B.3 Séquence Versailles 1 . . . 172

B.4 Séquence Versailles 2 173
B.5 Séquence ODIAAC 174

Bibliographie **175**

Table des figures **185**

Liste des tableaux **187**

Table des matières **194**

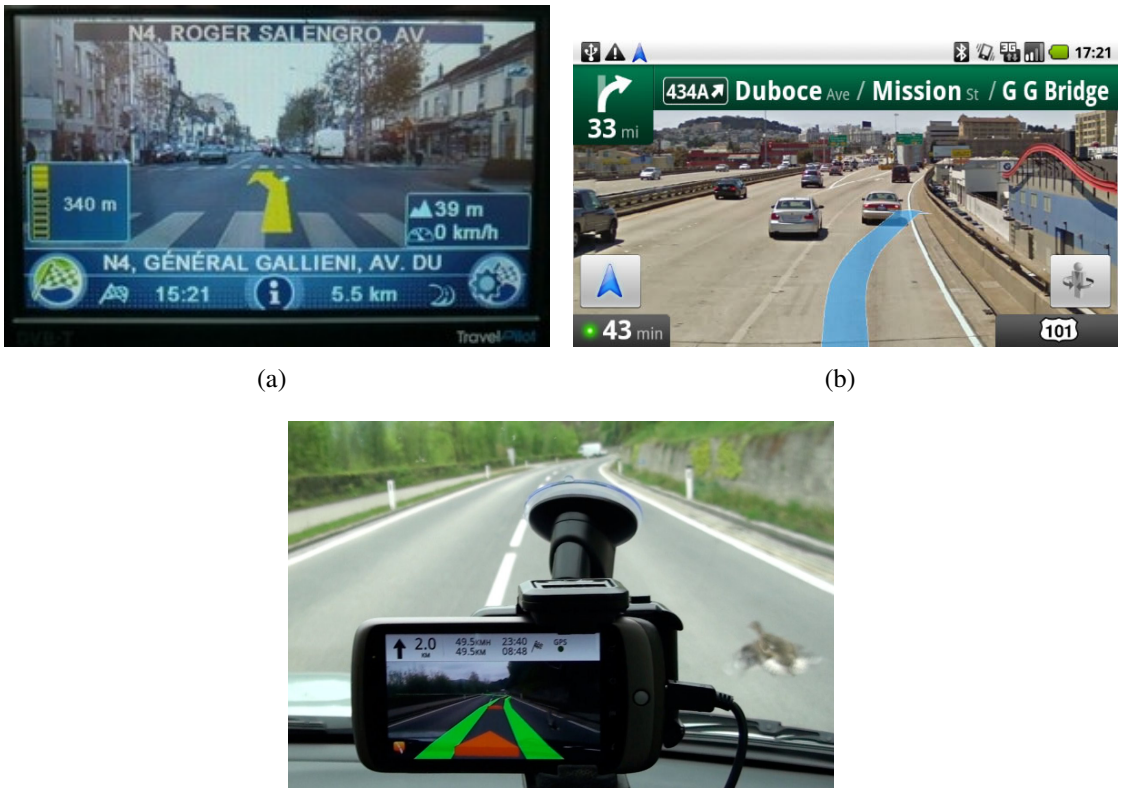
Introduction

Aujourd'hui ancré dans le quotidien d'une grande partie des populations, le GPS (pour *Global Positioning System*) est à l'origine un projet de recherche de l'armée américaine lancé dans les années 1960. En 1995, la flotte de satellites constituée est suffisante pour permettre aux militaires de localiser, avec une précision de l'ordre du décamètre, un récepteur GPS sur l'ensemble de la planète. A cette époque, certains signaux sont volontairement cryptés afin de limiter la précision exploitable par les applications civiles. Cette précision n'est alors que de l'ordre de la centaine de mètres. Le tournant du système GPS a lieu en l'an 2000, lorsque le gouvernement américain décide d'arrêter le cryptage de ces signaux. Dès lors, toutes les applications civiles peuvent bénéficier de la même précision que les militaires. C'est grâce à cette annonce que les systèmes de navigation GPS vont connaître l'explosion qu'on leur connaît aujourd'hui. En particulier, cette évolution rapide et l'importance stratégique de posséder un tel système de positionnement vont amener différents programmes visant à proposer des constellations de satellites alternatives au système américain. On peut penser en particulier au programme russe GLONASS ainsi qu'au programme européen GALILEO.

A l'heure actuelle, les systèmes de navigation sont principalement utilisés dans les domaines de la randonnée, de la navigation aérienne, maritime et terrestre. C'est ce dernier domaine qui sera l'objet de l'étude de ce mémoire. Les produits disponibles aujourd'hui sont suffisamment aboutis pour permettre de localiser un véhicule en temps-réel sur l'ensemble du territoire. Basés principalement sur le signal GPS, ces dispositifs exploitent une carte de l'environnement afin d'améliorer la précision de la localisation fournie. Néanmoins, quelques limites critiques existent encore aujourd'hui sur ces systèmes. En particulier, un des inconvénients principaux est la faible précision obtenue dans les milieux urbains, en particulier dans les centres-villes denses. On y parle souvent de la configuration de *canyon urbain*. Plusieurs phénomènes expliquent cette mauvaise précision. Tout d'abord, le nombre de satellites observables est souvent limité du fait de l'occultation des signaux par les bâtiments. Dès lors, il est possible que trop peu de satellites soient observables pour permettre la localisation. Même s'ils sont suffisamment nombreux, les satellites visibles peuvent alors présenter des conditions dégénérées (*e.g.* alignés le long de l'axe de la rue). A cela s'ajoute le problème du *multi-trajet* : avant de parvenir au récepteur, les signaux sont réfléchis par les différentes façades des bâtiments. La mesure du temps de parcours des signaux du satellite au récepteur est alors faussée. Tous ces différents phénomènes entraînent une dégradation notable de la précision de la localisation.

Localisation et vision par ordinateur

Dans le but de pallier ces difficultés, différentes communautés scientifiques (en particulier dans les domaines de la robotique et de la vision par ordinateur) ont travaillé de sorte à proposer un système de géolocalisation reposant sur l'utilisation d'une ou plusieurs caméras. L'intérêt de l'utilisation de la vision est multiple. Tout d'abord, la précision qu'il est envisageable d'obtenir à terme est nettement meilleure que celle du système GPS classique, en particulier dans les centres-villes. De plus, à l'estimation de la position courante du capteur s'ajoute son orientation, information qui n'est pas disponible lors de l'utilisation exclusive des signaux GPS. Enfin, et c'est un point important, l'utilisation d'un tel capteur permet une meilleure restitution des informations à l'utilisateur. En effet, les indications de navigation peuvent ainsi être directement ajoutées sur l'image courante de la route : on parle alors de *réalité augmentée*. La précision des indications est ainsi accrue et l'utilisateur ne perd plus de vue ce qu'il se passe sur la route.



(a) (b)

(c)

FIGURE 1 – **Exemples de systèmes d'aide à la navigation exploitant la vision.** La vidéo et les images sont récemment apparues dans les système d'aide à la navigation, par exemple (a) le Travel Pilot 700, (b) Google Navigation et (c) Wikitude Drive.

Preuve de l'intérêt de ces études, le monde industriel a récemment commercialisé les premiers produits tirant partie de cette nouvelle information apportée par les images. En premier lieu, la société Blaupunkt (figure 1(a)) a intégré une caméra à son boîtier *Travel Pilot* [1]. Celle-ci est utilisée en particulier afin de superposer les indications de changements de direction sur le flux vidéo. Néanmoins, cette information n'est pas recalée précisément sur les images. Le but ici est uniquement de permettre à l'utilisateur de voir la route même lorsqu'il consulte son système

1. www.blaupunkt.de/produkte/navigation/mobile-navigation/produkt/

de navigation. Plus récemment, les équipes Google ont proposé le logiciel *Google Navigation* [2] (figure 1(b)). L'idée de ce produit est de proposer un système où les informations de navigation sont superposées sur les images issues de *Google Street View*. Cela permet de rendre les directives fournies par le système de navigation plus facile à interpréter par l'utilisateur, en particulier dans les carrefours complexes. Cependant, les images utilisées sont ici des images statiques. On perd donc l'intérêt de l'utilisation d'un flux vidéo en temps-réel. En février 2010, la société Mobilizy a été primée au concours annuel de Navteq pour son logiciel pour smartphone intitulé *Wikitude Drive* [3]. Pour la première fois, un produit commercial propose un système d'aide à la navigation automobile utilisant de la réalité augmentée en temps-réel. Après un test grandeur nature sur 2000 utilisateurs, le produit devrait être commercialisé aux environs du mois de juin 2010. Les informations sur le fonctionnement technique ainsi que sur les possibilités de ce produit ne sont pas encore disponibles. Il est donc aujourd'hui difficile de juger l'efficacité (en terme de robustesse et de précision) de cette solution.

Les travaux présentés dans ce mémoire ont pour vocation de contribuer à la problématique liée aux systèmes de navigation par vision. En particulier, nos travaux visent à étudier la façon dont peut être exploité un Système d'Information Géographique dans ce contexte (en particulier un modèle 3D simple des bâtiments de la zone parcourue).

Contexte de la thèse

Cette thèse a été effectuée entre octobre 2007 et septembre 2010 au Laboratoire Vision et Ingénierie des Contenus (LVIC) du CEA LIST, à Saclay. L'ensemble des travaux ont été réalisés en cotutelle avec l'équipe Comsee du LASMEA, à Clermont-Ferrand.

Contributions

Nos contributions s'articulent autour de deux parties distinctes qui correspondent à deux approches différentes proposées pour résoudre le problème de la localisation par vision. La première approche proposée consiste à construire hors ligne une base d'amers visuels géoréférencée à partir de laquelle il sera possible de géolocaliser une caméra mobile en temps réel. Pour cela, nous proposons un processus permettant la correction hors ligne d'une reconstruction SLAM en exploitant un modèle 3D de l'environnement parcouru. Les contributions majeures de cette partie sont :

- ▷ **Correction grossière par recalage** (chapitre 3). Un modèle de déformation par morceaux approximant la dérive du processus SLAM est tout d'abord proposé. Ce modèle est alors exploité pour retrouver la cohérence globale de la reconstruction SLAM. Pour cela, un recalage non-rigide est effectué entre le nuage de points reconstruit et le modèle 3D de la ville.

- ▷ **Raffinement de la reconstruction** (chapitre 4). Dans le but de raffiner la reconstruction SLAM obtenue après l'étape de recalage, un ajustement de faisceaux spécifique est effectué sur l'ensemble de la reconstruction. La particularité de ce nouvel ajustement de

2. www.google.com/mobile/navigation/
3. www.wikitude.org/drive-2

faisceaux est de prendre en compte implicitement les contraintes géométriques apportées par le modèle 3D de la ville.

▷ **Application à la réalité augmentée** (chapitre 6). Dans le cadre des activités du laboratoire, une méthode de relocalisation exploitant la base d'amers créée a été proposée. Celle-ci permet de corriger ponctuellement la méthode SLAM grâce à l'information apportée par les points 3D géoréférencés de la base de données. Un concept d'application d'aide à la navigation par réalité augmentée est également présenté.

La deuxième méthode proposée explore la possibilité d'effectuer la correction du processus SLAM en ligne (*i.e.* au fur et à mesure de la reconstruction) sans nécessiter de construire au préalable une base de données de l'environnement parcouru. Pour cela, deux contributions principales sont présentées :

▷ **Correction du facteur d'échelle** (chapitre 7). La connaissance de la position relative entre la caméra et le sol permet de corriger la dérive en facteur d'échelle du processus SLAM. En particulier, nous proposons une approche qui exploite le mouvement estimé par une méthode SLAM de façon à simplifier et à rendre plus robuste l'estimation du facteur d'échelle.

▷ **Correction de l'erreur accumulée** (chapitre 8). Une méthode permettant de corriger ponctuellement l'erreur de positionnement courante de la caméra est détaillée. Pour cela, nous proposons de recaler, lorsque les contraintes géométriques sont suffisantes, la reconstruction SLAM réalisée jusqu'à l'instant courant avec les informations extraites du Système d'Information Géographique (modèle 3D de la ville ou carte de la route).

En plus du domaine de la localisation d'un véhicule visé dans ces travaux, certaines contributions peuvent être exploitées dans des contextes différents. Ainsi, le modèle de transformation proposé au chapitre 3 et la méthode de localisation du chapitre 6 ont été utilisés pour localiser un piéton au sein d'un bâtiment dans le cadre du projet EasyInteraction. De plus, l'ensemble de la méthode de correction hors ligne d'une reconstruction SLAM est actuellement à l'étude dans le projet CLIMB dans le but de géolocaliser un ensemble de caméras autour d'un bâtiment dont un modèle 3D grossier est connu. Ces caméras pourront être alors par exemple utilisées pour texturer ce modèle 3D. Enfin, le nouvel ajustement de faisceaux proposé pourra être utilisé sur des modèles CAO autres que ceux issus d'un SIG, en particulier dans le cadre de localisation et de suivi d'objets.

Les travaux réalisés au cours de cette thèse ont donné lieu à plusieurs publications (Lothe et al. (2009a,b,c,d, 2010a,c,b); Gay-Bellile et al. (2010)).

Organisation du mémoire

En premier lieu, le chapitre 1 fait un tour d'horizon des méthodes de SLAM classiques puis présente les travaux qui existent aujourd'hui sur l'exploitation d'un SIG pour la localisation d'une caméra. Le chapitre 2 présente quant à lui l'ensemble des notations et des outils de bases nécessaires à la bonne compréhension du mémoire.

La structure de la suite du mémoire est guidée par les deux solutions proposées pour la localisation d'un véhicule en temps-réel. Dans la partie I sont présentées l'étape de correction

grossière (chapitre 3) et l'étape de raffinement (chapitre 4) permettant de corriger hors ligne une reconstruction SLAM à l'aide d'un modèle 3D de ville. Les résultats obtenus sur cette méthode sont illustrés dans le chapitre 5 puis exploités dans une application de réalité augmentée (chapitre 6). La deuxième partie détaille les deux processus permettant la correction en ligne de la dérive du SLAM. Le chapitre 7 contient les informations relatives à la correction du facteur d'échelle. Le chapitre 8 décrit alors de quelle façon le SIG peut être exploité pour corriger ponctuellement la position courante de la caméra.

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

CHAPITRE 1

Etat de l'art

Ce premier chapitre a pour but, dans un premier temps, de faire le tour d'horizon des approches existantes pour la localisation d'une unique caméra sans connaissance a priori sur l'environnement parcouru. Dans un second temps, des méthodes exploitant une connaissance partielle sur cet environnement (issue d'un Système d'Information Géographique) seront présentées.

1.1 Localisation monoculaire en environnement inconnu

Dans cette section, nous présenterons tout d'abord l'approche générale classique permettant de résoudre le problème de la localisation d'une caméra mobile (embarquée sur un véhicule dans le cas qui nous intéresse). Ensuite, nous détaillerons les principales méthodes qui existent aujourd'hui. Enfin, les limites liées à ces méthodes seront mises en avant.

1.1.1 Idée générale

Le principe de la localisation par vision monoculaire d'un véhicule est identique pour la grande majorité des méthodes. La première position de la caméra est inconnue et est donc fixée arbitrairement. Par la suite, le déplacement en 3 dimensions de la caméra dans l'environnement qui l'entoure va se traduire visuellement dans l'image par un déplacement en 2 dimensions des éléments d'intérêt de l'image. Ces éléments d'intérêt sont généralement des zones ou plus couramment des points de l'image qu'il est possible de suivre au fil du flux vidéo, c'est à dire entre les différentes images.

A partir de l'observation du déplacement de ces éléments d'intérêt, il est possible d'inférer le mouvement de la caméra. Ceci peut être réalisé directement à partir de l'information 2D : on parlera alors généralement 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 dans la suite de SLAM (Simultaneous Localization and Mapping) ou de SfM (Structure from Motion) incrémental.

1.1.2 Familles de méthodes existantes

Les premiers travaux sur la localisation en environnement inconnu sont apparus dans le courant des années 1980 et sont souvent associés aux publications de Smith and Cheesman (1987) et de Moutarlier and Chatila (1989). Depuis, trois familles principales coexistent : l'odométrie visuelle, la localisation et cartographie simultanées (SLAM) et les méthodes de reconstructions 3D (Structure from Motion). Notons qu'il est souvent difficile de classer les méthodes de l'état de l'art dans ces différentes catégories. En effet, initialement, ces familles se distinguaient principalement par le but qu'elles visaient. L'odométrie visuelle était principalement axée sur la reconstruction de la trajectoire de la caméra, le Structure from Motion sur la géométrie 3D de l'environnement et le SLAM sur ces deux informations à la fois. Néanmoins, nous allons voir que ces différentes méthodes tendent aujourd'hui à se rapprocher.

1.1.2.1 Odométrie visuelle

Comme nous l'avons précisé précédemment, l'idée de l'odométrie visuelle est de relier directement le déplacement 2D des éléments d'intérêt dans les images au déplacement en 3D de la caméra. En particulier, ceci implique qu'aucune reconstruction 3D de l'environnement n'est nécessaire à la localisation de la caméra au fil du temps. Pour résoudre ce problème, il est par exemple possible de s'appuyer sur la géométrie épipolaire liant les images successives (comme le font Tardif et al. (2008) pour obtenir la rotation du mouvement) ou d'aligner au mieux ces images (Comport et al. (2007)). Néanmoins, l'estimation du déplacement à partir de la transformation 2D observée dans les images est un processus généralement coûteux et peu robuste. De ce fait, il est souvent nécessaire de simplifier la transformation 2D recherchée entre les images. Pour cela, une idée classique utilisée en odométrie visuelle est d'exploiter une hypothèse simple sur l'environnement. En particulier, dans le cadre du déplacement d'un véhicule terrestre, il est courant d'utiliser 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)) ou de plusieurs plans de la scène (Simond and Rives (2004); Silveira et al. (2008)). Cette idée utilise alors le fait que les observations d'un même plan entre plusieurs images décrivent une homographie. Une fois cette homographie calculée, il est alors possible d'en extraire en particulier le déplacement de la caméra (Triggs (1998)). Ainsi, la trajectoire du véhicule peut être reconstruite sur l'ensemble de la séquence vidéo.

1.1.2.2 Localisation et cartographie simultanées par vision

Le processus de localisation et cartographie simultanées (SLAM) est un problème relativement ancien dans le domaine de la robotique. L'idée sous-jacente à cette approche est que l'observation d'un même point d'intérêt à des instants différents permet de connaître 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 réalisées à l'instant $t+1$: de nouveaux amers sont créés et la position des amers déjà existants est raffinée. Pour répondre à ce problème, la communauté 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 de SLAM par vision basés sur le filtre de Kalman, le vecteur d'état

est composé à la fois de la pose courante de la caméra et de la position de l'ensemble des amers qui constituent la carte. Lorsqu'une nouvelle image est disponible, elle contient des amers qui sont déjà dans la carte de l'environnement. Ce sont ces observations qui vont permettre de localiser la caméra. La carte est alors enrichie. La position des amers existants est raffinée et de nouveaux amers sont ajoutés, ce qui permet de cartographier des zones de l'environnement qui n'ont pas encore été explorées. De plus, à tout instant, une matrice de covariance est associée au vecteur d'état. Celle-ci permet de quantifier l'incertitude sur chacune des données de l'état courant. Notons de plus que, du fait qu'elle exploite constamment l'ensemble de la carte, l'approche SLAM permet de prendre en compte intrinsèquement la fermeture de boucle. Celle-ci consiste à corriger la carte de l'environnement (et donc de la position courante de la caméra) à partir d'amers rencontrées précédemment puis observés à nouveau.

Néanmoins, la limite la plus importante de l'approche SLAM basée sur le filtre de Kalman 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 N^2, où N 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. Notons cependant que des travaux récents cherchent à pallier ce problème. Par exemple, Civera et al. (2009) suppriment de la carte les points qui ne sont plus observés au fur et à mesure de la trajectoire. De leur côté, Leonard and Newman (2003) proposent de gérer plusieurs sous-cartes. Notons néanmoins que dans ces différents cas, la gestion de la fermeture de boucles n'est alors plus automatique. Des solutions spécifiques doivent alors être proposées (Clemente et al. (2007)).

1.1.2.3 Structure from Motion

En parallèle, la communauté de vision par ordinateur s'est également intéressée au problème de la localisation d'une caméra mobile (et donc par extension d'un robot). Au contraire de la communauté robotique, les outils utilisés sont des outils d'optimisation. En particulier, le filtre de Kalman est remplacé par l'ajustement de faisceaux (Triggs et al. (2000)). Dans un premier temps, le défi de cette communauté a été de réaliser une reconstruction 3D (généralement sous la forme d'un nuage de points 3D et d'un ensemble de caméras localisées par rapport à ce nuage) à partir d'une collection d'images. On parle alors de *Structure from Motion* (parfois traduit *Stéréo-mouvement* en français). Des travaux récents montrent qu'il est désormais possible d'obtenir des reconstructions de qualité pour des environnements de très grande ampleur (à l'échelle d'une ville) à partir de photos librement disponibles sur internet (Agarwal et al. (2009)). La qualité des reconstructions obtenues est en particulier liée au fait que, contrairement à l'approche SLAM, toutes les images sont connues dès le début du processus. Néanmoins, l'inconvénient de l'approche SfM est que la localisation des caméras est réalisée hors ligne.

De façon à pallier ce problème, une nouvelle approche de Structure from Motion incrémental est apparue (Nister et al. (2004); Mouragnon et al. (2006); Engels et al. (2006)). Celle-ci consiste à utiliser en ligne et de façon incrémentale les outils classiques de vision par ordinateur (triangulation, calcul de pose, ajustement de faisceaux, *etc.*). La reconstruction 3D ainsi que la localisation du capteur se fait dans ce cas au fur et à mesure de l'arrivée de nouvelles images. A l'heure actuelle, ces méthodes permettent d'obtenir en temps-réel une localisation de caméra sur de longues distances (plusieurs kilomètres). En particulier, certaines études (Civera et al. (2009); Klein and Murray (2007)) ont mis en avant que la précision obtenue grâce à l'ap-

proche de Structure from Motion incrémental avec ajustement de faisceaux est meilleure que pour les méthodes de SLAM. Notons de plus que des travaux récents permettent de combler certaines différences qui existaient avec le SLAM, à savoir la propagation des covariances à travers l'ajustement de faisceaux (Eudes and Lhuillier (2009)), l'exploitation de points reconstruits déjà rencontrés auparavant (Klein and Murray (2007)) en particulier pour la fermeture de boucles (Strasdat et al. (2010)).

Comme nous venons de le constater, sous le nom de SLAM et de Structure from Motion incrémental se retrouvent des méthodes ayant le même but : 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 plus, même si les méthodes d'odométrie visuelle n'exploitent pas de carte de l'environnement pour se localiser, il est tout à fait possible d'en construire une au fur et à mesure que la trajectoire est reconstruite (*e.g.* Scaramuzza et al. (2009b)). En ce sens, les trois méthodes présentées 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 (ou localisation et cartographie simultanées) puisque selon nous, c'est ce terme qui traduit le mieux l'idée générale sur laquelle s'appuient ces méthodes.

1.1.3 Limites des méthodes existantes

Si les méthodes de SLAM monoculaire actuelles permettent de reconstruire la trajectoire d'une caméra sur des distances importantes, elles présentent néanmoins encore aujourd'hui des limites qui empêchent leur utilisation dans un grand nombre d'applications.

1.1.3.1 Localisation non géoréférencée

Dans l'approche classique du SLAM, aucune information n'est disponible sur l'environnement dans lequel la caméra évolue. L'endroit précis dans le monde (*i.e.* la géolocalisation) de cette caméra est par conséquent inconnu. En effet, la seule information fournie par le SLAM est le déplacement relatif de la caméra au fil du temps. L'absence de géolocalisation 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.

1.1.3.2 Dérives sur les longues trajectoires

La cause principale de la dérive observée dans les processus de SLAM est l'erreur qui existe sur les 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 des points d'intérêt n'est pas parfaite. Les coordonnées des points d'intérêt présentent en effet un bruit, généralement supposé gaussien. Naturellement, les erreurs effectuées sur les mesures 2D ont un impact direct sur la reconstruction 3D des points observés et sur la localisation de la caméra. En particulier, nos expériences menées sur la méthode proposée par Mouragnon et al. (2006) mettent en avant deux types de dérives principales : l'accumulation d'erreur et la dérive du facteur d'échelle.

Accumulation d'erreur. Le processus de SLAM est un processus incrémental. En effet, la position reconstruite de la caméra à l'instant t dépend des données collectées jusqu'à l'instant $t-1$. Or, comme nous l'avons souligné ci-avant, l'estimation de l'état courant (*i.e.* la position de la caméra et des amers dans la carte) à partir des données de l'instant précédent est entachée d'erreurs. En effet, des erreurs directement liées à la méthode (erreurs dans les données observées) et à son exécution (précision de calcul limitée sur les machines) impliquent une erreur sur le calcul du déplacement relatif entre les instants $t-1$ et t. De plus, à l'erreur réalisée sur l'estimation du déplacement entre les instants $t-1$ et t s'ajoute l'erreur initialement réalisée sur l'état à l'instant $t-1$. Au cours du processus de SLAM, les erreurs s'ajoutent donc à chaque nouvelle estimation : on parle alors du phénomène d'*accumulation d'erreur*. Ainsi, l'erreur réalisée sur la position de la caméra et sur la structure de la carte augmente au fil du temps. Sur des séquences vidéos de taille importante, l'état courant de la scène fourni par le SLAM peut donc être très éloigné de la réalité de la scène. Dès lors, les méthodes de SLAM peuvent difficilement être utilisées en l'état sur plusieurs kilomètres. Dans la suite de ce mémoire, nous désignerons ce problème comme étant la *dérive liée à l'accumulation d'erreur*.

Dérive du facteur d'échelle. Dans certaines configurations, l'estimation de la pose à partir d'observations 3D peut être fortement erronée. On peut par exemple penser aux cas où les points observés sont mal répartis dans l'image, le cas où la scène n'est pas rigide (*e.g.* nombreux véhicules se déplaçant), *etc.* Si l'accumulation d'erreur est une dérive relativement lente et lisse, l'erreur associée à ce genre de mauvaises configurations est généralement beaucoup plus importante et localisée dans le temps. En théorie, dans ces configurations, c'est l'ensemble de la pose de la caméra (*i.e.* les 3 composantes en translation et les 3 composantes en rotation) qui peut être erronée. Néanmoins, nous observons expérimentalement que l'erreur se situe quasiment uniquement sur la norme du déplacement entre la caméra courante et la caméra précédente. Cette norme étant incorrecte, l'ensemble des points 3D observés sont reconstruits à une mauvaise échelle. Par la nature incrémentale du processus SLAM, ce mauvais facteur d'échelle est alors transmis de proche en proche à l'ensemble de la suite de la reconstruction SLAM. Dans ce mémoire, nous parlerons de ce phénomène sous le nom de *dérive en facteur d'échelle*.

1.2 Environnements partiellement connus : exploitation des Systèmes d'Information Géographique

Afin de pallier les limites des méthodes SLAM citées dans la section précédente, une idée récente consiste à exploiter une information additionnelle sur l'environnement parcouru issue d'un Système d'Information Géographique (SIG). En pratique, la donnée utilisée est très souvent une image satellite ou un modèle 2D (empreinte au sol) ou 3D des bâtiments de la zone explorée. L'augmentation récente du nombre de travaux basés sur cette approche peut s'expliquer par différentes raisons. Tout d'abord, les données des SIG (détaillées à la section 2.6.2) sont exemptes de dérive. De plus, ces données sont de plus en plus largement utilisées, 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 (ceci sera détaillé à la section 2.6.2). Pour exploiter ce nouveau type d'information, différentes approches sont proposées.

1.2.1 Exploitation de l'information photo-géométrique

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 et qui est appliquée avec précision sur ce modèle. Dès lors, il est envisageable d'exploiter l'information photométrique associée à ce modèle 3D. L'image courante de la caméra peut en effet être recalée à chaque instant à la texture de ce modèle. La caméra est alors localisée en temps-réel par rapport au modèle 3D. Si celui est géolocalisé, la caméra l'est donc également.

Par exemple, Reitmayr and Drummond (2006) utilisent la texture du modèle dans le but d'améliorer leur algorithme de suivi de contours. En effet, aux contours correspondant aux arêtes des bâtiments s'ajoutent les contours liés aux motifs de la texture (changement de couleur, de matière, texture avec motifs apparents, *etc.*). L'information disponible est par conséquent plus importante, ce qui permet d'améliorer la robustesse et la précision du recalage. Cappelle et al. (2010) proposent une approche différente dans un contexte véhicule. Un odomètre couplé à un gyroscope fournit une première estimation de la pose courante de la caméra. Cette position est alors raffinée 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 1.1). La pose correspondant à l'image virtuelle visuellement la plus proche de l'image réelle courante est alors retenue comme une nouvelle observation de la pose courante.



FIGURE 1.1 – **Utilisation de 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)).*

Les approches présentées ci-avant permettent d'obtenir en temps-réel 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. Dès lors, la communauté de vision par ordinateur a proposé d'utiliser comme modèles 3D texturés les reconstructions obtenues à partir des méthodes de Structure from Motion. En effet, une fois un tel nuage de points reconstruits, il est possible de relocaliser en temps-réel une caméra en mettant en relation ses observations courantes avec les points 3D de la reconstruction (Royer et al. (2007); Irschara et al. (2009)). Un des problèmes de cette approche est que la caméra est alors uniquement localisée par rapport à ce nuage de points. En particulier, pour que la caméra soit géolocalisée, il est nécessaire que le nuage de points le soit au préalable. De plus, tout comme pour la méthode proposée par Cappelle et al. (2010), 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 changements de point de vue et aux conditions d'illumination (position du soleil, heure de la journée, *etc.*). Notons cependant que dans l'approche retenue par Cappelle et al. (2010), le passage par la génération d'une image de synthèse permet d'éviter les problèmes de robustesse liés aux changements de point de vue.

1.2.2 Exploitation exclusive de l'information géométrique

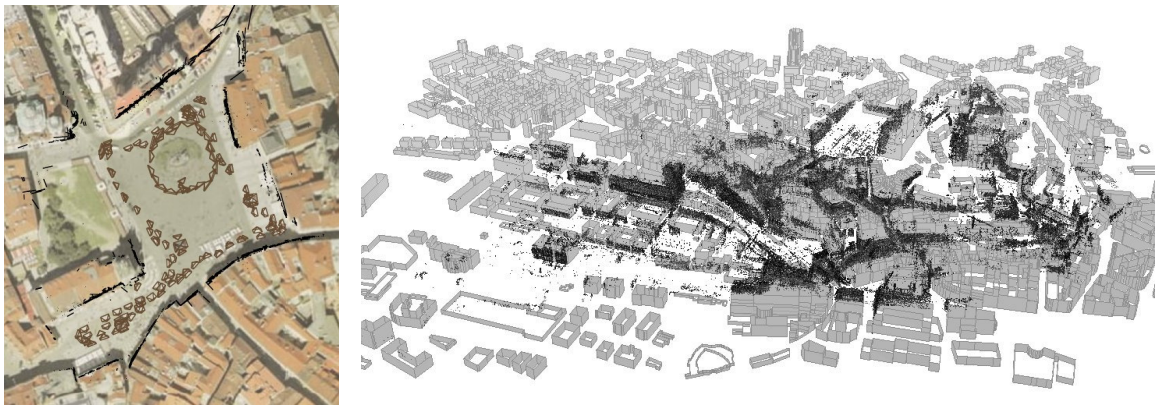
Pour pallier les limites liées à l'utilisation unique d'une information photométrique, il est intéressant d'exploiter uniquement l'information géométrique apportée par le Système d'Information Géographique. En effet, cette information est durable dans le temps et est indépendante des conditions d'observation. Peu de travaux sur cette approche existaient au début de notre étude (en 2007) et, comme nous allons le voir, l'activité dans ce domaine a fortement augmentée au cours de ces trois dernières années (*i.e.* en parallèle de nos travaux ou par la suite). Les travaux existants aujourd'hui peuvent être différenciés par le fait que la localisation de la caméra soit réalisée hors ligne ou en ligne.

1.2.2.1 Localisation hors ligne

Comme indiqué dans la section 1.1.2.3, les méthodes de Structure from Motion permettent de reconstruire à partir d'une collection d'images un nuage de points décrivant la géométrie de la scène ainsi que la position de l'ensemble de ces images. L'information issue du Système d'Information Géographique 3D peut alors être utilisée de façon à géolocaliser cette reconstruction.

Dans le cas où les prises de vue sont convergentes (*i.e.* points de vue différents du même objet), les dérives décrites dans la section 1.1.3 peuvent souvent être négligées. La géolocalisation de la reconstruction revient alors à chercher la similitude (dans l'espace ou dans le plan) qui permet d'aligner au mieux la reconstruction SfM au SIG. Par exemple, Grzeszczuk et al. (2009) cherchent à recaler le nuage de points reconstruit sur l'empreinte des bâtiments extraite d'une image satellite. Plus récemment, Strecha et al. (2010) ont montré que la même approche pouvait être employée en substituant l'empreinte des bâtiments par leur modèle 3D (figure 1.2(b)). Dans leurs travaux, en plus de la distance entre le nuage de points et les bâtiments, Kaminsky et al. (2009) proposent de prendre en compte la notion d'espace vide (figure 1.2(a)). Pour cela, ils partent du principe que si une caméra observe un point, c'est nécessairement qu'aucun obstacle ne se situe entre ces deux éléments. Le fait d'ajouter cet aspect à la fonction de coût à minimiser permet alors de désambiguïser certaines configurations et donc d'améliorer la robustesse du recalage.

Dans le cas où la dérive de la reconstruction ne peut plus être négligée (trajectoire d'un véhicule sur une distance importante par exemple), la transformation nécessaire pour aligner la reconstruction au SIG n'est plus uniquement une similitude. Un modèle de transformation plus complexe (à définir) doit être utilisé. Dans ce cas, le recalage entre la reconstruction et le SIG permet à la fois de corriger et de géolocaliser la reconstruction SfM. Plusieurs types de données ont été récemment exploitées pour réaliser cette correction. Par exemple, Levin and Szeliski (2004) utilisent une carte de la trajectoire dessinée à la main. Cette méthode ayant pour limite la faible précision de la carte dessinée, Saurer et al. (2010) ont proposé de la remplacer par un ensemble de points de contrôle, ceux-ci étant fixés par l'utilisateur sur le plan de la zone parcourue. Dans le même esprit, Pylvänäinen et al. (2010) se sont inspirés des travaux que nous avons proposés (Lothe et al. (2009d)) afin d'exploiter le modèle 3D des bâtiments d'un quartier pour corriger le nuage de points reconstruit (obtenu dans ce cas non pas par vision mais grâce à



(a) (b)

FIGURE 1.2 – **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)*).

un LIDAR).

Une fois la reconstruction SfM ainsi alignée sur le SIG, toutes les images qui composent cette reconstruction sont bien géolocalisées. Cette localisation hors ligne est par exemple utile pour des applications de visites virtuelles (Irschara et al. (2009); Saurer et al. (2010)). De plus, comme cela a été mentionné à la section 1.2.1, cette reconstruction SfM peut aussi être considérée comme étant une base de données à partir de laquelle une caméra mobile pourra alors être localisée en ligne (Royer et al. (2007)). Néanmoins, dans ce cas, la méthode retombe sur les limites liées à l'utilisation de données photométriques.

1.2.2.2 Localisation en ligne

Localisation d'une unique image. Une première famille de travaux visent à estimer la position absolue d'une unique image au sein de l'environnement. Par exemple, Pink (2008) propose de construire au préalable (*i.e.* hors ligne) une carte de la zone couverte. Cette carte rassemble l'ensemble des marquages au sol extraits des images satellites. Une fois cette carte créée, elle peut alors être exploitée pour localiser la caméra. Une première position grossière est obtenue par l'intermédiaire d'un GPS. Les marquages au sol de l'image sont alors détectés et alignés au mieux avec la carte par l'intermédiaire d'un algorithme de type ICP (Iterative Closest Point, Rusinkiewicz and Levoy (2001)). Une autre approche récemment proposée est d'utiliser l'empreinte au sol des bâtiments (Bioret et al. (2009); Cham et al. (2010)). 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 estimée est alors recherchée au sein de la carte des empreintes au sol. Le principal inconvénient de ces méthodes est de se limiter à une unique image. De ce fait, la position de la caméra ne peut pas toujours être déterminée (plusieurs solutions possibles) ou peut être parfois erronée (erreur ou manque de mise à jour dans la carte utilisée). Pour éviter ces problèmes, il est alors nécessaire d'utiliser un autre capteur (comme par exemple le fait Pink (2008)) qui permet alors de filtrer la pose estimée à partir de l'image.

Localisation d'une caméra mobile. Une autre famille de méthodes tend à localiser une caméra en mouvement en exploitant en ligne l'information issue du SIG. Contrairement aux mé-

thodes présentées dans la section 1.2.1, seule l'information géométrique du SIG est ici exploitée. Il est ainsi possible de localiser en temps-réel une caméra mobile sans reposer sur la mise en correspondance d'images temporellement éloignées. Au jour d'aujourd'hui, peu de travaux ont été réalisés dans ce domaine. Néanmoins, de premières idées apparaissent. En particulier, Sourimant et al. (2007) ont proposé, de sorte à calculer la position 3D des points observés, de remplacer la géométrie multi-vue classique par la rétroprojection de leurs observations sur le modèle 3D (figure 1.3). 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 (figure 1.3). Cependant, une fois calculée, la position des points 3D n'est jamais remise en cause. Ainsi, la précision de cette position est directement liée à la précision du modèle 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. Par ailleurs, dans leurs travaux de suivi d'objets, Vacchetti et al. (2004) ont montré que cette approche pouvait être utilisée avec succès dès lors que le modèle 3D utilisé est suffisamment précis. Notons cependant que dans ces travaux, en plus des points rétroprojetés sur le modèle 3D, des points 3D du modèle connus au préalable sont également utilisés lors du calcul de pose. Cela permet d'éviter les problèmes de dérive inhérents aux méthodes de localisation incrémentales. De tels points 3D peuvent être par exemple récoltés dans une phase d'apprentissage préalable (Platonov et al. (2006)). Cependant, la nécessité de reconnaître des points 3D connus de l'environnement nous ramènent aux problèmes liés à l'utilisation de l'information photométrique d'un modèle 3D.

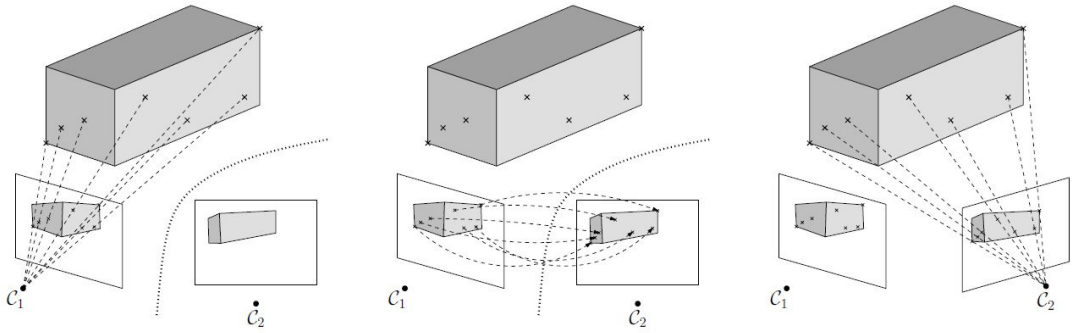


FIGURE 1.3 – **Exploitation de la géométrie du modèle 3D.** Si un modèle 3D de l'environnement est connu, la géométrie multi-vue peut être remplacée par la rétroprojection des observations sur ce modèle *(extrait de Sourimant et al. (2007))*.

CHAPITRE 2

Notions de base et données utilisées

Dans ce chapitre, nous introduisons les notions de base et notations nécessaires à la compréhension du mémoire. Après avoir introduit le concept de géométrie projective, nous présenterons les caméras perspectives et la géométrie qui leur est associée. Plus de détails sur ces notions peuvent être trouvées dans le livre de Hartley and Zisserman (2004). Nous décrirons enfin les méthodes et données d'entrée de nos travaux, à savoir un algorithme de localisation et cartographie simultanées par vision monoculaire et les modèles 3D des milieux urbains.

2.1 Géométrie projective

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

$$\mathbf{u} \sim \mathbf{v} \Longleftrightarrow \exists \lambda \in \mathbb{R}^* / \mathbf{u} = \lambda \mathbf{v} \tag{2.1}$$

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 :

$$\tilde{\mathbf{x}} = (x_1 \ \dots \ x_{n+1})^\mathsf{T} \tag{2.2}$$

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 ix. Nous appellerons π la fonction permettant de passer des coordonnées homogènes aux coordonnées euclidiennes, à savoir :

$$\begin{array}{rccc} \pi \ : & \mathbb{P}^n & \rightarrow & \mathbb{R}^n \\ & (x_1 \ \dots \ x_{n+1})^\mathsf{T} & \mapsto & (x_1/x_{n+1} \ \dots \ x_n/x_{n+1})^\mathsf{T} \end{array} \tag{2.3}$$

Dans la suite de ce chapitre, nous allons étudier le cas particulier de cette géométrie en deux puis trois dimensions.

2.1.1 Le plan projectif

L'espace projectif de dimension 2 est appelé *plan projectif*. Un point de $\mathbb{P}^2$ est représenté par un vecteur de dimension 3 : $\tilde{\mathbf{q}} = (x\ y\ w)^\mathsf{T}$. De même, une droite d'équation $ax + by + c = 0$ peut être représentée par le vecteur $\mathbf{l} = (a\ b\ c)^\mathsf{T}$. Cette notation homogène permet de définir simplement la notion d'appartenance du point $\tilde{\mathbf{q}}$ à la droite $\mathbf{l}$, à savoir :

$$\mathbf{l}^\mathsf{T}\tilde{\mathbf{q}} = 0 \tag{2.4}$$

L'équation de la droite $\mathbf{l}$ passant par les points $\mathbf{q}_1$ et $\mathbf{q}_2$ est obtenue en calculant leur produit vectoriel :

$$\mathbf{l} = \tilde{\mathbf{q}}_1 \wedge \tilde{\mathbf{q}}_2 \tag{2.5}$$

Dans l'espace $\mathbb{P}^2$, droites et points jouent un rôle équivalent : c'est ce qu'on appelle le *principe de dualité*. En particulier, à partir de l'équation duale de l'équation précédente, il est possible de calculer le point d'intersection de deux droites $\mathbf{l}_1$ et $\mathbf{l}_2$:

$$\tilde{\mathbf{q}} = \mathbf{l}_1 \wedge \mathbf{l}_2 \tag{2.6}$$

2.1.2 L'espace projectif 3D

Un point $\mathcal{Q}$ de $\mathbb{R}^3$ aura pour coordonnées homogènes dans $\mathbb{P}^3$ le vecteur $\tilde{\mathcal{Q}} = (X\ Y\ Z\ W)^\mathsf{T}$. Dans cet espace de dimension 3, le dual du point $\tilde{\mathcal{Q}}$ est le plan $\mathbf{\Pi}$ d'équation $aX + bY + cZ + d = 0$ qui est représenté par le vecteur $\mathbf{\Pi} = (a\ b\ c\ d)^\mathsf{T}$. L'appartenance du point $\tilde{\mathcal{Q}}$ au plan $\mathbf{\Pi}$ est alors donné par la relation :

$$\mathbf{\Pi}^\mathsf{T}\tilde{\mathcal{Q}} = 0 \tag{2.7}$$

2.2 Caméras perspectives et géométrie associée

Dans le cadre de nos travaux, nous travaillerons sur les caméras perspectives. Ces caméras sont des caméras centrales, c'est à dire qu'il est considéré que l'ensemble des rayons lumineux passent par un seul et unique point avant d'atteindre le capteur (voir figure 2.1). De plus, les caméras utilisées respectent le modèle des *caméras sténopé* idéales, ce qui permet en particulier de préserver la colinéarité.

Dans la suite, nous présenterons tout d'abord les différents paramètres qui caractérisent 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.

2.2.1 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 2.1).

Cette projection peut être vue comme un enchaînement de trois transformations géométriques (figure 2.1) :

- ▷ 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 points 3D seront indicés $\mathcal{W}$ ou $\mathcal{C}$ en fonction du repère dans lequel sont exprimées leurs coordonnées (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.

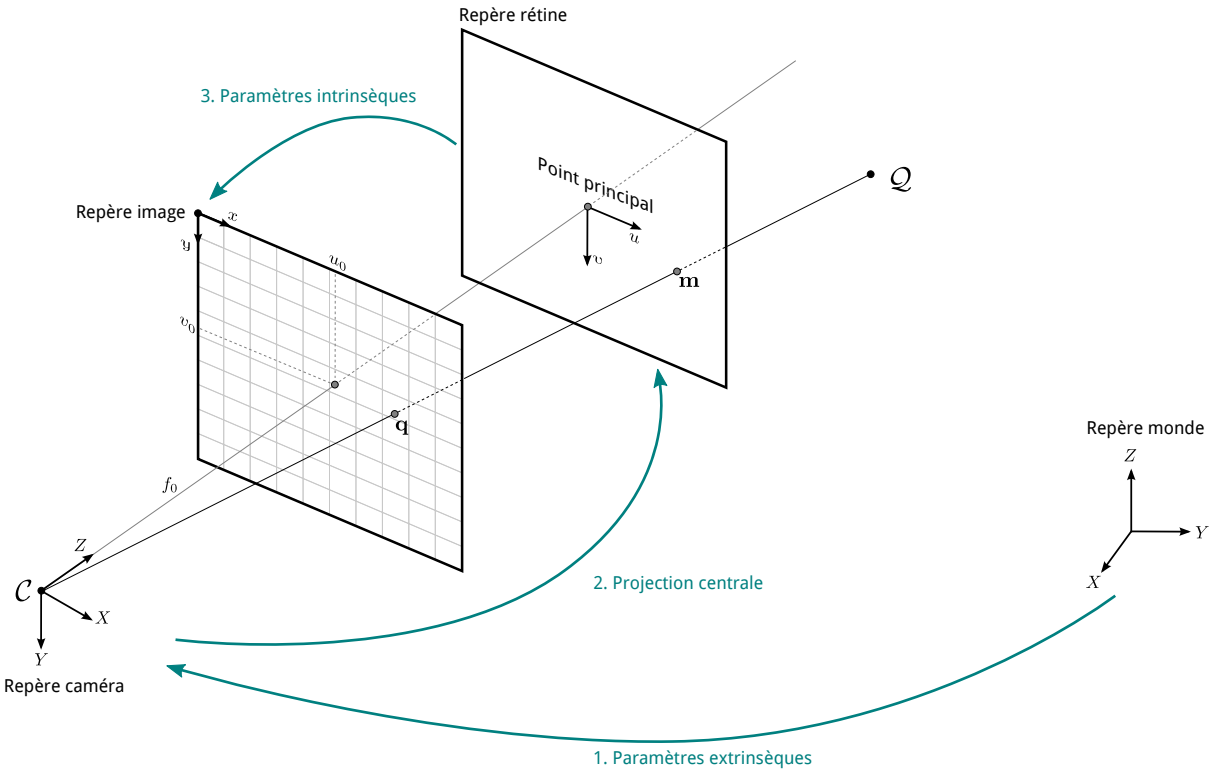


FIGURE 2.1 – **Projection perspective.** La projection perspective peut être vue comme trois transformations géométriques consécutives pour les points 3D.

Dans la suite du mémoire, afin d'éviter les confusions, nous différencierons les notations utilisées pour les points 2D en fonction du repère dans lequel ils sont exprimés. Ainsi, les points

2D du repère rétinien seront notés $\mathbf{m}$ et ceux du repère image seront notés $\mathbf{q}$.

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

$$\tilde{\mathbf{q}} \sim \tilde{\mathsf{P}} \tilde{\mathcal{Q}}_{\mathcal{W}} \tag{2.8}$$

La matrice $\tilde{\mathsf{P}}$ se décompose selon les trois transformations citées précédemment :

$$\tilde{\mathsf{P}} = \mathsf{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathsf{R}^\mathsf{T} & -\mathsf{R}^\mathsf{T}\mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix} \tag{2.9}$$

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 $\begin{pmatrix} \mathsf{R}^\mathsf{T} & -\mathsf{R}^\mathsf{T}\mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix}$ la *matrice de pose*.

2.2.1.1 Paramètres extrinsèques

Les paramètres extrinsèques d'une caméra caractérisent la pose de celle-ci dans le repère monde. La pose d'une caméra possède 6 degrés de liberté :

- ▷ La position 3D du centre optique, décrit par le vecteur $\mathbf{t} = (t_x\ t_y\ t_z)^\mathsf{T}$
- ▷ 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 d'Euler par exemple.

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

$$\tilde{\mathcal{Q}}_{\mathcal{C}} \sim \begin{pmatrix} \mathsf{R}^\mathsf{T} & -\mathsf{R}^\mathsf{T}\mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{W}} \tag{2.10}$$

$$\tilde{\mathcal{Q}}_{\mathcal{W}} \sim \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{C}} \tag{2.11}$$

2.2.1.2 Projection centrale

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

$$\tilde{\mathbf{q}} \sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{C}} \tag{2.12}$$

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 :

$$\begin{aligned}\tilde{\mathbf{m}} &\sim \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}\mathsf{R}^\mathsf{T} & -\mathsf{R}^\mathsf{T}\mathbf{t}\\ 0_{1\times 3} & 1\end{pmatrix}\tilde{\mathcal{Q}}_{\mathcal{W}}\\ &\sim \begin{pmatrix}\mathsf{R}^\mathsf{T} & -\mathsf{R}^\mathsf{T}\mathbf{t}\end{pmatrix}\tilde{\mathcal{Q}}_{\mathcal{W}}\end{aligned} \tag{2.13}$$

La projection du monde dans l'image (équation 2.8) s'écrit donc généralement sous cette forme :

$$\tilde{\mathbf{q}} \sim \mathsf{K}\begin{pmatrix}\mathsf{R}^\mathsf{T} & -\mathsf{R}^\mathsf{T}\mathbf{t}\end{pmatrix}\tilde{\mathcal{Q}}_{\mathcal{W}} \tag{2.14}$$

2.2.1.3 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 :

$$\mathsf{K} = \begin{pmatrix}f_0 & 0 & u_0\\ 0 & f_0 & v_0\\ 0 & 0 & 1\end{pmatrix} \tag{2.15}$$

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 2.1).

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).

2.2.2 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 :

$$\tilde{\mathcal{Q}}(\lambda) \sim \tilde{\mathsf{P}}^{+}\tilde{\mathbf{q}} + \lambda\tilde{\mathcal{C}} \tag{2.16}$$

où $\tilde{\mathsf{P}}^{+}$ désigne la pseudo-inverse de la matrice $\tilde{\mathsf{P}}$:

$$\tilde{\mathsf{P}}^{+} \;=\; \tilde{\mathsf{P}}^{\mathsf{T}}(\tilde{\mathsf{P}}\tilde{\mathsf{P}}^{\mathsf{T}})^{-1} \tag{2.17}$$

De la même façon, on peut définir la rétroprojection d'une droite $\mathbf{l}$ dans l'image qui décrit un plan de l'espace 3D $\mathbf{\Pi}$:

$$\mathbf{\Pi} \sim \tilde{\mathsf{P}}^{\mathsf{T}}\mathbf{l} \tag{2.18}$$

2.3 Géométrie multi-vue

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 pencherons sur 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).

2.3.1 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 2.2). 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 est 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.

2.3.1.1 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 $\mathbf{l}_2$ sur laquelle se situe l'observation correspondante dans la deuxième caméra (figure 2.2) :

$$\mathbf{l}_2 \sim \mathsf{F}\tilde{\mathbf{q}}_1 \tag{2.19}$$

La droite $\mathbf{l}_2$ 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 :

$$\tilde{\mathbf{q}}_2{}^{\mathsf{T}}\mathsf{F}\tilde{\mathbf{q}}_1 = 0 \tag{2.20}$$

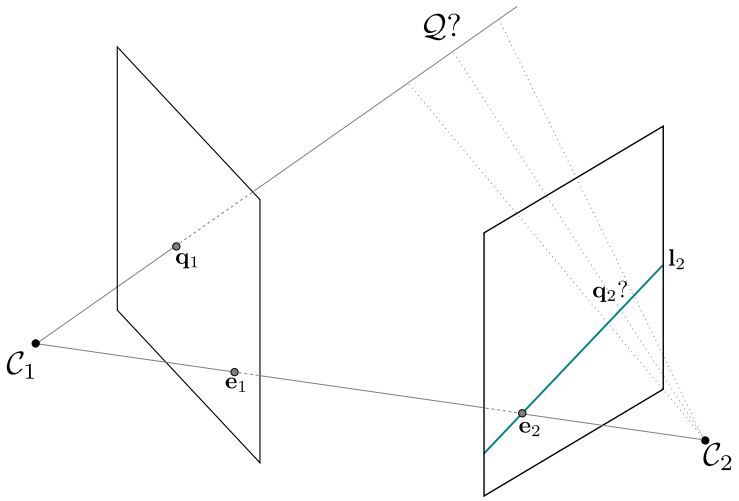


FIGURE 2.2 – **Géométrie épipolaire.** La géométrie épipolaire définit des contraintes géométriques entre les différentes observations d'un même point de l'espace.

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, ils 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.

2.3.1.2 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 :

$$\mathsf{E} \sim {\mathsf{K}_2}^{\mathsf{T}} \mathsf{F} \mathsf{K}_1 \tag{2.21}$$

L'équation 2.20 devient dans ce cas :

$$\tilde{\mathbf{q}}_2{}^{\mathsf{T}} \left({\mathsf{K}_2}^{-\mathsf{T}} \mathsf{E} {\mathsf{K}_1}^{-1} \right) \tilde{\mathbf{q}}_1 = 0 \tag{2.22}$$

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

2.3.1.3 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 $(\mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$. Une formalisation en sera faite à la section 2.3.2.1. La relation entre E, $\mathsf{R}_{1\to2}$ et $\mathbf{t}_{1\to2}$ s'écrit :

$$\mathsf{E} = [\mathbf{t}_{1\to2}]_\times \mathsf{R}_{1\to2} \tag{2.23}$$

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

$$[\mathbf{t}]_\times = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{2.24}$$

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 2.3.2.2.

2.3.2 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.

2.3.2.1 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 :

$$\tilde{\mathcal{Q}}_{\mathcal{C}_1} \sim \begin{pmatrix} \mathsf{R}_1{}^\mathsf{T} & -\mathsf{R}_1{}^\mathsf{T}\mathbf{t_1} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{W}} \tag{2.25}$$

$$\tilde{\mathcal{Q}}_{\mathcal{C}_2} \sim \begin{pmatrix} \mathsf{R}_2{}^\mathsf{T} & -\mathsf{R}_2{}^\mathsf{T}\mathbf{t_2} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{W}} \tag{2.26}$$

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 $(\mathsf{R}_{1\to 2}, \mathbf{t}_{1\to 2})$ 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 $\mathcal{C}_2$:

$$\tilde{\mathcal{Q}}_{\mathcal{C}_2} \sim \begin{pmatrix} \mathsf{R}_{1\to 2} & \mathbf{t}_{1\to 2} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{C}_1} \tag{2.27}$$

Des équations 2.25 et 2.26, on peut obtenir le système d'équations suivant :

$$\begin{cases} \tilde{\mathcal{Q}}_{\mathcal{W}} \sim \begin{pmatrix} \mathsf{R}_1 & \mathbf{t_1} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{C}_1} \\ \tilde{\mathcal{Q}}_{\mathcal{C}_2} \sim \begin{pmatrix} \mathsf{R}_2{}^\mathsf{T} & -\mathsf{R}_2{}^\mathsf{T}\mathbf{t_2} \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{W}} \end{cases} \tag{2.28}$$

d'où

$$\tilde{\mathcal{Q}}_{\mathcal{C}_2} \sim \begin{pmatrix} \mathsf{R}_2{}^\mathsf{T}\mathsf{R}_1 & \mathsf{R}_2{}^\mathsf{T}(\mathbf{t}_1 - \mathbf{t}_2) \\ 0_{1\times 3} & 1 \end{pmatrix} \tilde{\mathcal{Q}}_{\mathcal{C}_1} \tag{2.29}$$

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

$$\begin{cases} \mathsf{R}_{1\to2} = \mathsf{R}_2{}^{\mathsf{T}}\mathsf{R}_1 \\ \mathbf{t}_{1\to2} = \mathsf{R}_2{}^{\mathsf{T}}(\mathbf{t}_1 - \mathbf{t}_2) \end{cases} \tag{2.30}$$

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

Lorsque la structure de l'environnement est inconnue, 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 2.3.1), ceci permet de calculer la matrice fondamentale (algorithme des 8 points, Hartley (1997)) ou essentielle (algorithme des 5 points, Nistér (2004)). Il est alors possible d'extraire d'une de ces matrices le déplacement inter-caméra $(\mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$.

Dans le cas de caméras non-calibrées, $\mathsf{R}_{1\to2}$ et $\mathbf{t}_{1\to2}$ 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 $\mathsf{R}_{1\to2}$ 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 Nistér (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 :

$$\mathsf{S} \sim \begin{pmatrix} s\mathsf{R} & \mathbf{t} \\ 0_{1\times3} & 1 \end{pmatrix} \tag{2.31}$$

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

2.3.2.3 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 2.2.2). 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, la triangulation du point 3D peut par exemple être considéré comme étant le point équidistant des deux rayons (figure 2.3).

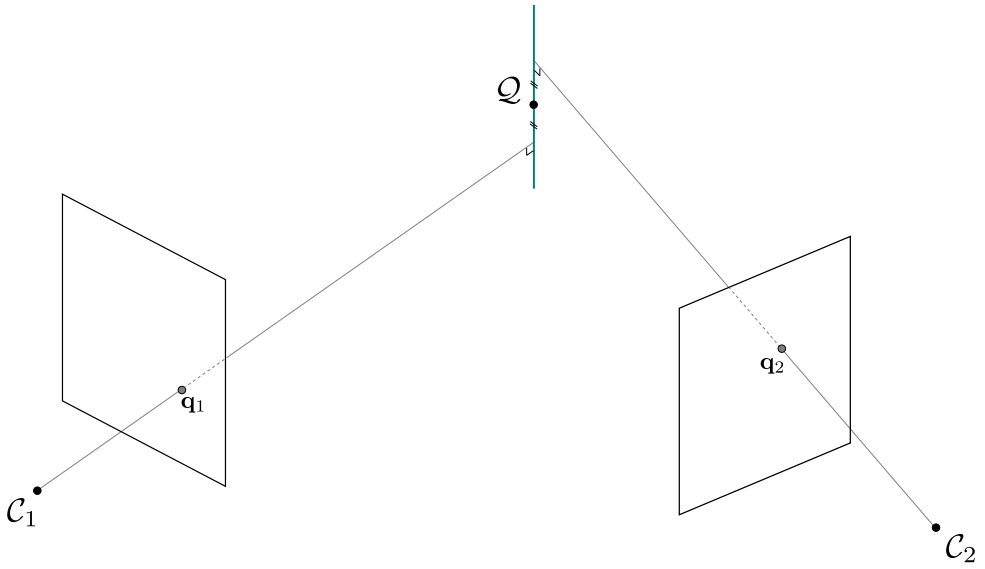


FIGURE 2.3 – **Triangulation de points 3D.** La structure de l'environnement peut être obtenue par triangulation des observations dans les images.

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)).

2.3.2.4 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.

2.3.2.5 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 2.4). 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 :

$$r = ||\mathbf{q} - \pi(\tilde{\mathsf{P}}\tilde{\mathcal{Q}})|| \tag{2.32}$$

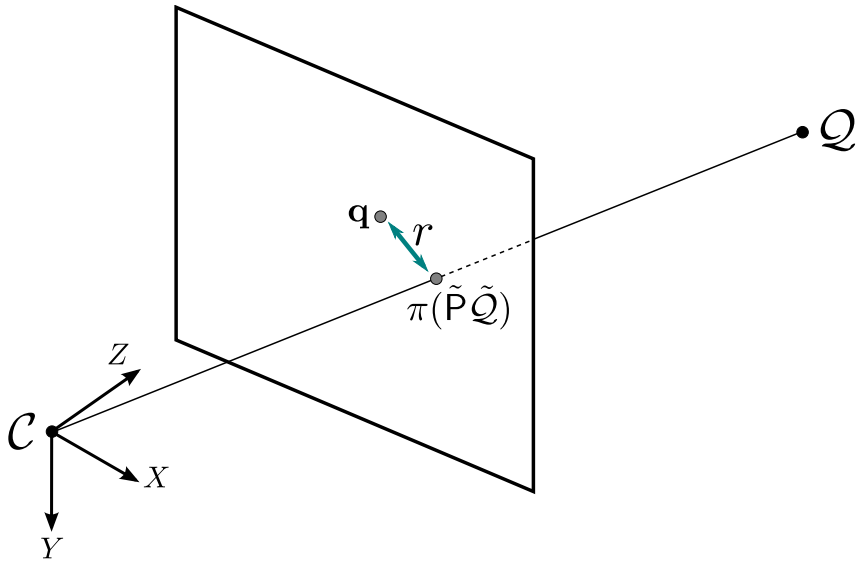


FIGURE 2.4 – **Erreur de reprojection.** L'erreur de reprojection est la distance entre l'observation $\mathbf{q}$ d'un point $\mathcal{Q}$ et sa projection dans l'image $\pi(\tilde{\mathsf{P}}\tilde{\mathcal{Q}})$.

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 6 paramètres de pose de chaque caméra et les 3 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 :

$$\mathcal{F}(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E, \mathcal{Q}^1, \ldots, \mathcal{Q}^M) = \sum_{1 \leq j \leq N} \sum_{i \in \mathcal{A}_j} ||\mathbf{q}_j^i - \pi(\tilde{\mathsf{P}}_j \tilde{\mathcal{Q}}^i)||^2 \tag{2.33}$$

où les $(\mathsf{C}_i^E)_i$ sont les 6 paramètres extrinsèques et $(\tilde{\mathsf{P}}_j)_j$ les matrices de projection des caméras, $(\mathcal{Q}^i)_i$ les points 3D et $\mathbf{q}_j^i$ l'observation du point i dans la caméra j. L'ensemble $\mathcal{A}_j$ contient

l'ensemble des indices des points 3D vus par la caméra j. Afin de minimiser cette fonction de coût, on utilisera un algorithme de minimisation non-linéaire. Ce type d'algorithme sera décrit dans la section 2.5.3.

2.4 Cas d'une scène plane

Dans cette section, nous présentons ce que deviennent les relations qui existent entre deux caméras dans le cas où la scène observée est planaire.

2.4.1 Homographies 2D

Une *homographie* $\mathcal{H}$ (ou *transformation projective*) 2D est 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 $\mathcal{H} : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ 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 $\mathbb{P}^2$, $\mathcal{H}(\tilde{\mathbf{q}}) = \mathsf{H}\tilde{\mathbf{q}}$.*

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 2.5. 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 Π, 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 :

$$\begin{aligned} \tilde{\mathbf{q}}_2 &\sim \mathsf{H}_{1\rightarrow 2}\tilde{\mathbf{q}}_1 \\ &\sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tilde{\mathbf{q}}_1 \end{aligned} \tag{2.34}$$

2.4.2 Relation entre homographie 2D, équation de plan et déplacement relatif

Le but de cette partie est de montrer qu'il est possible de définir une relation entre le déplacement relatif des caméras $(\mathsf{R}_{1\rightarrow 2}, \mathbf{t}_{1\rightarrow 2})$, l'équation du plan observé Π et l'homographie liant les observations de ce plan dans les 2 images.

Pour cela, nous nous plaçons dans le cadre d'étude présenté par la figure 2.5. Deux caméras $\mathcal{C}_1$ et $\mathcal{C}_2$ observent un même plan Π. La normale de ce plan, dans le repère lié à $\mathcal{C}_1$, est notée $\mathbf{n}$ et l'équation du plan Π dans ce même repère est $(\mathbf{n}\ d)^\mathsf{T}$, où d est la distance de $\mathcal{C}_1$ au plan.

Les coordonnées de $\mathbf{q}_1$ sont exprimées dans le repère image (figure 2.1, page 19). Notons $\mathbf{m}_1$ les coordonnées du point $\mathbf{q}_1$ dans le repère de la rétine :

$$\tilde{\mathbf{m}}_1 \sim {\mathsf{K}_1}^{-1}\tilde{\mathbf{q}}_1 \tag{2.35}$$

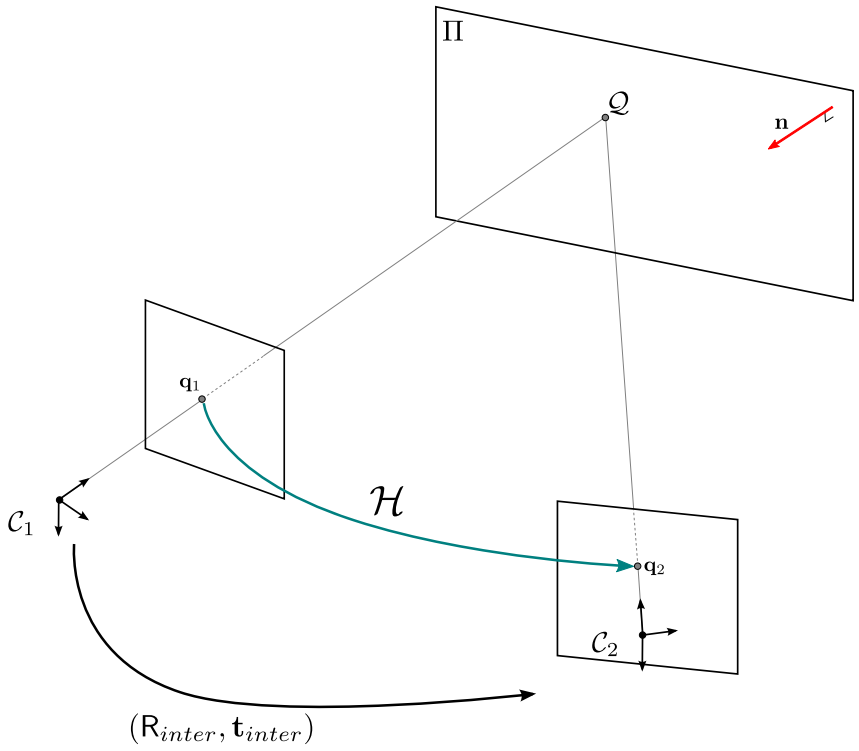


FIGURE 2.5 – **Homographies 2D.** Les coordonnées des observations correspondantes de points 3D situés sur un même plan de l'espace sont reliées par une homographie 2D.

Dans ce repère, la position 3D homogène de $\tilde{\mathcal{Q}}$ s'écrit :

$$\tilde{\mathcal{Q}} = \begin{pmatrix} \tilde{\mathbf{m}}_1 \\ \rho \end{pmatrix} \tag{2.36}$$

Puisque le point $\mathcal{Q}$ est situé sur le plan, il doit en respecter l'équation :

$$\begin{pmatrix} \mathbf{n} \\ d \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \tilde{\mathbf{m}}_1 \\ \rho \end{pmatrix} = 0 \tag{2.37}$$

d'où on déduit que

$$\rho = -\frac{\mathbf{n}^{\mathsf{T}} \tilde{\mathbf{m}}_1}{d} \tag{2.38}$$

A partir de l'équation 2.27, on peut alors obtenir la projection de $\mathcal{Q}$ dans le plan rétine de la caméra 2 :

$$\begin{aligned} \tilde{\mathbf{m}}_2 &\sim \begin{pmatrix} \mathsf{R}_{1\to 2} & \mathbf{t}_{1\to 2} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{m}}_1 \\ \rho \end{pmatrix} \\ &\sim (\mathsf{R}_{1\to 2} - \frac{\mathbf{t}_{1\to 2}\mathbf{n}^{\mathsf{T}}}{d}) \tilde{\mathbf{m}}_1 \end{aligned} \tag{2.39}$$

On peut revenir à la relation entre les points $\tilde{\mathbf{q}}_1$ et $\tilde{\mathbf{q}}_2$, c'est à dire aux points 2D exprimés dans le repère image :

$$\tilde{\mathbf{q}}_2 \sim \mathsf{K}_2 (\mathsf{R}_{1\to 2} - \frac{\mathbf{t}_{1\to 2}\mathbf{n}^{\mathsf{T}}}{d}) \mathsf{K}_1{}^{-1} \tilde{\mathbf{q}}_1 \tag{2.40}$$

La relation entre homographie 2D, équation du plan et déplacement inter-caméra s'écrit donc :

$$\begin{aligned} \mathsf{H}_{1\to2} &\sim \mathsf{K}_2(\mathsf{R}_{1\to2} - \frac{\mathbf{t}_{1\to2}\mathbf{n}^\mathsf{T}}{d})\mathsf{K}_1{}^{-1} \\ &\sim \mathsf{K}_2(\mathsf{R}_2{}^\mathsf{T}\mathsf{R}_1 - \frac{\mathsf{R}_2{}^\mathsf{T}(\mathbf{t}_1 - \mathbf{t}_2)\mathbf{n}^\mathsf{T}}{d})\mathsf{K}_1{}^{-1} \end{aligned} \tag{2.41}$$

L'équation 2.41 est en particulier couramment utilisée pour estimer le déplacement d'une caméra. En effet, à partir de l'estimation de la matrice H (obtenue par exemple avec la méthode DLT détaillée dans le livre de Hartley and Zisserman (2004)), Faugeras (1993) a montré qu'il est possible d'extraire les 6 paramètres $(\mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$ du déplacement relatif et 2 paramètres liés au plan (2 paramètres suffisent pour la normale, celle-ci étant normée). Il est à noter qu'il y a une ambiguité entre la norme de $\mathbf{t}_{1\to2}$ et la distance au plan d. Cela revient à dire que, comme pour la matrice essentielle, la norme du déplacement entre les 2 caméras n'est pas directement estimable. Pour cela, il est nécessaire de connaître *a priori* la distance d.

2.5 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.

2.5.1 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 :

$$\mathcal{F}(\widehat{\mathbf{x}}) = \mathbf{y} \tag{2.42}$$

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 :

$$r(\mathbf{x}) = \mathcal{F}(\mathbf{x}) - \mathbf{y} \tag{2.43}$$

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 :

$$\widehat{\mathbf{x}} = \underset{\mathbf{x}}{argmin}\, \varepsilon(\mathbf{x}) \tag{2.44}$$

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

$$\begin{aligned} \varepsilon(\mathbf{x}) &= ||\mathcal{F}(\mathbf{x}) - \mathbf{y}||^2 \\ &= ||r(\mathbf{x})||^2 \\ &= \sum_i ||r_i(\mathbf{x})||^2 \end{aligned} \tag{2.45}$$

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 $\mathcal{F}$.

2.5.2 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 :

$$\varepsilon(\mathbf{x}) = ||\mathsf{F}\mathbf{x}||^2 \tag{2.46}$$

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 (Singular Value Decomposition) 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 :

$$\varepsilon(\mathbf{x}) = ||\mathsf{F}\mathbf{x} - \mathbf{y}||^2 \tag{2.47}$$

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 2.17) :

$$\widehat{\mathbf{x}} = \mathsf{F}^{+}\mathbf{y} \tag{2.48}$$

2.5.3 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. En fonction de la méthode retenue, l'hypothèse faite pour permettre la résolution est que la fonction ε est localement linéaire ou quadratique. 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 et la longueur de pas 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. La longueur de pas est généralement fixée à 1. L'avantage de cette approche est qu'elle converge efficacement lorsque la solution initiale est éloignée du minimum recherché.

▷ **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 et la longueur de l'incrément 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.

2.5.4 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, différentes approches ont été proposées.

2.5.4.1 RANSAC et LMedS

Les méthodes *RANSAC* (RANdom SAmple Consensus, Fischler and Bolles (1981)) et *LMedS* (Least Median of Squares, Rousseeuw and Leroy (1987)) sont des méthodes robustes où l'idée est de trouver un sous-ensemble des M mesures $\mathbf{y}$, de taille N (où N est le nombre minimum de mesures nécessaires à la résolution du problème posé), qui offre la meilleure estimation des paramètres $\widehat{\mathbf{x}}$. Les deux méthodes s'appuient sur le même algorithme. Cet algorithme et les notations qui lui sont associées sont décrits dans le tableau 2.1. Néanmoins, les deux méthodes diffèrent sur la définition de l'optimalité des paramètres $\widehat{\mathbf{x}}$.

RANdom SAmple Consensus (RANSAC). La définition de l'optimalité de $\widehat{\mathbf{x}}$ pour la méthode RANSAC est la suivante :

▷ La mesure de qualité est la taille du *support* correspondant au jeu de paramètres $\widehat{\mathbf{x}}_i$. Le support $\mathcal{S}_i$ est l'ensemble des mesures, parmi les M mesures $\mathbf{y}$, qui sont satisfaites par les paramètres, c'est à dire pour lesquelles $||\mathcal{F}_j(\widehat{\mathbf{x}}_i) - y_j|| < \xi$ où ξ est un seuil à définir.

▷ Le tirage finalement retenu est celui qui donne le support le plus grand : $i_f = \underset{i}{argmax}\, \text{card}(\mathcal{S}_i)$ où $\text{card}(\mathcal{S}_i)$ est le cardinal de l'ensemble $\mathcal{S}_i$.

La performance de l'algorithme RANSAC est directement liée au fait de pouvoir définir le seuil ξ avec précision de façon à filtrer les points aberrants et ceux qui ne le sont pas.

Least Median of Squares (LMedS). L'approche LMedS permet de s'affranchir de ce seuil lorsque le taux de points non-aberrants est supérieur à 50%. Alors que l'algorithme de RANSAC vise à maximiser la taille du support, c'est à dire le nombre de points non-aberrants, la méthode LMedS a pour but de minimiser l'erreur médiane des résidus. Le critère d'optimalité pour le LMedS s'écrit donc comme suit :

▷ La mesure de qualité est ici le calcul de la médiane des résidus : $e_i = \text{med}_j ||\mathcal{F}_j(\widehat{\mathbf{x}}_i) - y_j||$.

▷ La sélection du meilleur tirage se fait en sélectionnant celui qui donne l'erreur la plus faible : $i_f = \underset{i}{argmin}\ e_i$.

L'hypothèse de 50% de points non-aberrants peut être remplacée par n'importe quel autre pourcentage en remplaçant l'utilisation de la médiane par la mesure adaptée. On ne parle alors plus de *minimisation aux moindres médians* mais de *minimisation aux moindres quantiles*. Notons que la qualité de l'estimation réalisée grâce au LMedS est fonction de la précision avec laquelle la valeur du quantile a été fixée.

RANdom SAmple Consensus (RANSAC) / Least Median of Squares (LMedS)

▷ **Estimation du nombre de tirages nécessaires.** On calcule tout d'abord le nombre de tirages à réaliser parmi les mesures $\mathbf{y}$. En effet, afin de diminuer les temps de calcul, tous les sous-ensembles de N mesures ne seront pas testés. Une estimation du ratio entre points aberrants et points corrects permet d'obtenir le nombre de tirages à réaliser afin de s'assurer de trouver la solution optimale avec une probabilité choisie (voir Fischler and Bolles (1981)).

▷ Pour chacun des tirages, indicés i :
- **Tirage.** On tire un ensemble de N mesures parmi les M mesures de départ $\mathbf{y}$. Cet ensemble est noté $\mathcal{Y}_i$.
- **Résolution.** On résout l'équation décrivant le problème étudié (à partir d'une des méthodes décrites aux sections 2.5.2 et 2.5.3) pour d'obtenir une solution $\widehat{\mathbf{x}}_i$ à partir des mesures $\mathcal{Y}_i$.
- **Mesure de qualité.** On mesure alors la qualité de l'estimation des paramètres $\widehat{\mathbf{x}}_i$. Ce critère dépend de la méthode choisie (RANSAC ou LMedS).

▷ **Sélection du meilleur tirage.** On garde le tirage, indicé i_f, qui optimise la mesure de qualité définie.

▷ **Calcul des paramètres $\widehat{\mathbf{x}}$.** Les paramètres $\widehat{\mathbf{x}}$ sont alors calculés à partir de toutes les mesures qui respectent le critère de qualité pour la valeur $\widehat{\mathbf{x}}_{i_f}$ des paramètres. Ces mesures, considérées non-aberrantes, sont également appelées *inliers*.

TABLE 2.1 – Description simplifiée de l'algorithme à la base de RANSAC et LMedS

2.5.4.2 M-estimateurs

Nous avons vu à l'équation 2.45 que 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 2.6(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 :

$$\varepsilon(\mathbf{x}) = \sum_i \rho(r_i(\mathbf{x})) \tag{2.49}$$

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 2.6(b)) rend l'influence des résidus constante pour tous les points aberrants :

$$\rho_{Tukey}(x) = \begin{cases} \dfrac{\sigma^2}{2}\left(1 - \left[1 - \left(\dfrac{x}{c}\right)^2\right]^3\right) & \text{si } |x| \leq \sigma \\ \dfrac{c^2}{6} & \text{sinon} \end{cases} \tag{2.50}$$

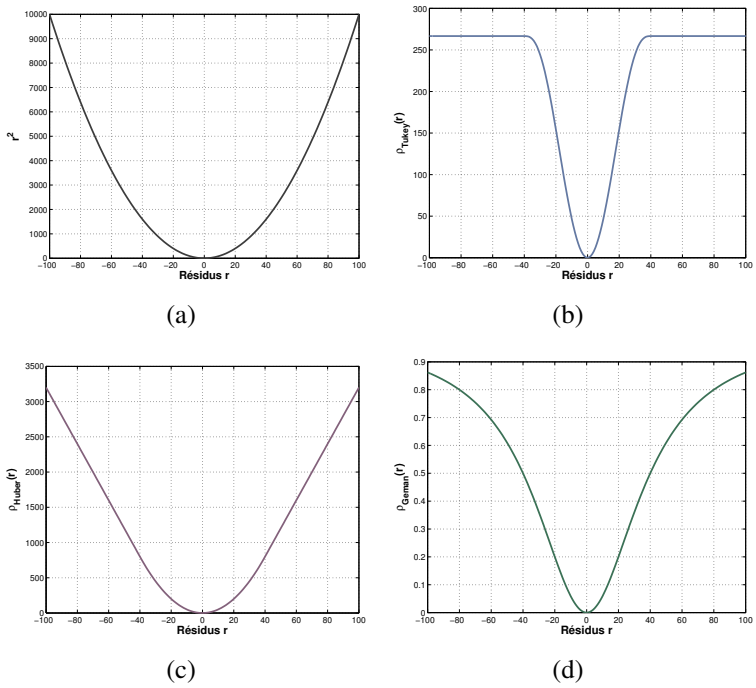
▷ Avec le M-estimateur de Huber (figure 2.6(c)), l'évolution du poids des points aberrants est linéaire avec leur résidu :

$$\rho_{Huber}(x) = \begin{cases} \dfrac{x^2}{2} & \text{si } |x| \leq \sigma \\ \sigma(|x| - \dfrac{\sigma}{2}) & \text{sinon} \end{cases} \tag{2.51}$$

▷ Enfin, le M-estimateur de Geman-McClure (figure 2.6(d)) donne une évolution asymptotique à l'influence des résidus des points aberrants :

$$\rho_{Geman}(x) = \frac{x^2}{x^2 + c^2} \tag{2.52}$$

Pour tous les M-estimateurs, il est nécessaire de régler le seuil σ 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 2.6 – **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 $\sigma = 40$.

2.6 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 modèles 3D de villes que nous utilisons.

2.6.1 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 dans la figure 2.7). Comme cela a été décrit dans la section 1.1, 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.

Sans entrer dans les détails d'implémentation, nous allons détailler ci-après 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.

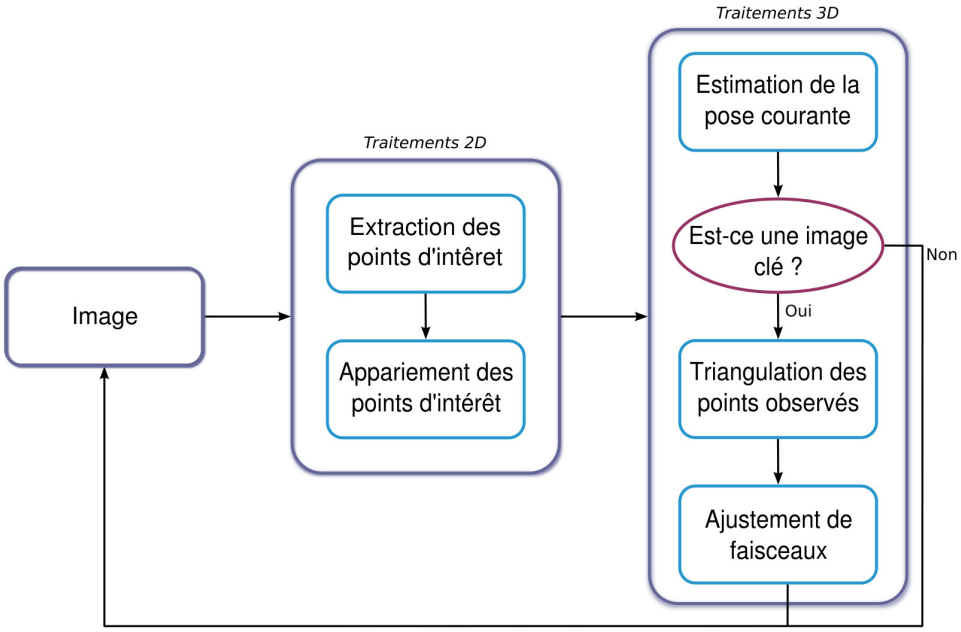


FIGURE 2.7 – **Schéma du fonctionnement du SLAM de Mouragnon et al. (2006).**

2.6.1.1 Traitements 2D

Le but des traitements 2D (c'est à dire au niveau des images) de la méthode d'odométrie visuelle 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 le descripteur 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. Le critère proposé par Mouragnon et al. (2006) est un critère essentiellement 2D : une image est déclarée comme étant clé lorsque le nombre d'appariements 2D entre cette image et la dernière image clé est inférieur à un seuil M ($M = 400$ points ici).

2.6.1.2 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.

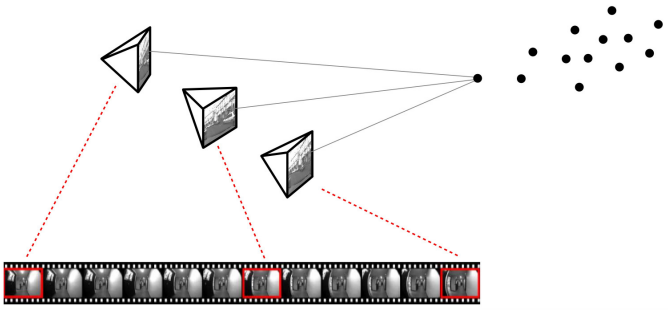
Initialisation. Au début de la reconstruction, seules les informations calculées par le suivi 2D sont disponibles. En particulier, aucune information sur la structure de l'environnement n'est fournie. La première étape de la reconstruction 3D est donc d'initialiser la carte de l'environnement, c'est à dire la pose des premières caméras et la position des points 3D observés (figure 2.8(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 2.3.2.2). Une fois la reconstruction initialisée, le processus incrémental est lancé.

Traitement des caméras. 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 2.8(b)) peut alors être estimée à partir de ces appariements (section 2.3.2.4).

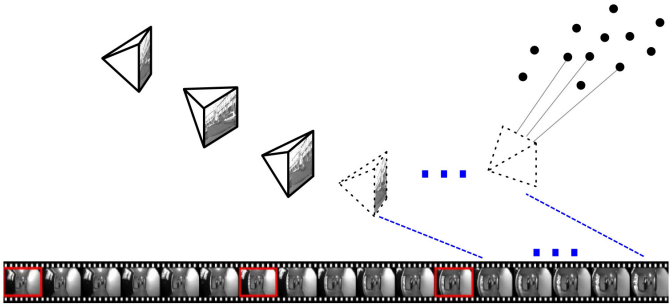
Si la caméra courante est détectée comme étant une caméra clé, elle est alors utilisée pour augmenter la reconstruction de l'environnement (figure 2.8(c)) :

- ▷ Sa pose et ses observations sont utilisées pour trianguler de nouveaux points 3D (section 2.3.2.3).
- ▷ Un ajustement de faisceaux est appliqué pour raffiner la géométrie de la reconstruction.

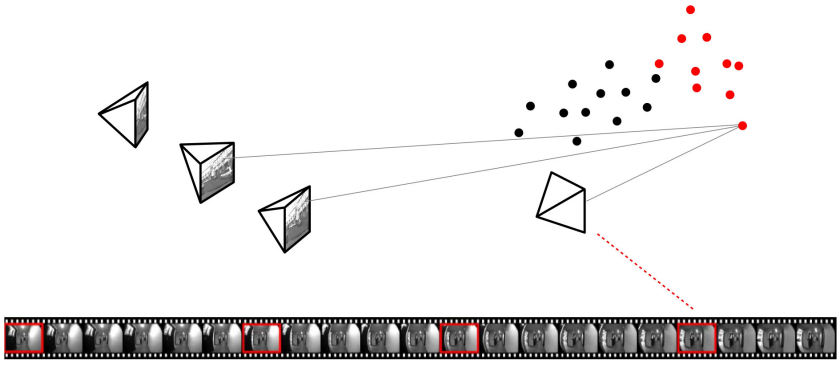
La particularité des travaux de Mouragnon et al. (2006) est que l'ajustement de faisceaux ne raffine pas l'ensemble de la reconstruction. En effet, afin d'assurer un traitement temps-réel, l'ajustement de faisceaux est uniquement réalisé sur une sous-partie de la reconstruction. Cette sous-partie est constituée des M dernières caméras clés ($M = 20$ dans notre étude) et des points 3D associés. De plus, parmi cette structure, seuls les paramètres des N dernières caméras clés (dans notre cas $N = 3$) et des points 3D qu'elles observent sont raffinés. Les $M - N$



(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.

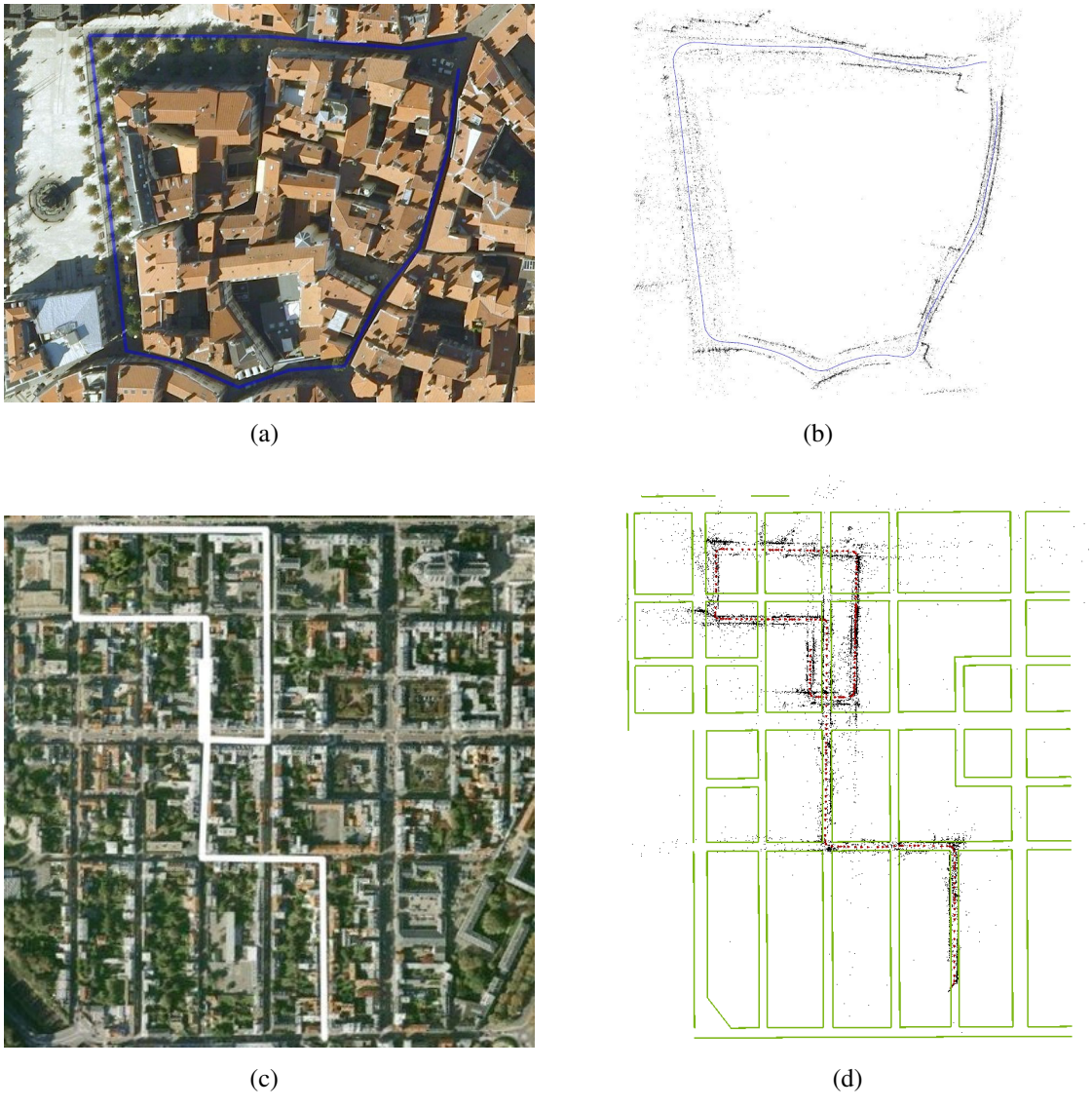


(c) **Création d'une nouvelle caméra clé.** Si la caméra courante est clé, elle est alors utilisée pour trianguler de nouveaux points 3D. Les paramètres des 3 dernières caméras clés et des points 3D qu'elles observent sont alors raffinés à l'aide d'un ajustement de faisceaux.

FIGURE 2.8 – **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.

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 2.9.



(a) (b)

(c) (d)

FIGURE 2.9 – **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).

Cependant, et comme nous l'avons détaillé à la section 1.1.3, 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 modèles 3D de villes qui seront utilisés dans notre méthode

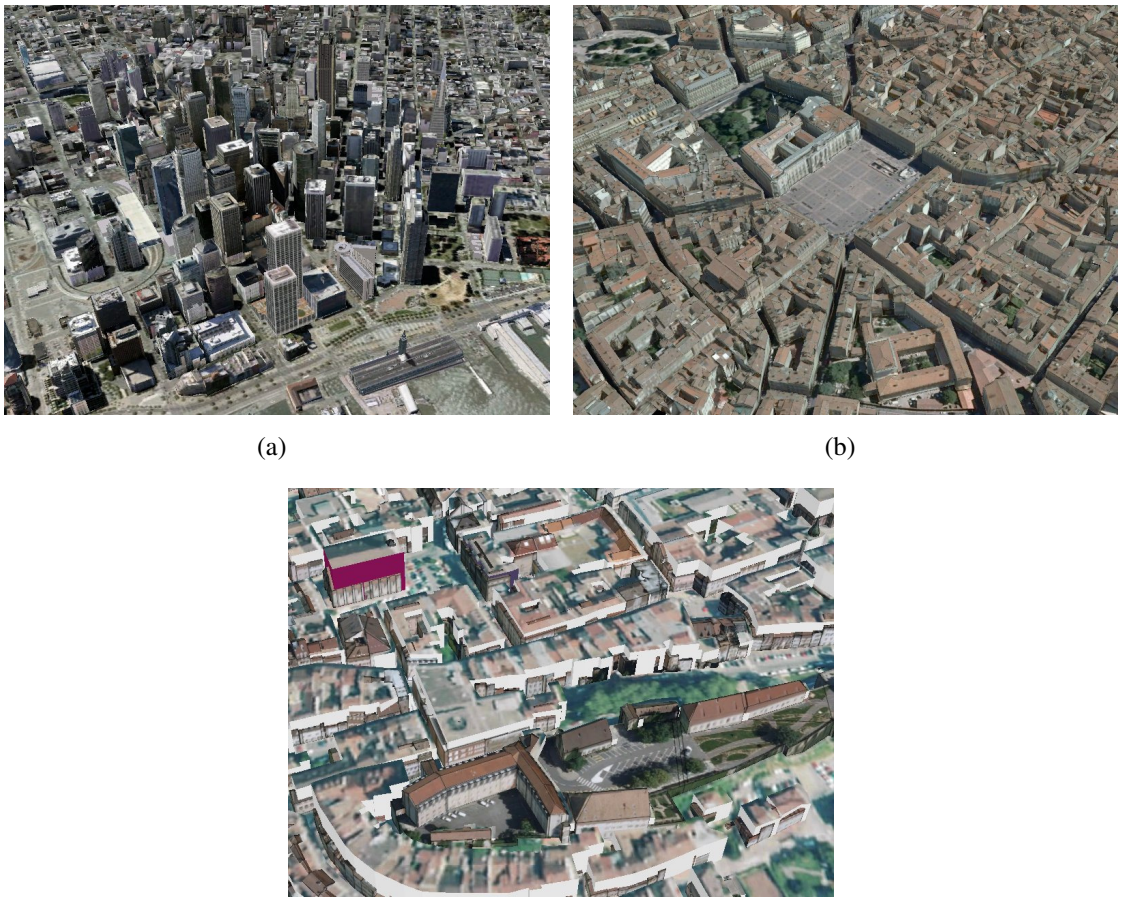
afin de corriger ces dérives.

2.6.2 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.

2.6.2.1 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, route, 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 2.10).



(a) (b)

(c)

FIGURE 2.10 – **Exemples de SIG.** (a) Google Earth, (b) Microsoft Bing Map 3D et (c) TerraExplorer (interface du Géoportail) sont parmi les SIG les plus consultés.

2.6.2.2 Les modèles 3D à grande échelle.

Les modèles 3D disponibles dans les SIG ont différentes provenances. Ils sont majoritairement issus d'instituts nationaux (par exemple l'IGN [1] pour le Géoportail [2] 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 [3] 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. 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 :

- ▷ **Simplification de la géométrie.** La géométrie des modèles 3D ne détaille que la structure globale des bâtiments. Ainsi, la surface des façades est discrétisée en un nombre fini de plans 3D, en général un seul plan par façade. En particulier, les portes, les fenêtres et les colonnades sont absentes en 3D et n'apparaissent que sur la texture du modèle. 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.

- ▷ **Texture uniquement photoréaliste.** Actuellement, les textures associées aux modèles 3D sont de faible qualité (figure 2.11), par exemple afin de limiter la taille du SIG. De plus, leur recalage sur les modèles est généralement très grossier. Ceci implique que l'information de texture n'est pas utilisable dans notre contexte.

Il est important de noter que les modèles décrits ci-dessus correspondent à l'état des modèles au début de nos travaux, en 2007. Au jour d'aujourd'hui, la géométrie tend à s'améliorer pour atteindre dans certains endroits une précision de 10 cm (Hyères et Montbéliard par exemple pour le Géoportail). On peut raisonnablement imaginer que la précision des modèles va continuer à s'améliorer tandis que leur disponibilité va croître de façon importante. L'intérêt de l'utilisation de modèles plus précis sera étudié dans les perspectives de nos travaux (page 158).

2.6.2.3 Caractéristiques des modèles 3D utilisés

La majorité de nos expériences se situent dans les rues de Versailles (figure 2.12(a)). Ne disposant pas, au début de nos travaux, de modèles 3D de cet environnement, nous avons utilisé ceux provenant du site du Géoportail. Plus précisément, nous avons utilisé les outils de mesure disponibles dans l'interface graphique associée au Géoportail afin de recréer le modèle 3D du quartier qui nous intéresse. Le modèle obtenu est affiché sur la figure 2.12(b).

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

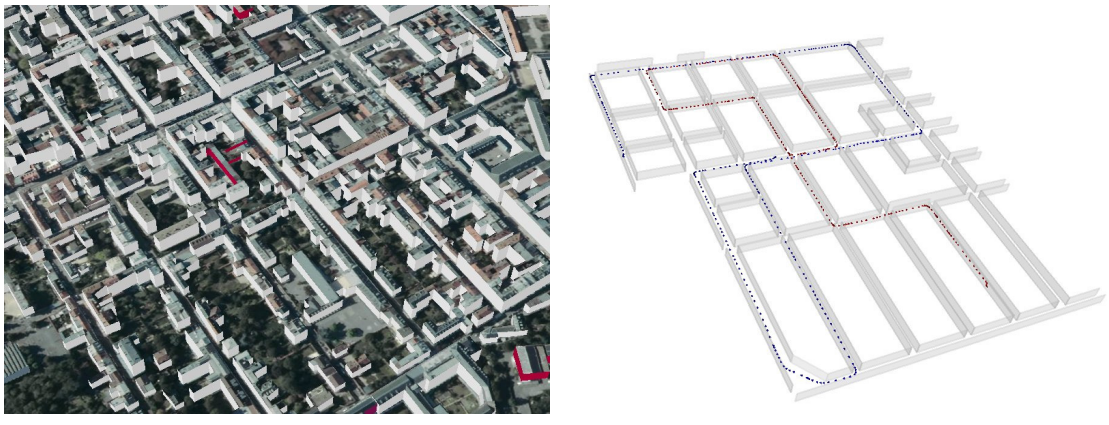


(a) (b)

FIGURE 2.11 – **Modèles 3D texturés.** La texture des modèles 3D est généralement de faible qualité *(Google Earth)*.

Les modèles du Géoportail, pour Versailles, ont une précision de l'ordre de 1 mètre. Ayant utilisé les outils de mesure, nous pensons qu'il est raisonnable de considérer que notre modèle 3D a une précision de l'ordre de 2 mètres. Comme nous l'avons vu précédemment, la géométrie locale (c'est à dire les portes, balcons, *etc.*) est absente. La reconstruction 3D issue du SLAM ne pourra donc pas s'aligner parfaitement avec le modèle 3D. Afin de modéliser cette erreur, nous faisons l'hypothèse que la distance d_r entre le modèle 3D et l'environnement réel (c'est à dire la distance orthogonale entre le point sur la surface du modèle et sa position réelle dans la scène) est une variable gaussienne de faible écart-type σ (dans l'idéal moins de 2 mètres) et centrée en l'origine (c'est à dire d'espérance nulle). Sa densité de probabilité p peut s'écrire :

$$p(d_r) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{1}{2}(\frac{d_r}{\sigma})^2} \tag{2.53}$$



(a) (b)

FIGURE 2.12 – **Modèle 3D utilisé dans nos travaux.** (a) est le quartier de Versailles que nous avons modélisé. (b) est le modèle 3D associé à cet environnement dans lequel sont représentées deux trajectoires reconstruites.

Les différentes observations et hypothèses faites sur les modèles 3D utilisés seront nos hypothèses de travail pour l'ensemble de ce mémoire.

Première partie

Création d’une base d’amers visuels et relocalisation d’une caméra mobile

---

Contenu de la partie

Présentation de la méthode **47**

3 ICP non-rigide **51**
3.1 Méthodes d'alignement 3D 51
3.2 Espace de transformations utilisé 52
3.3 Recherche de l'alignement optimal 56
3.4 Discussion 59

4 Ajustements de faisceaux contraints par un SIG **61**
4.1 Méthodes classiques et limites 61
4.2 Approche proposée : ST-CBA 63

5 Résultats expérimentaux **69**
5.1 Evaluation quantitative sur des données de synthèse 69
5.2 Evaluation qualitative sur des données réelles 77
5.3 Discussion 82

6 Relocalisation et réalité augmentée **87**
6.1 Présentation de l'application visée 87
6.2 Localisation absolue dans la base d'amers 87
6.3 Vers une application d'aide à la navigation 89
6.4 Discussion 92

Présentation de la méthode

La première partie de ce mémoire a pour but de présenter une chaîne complète permettant la relocalisation d'une caméra dans un centre urbain dense. La méthode proposée est schématisée dans la figure 2.13. Le processus complet présente deux sous-processus distincts : la construction de la base d'amers (hors-ligne) et la relocalisation dans cette base (en ligne).

Construction de la base d'amers visuels

Problématique

De nombreuses méthodes de relocalisation reposent sur la disponibilité d'un modèle de l'environnement constitué d'amers visuels, c'est à dire d'un nuage de points 3D pour lesquels on dispose d'une description de leur apparence photométrique. Ce modèle constitue une base de données à partir de laquelle il est alors possible de relocaliser une caméra mobile. Cette idée a déjà été largement utilisée, particulièrement en robotique pour la localisation et la navigation autonome. Généralement, la base de données d'amers est construite à partir d'un algorithme de type Structure from Motion (Royer et al. (2007); Irschara et al. (2009)). Cependant, la base d'amers ainsi obtenue n'est pas exempte de défauts. En effet, celle-ci est exprimée dans un repère arbitraire et, dans le cas de SfM monoculaire, à une échelle arbitraire. Une localisation estimée à partir d'une telle base d'amers ne peut donc pas fournir une localisation géoréférencée. De plus, comme nous l'avons vu précédemment (section 1.1.3), ces reconstructions sont sensibles aux dérives, ce qui rend la base incohérente à grande échelle. Cette incohérence a peu d'influence lorsque l'information recherchée est uniquement un déplacement relatif, par exemple pour le suivi de convois. Néanmoins, l'impact devient important lorsque l'information recherchée est une localisation absolue. Or, le but de nos travaux étant de localiser un véhicule dans un centre urbain, c'est ce type d'information absolue qui nous intéresse. C'est en ce sens que nous proposons dans cette partie une méthode permettant de corriger *a posteriori* une reconstruction SLAM en milieu urbain afin de la rendre exploitable par un processus de relocalisation absolue.

Limites des méthodes existantes

Comme cela a été vu dans la section 1.2.2, plusieurs méthodes existent dans le cas où les reconstructions sont supposées sans dérive. La correction se limite alors à corriger le facteur d'échelle et à géoréférencer la base d'amers résultante. Par exemple, il a récemment été proposé des méthodes permettant d'aligner les nuages de points reconstruits avec une image satellite (Kaminsky et al. (2009)) ou un modèle 3D de l'environnement (Strecha et al. (2010)). Néanmoins, dans ces différentes approches, la géométrie de la reconstruction 3D n'est jamais remise en cause.

Plusieurs méthodes ont déjà été proposées pour à la fois géoréférencer et corriger la reconstruction SLAM lorsque celle-ci présente une dérive importante. Tout d'abord, Clemente et al. (2007) proposent d'utiliser la détection et la correction de boucles afin de corriger la reconstruction SLAM sans l'apport d'information supplémentaire. Cependant, cette méthode de correction n'est applicable que dans les contextes particuliers où la trajectoire repasse plusieurs fois au même endroit. D'autres travaux ont été proposés dans le cas où une information grossière sur l'environnement parcouru est disponible. En particulier, les problématiques rencontrées par Levin and Szeliski (2004) sont proches des nôtres. Leur idée est d'utiliser une carte simple (dans leur cas dessinée à la main) de la trajectoire parcourue pour corriger *a posteriori* la dérive subie par leur méthode de SLAM. Pour cela, ils proposent tout d'abord d'aligner grossièrement la reconstruction avec la carte associée en utilisant une transformation simple. Pour raffiner la reconstruction obtenue, ils appliquent alors un ajustement de faisceaux global. On notera que cette deuxième étape n'intègre aucune information issue de la carte. En l'absence de ces contraintes supplémentaires, l'ajustement de faisceaux global peut converger vers une solution incohérente avec la carte, voire revenir à la solution initiale.

Approche proposée

Afin de corriger les reconstructions SLAM, nous proposons d'utiliser les contraintes géométriques apportées par un modèle 3D simple de l'environnement, comme décrit à la section 2.6.2.3. Notre approche consiste à estimer la transformation permettant d'aligner les points 3D de la base d'amers appartenant à des façades de bâtiments avec les plans représentant ces même façades dans le modèle. La dérive de la reconstruction SLAM étant généralement trop complexe pour que cette transformation soit estimée directement, notre méthode de correction se décompose en 2 étapes progressives :

- ▷ **ICP non-rigide.** La première étape (décrite au chapitre 3) consiste à appliquer à la reconstruction 3D une transformation simplifiée de façon à retrouver sa cohérence globale. Pour cela, nous proposons tout d'abord une classe de transformations décrivant au mieux les déformations induites par le processus du SLAM. Les paramètres de la transformation recherchée sont alors estimés à l'aide d'un algorithme ICP (Iterative Closest Point).

- ▷ **Nouvel ajustement de faisceaux.** La seconde étape (décrite au chapitre 4) consiste à raffiner la correction. Pour cela, un modèle de transformation plus complexe est optimisé à l'aide d'un nouvel ajustement de faisceaux. Cet ajustement de faisceaux intègre à la fois des contraintes visuelles, c'est à dire l'erreur de reprojection des points observés, et la contrainte d'appartenance des points 3D au modèle.

L'ensemble de ce processus sera testé sur des séquences de synthèse et réelles au chapitre 5 afin d'évaluer qualitativement et quantitativement sa précision et sa robustesse.

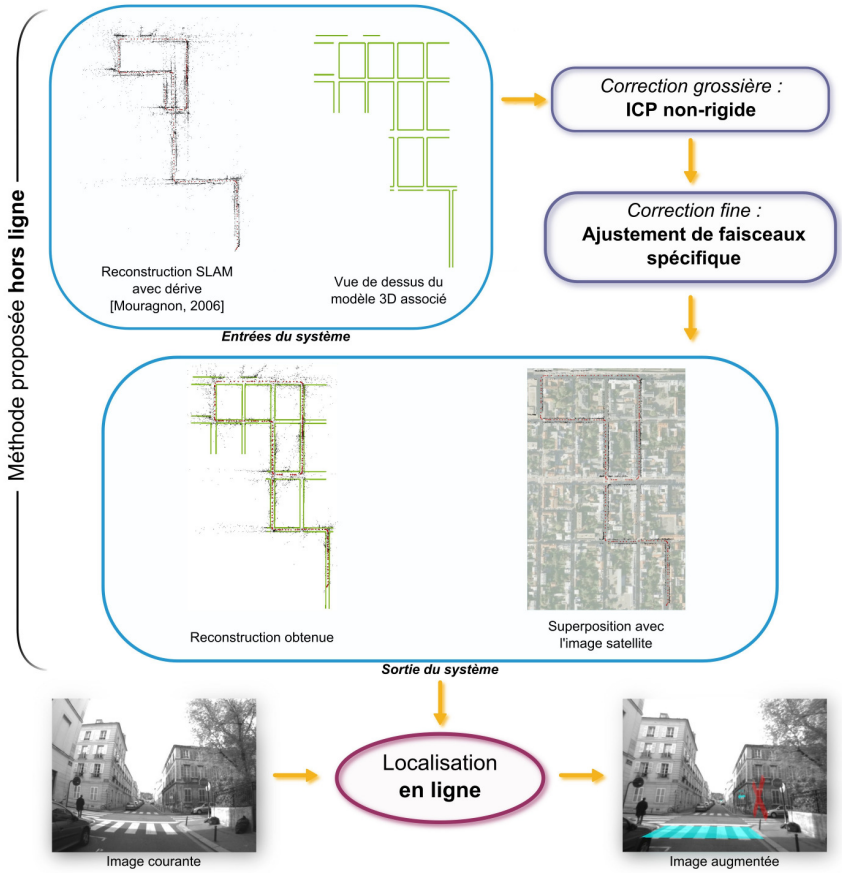


FIGURE 2.13 – **Résumé de la méthode.** La méthode proposée consiste à construire une base d'amers visuels à partir de laquelle il sera possible de se relocaliser.

Relocalisation d'une caméra mobile

Le deuxième processus de cette partie a pour objectif de relocaliser une caméra mobile dans l'environnement précédemment appris. En effet, une fois la base d'amers visuels créée (hors ligne), il est possible de relocaliser une caméra se déplaçant au sein de cette base de données (en ligne). Ceci est possible grâce à l'appariement des points d'intérêt de l'image courante avec ceux constituant la base.

Nous montrerons également dans ce chapitre que la pose de la caméra obtenue est suffisamment précise pour pouvoir être utilisée dans des applications de réalité augmentée, par exemple pour des scénarii d'aide à la navigation.

CHAPITRE 3

ICP non-rigide

Dans ce chapitre, nous présentons un algorithme permettant d'aligner grossièrement une reconstruction SLAM contenant des dérives (obtenue dans notre cas avec l'algorithme de Mouragnon et al. (2006)) avec un modèle 3D de ville. Même si ces modèles 3D sont peu précis (voir leur description à la section 2.6.2.3), ils sont considérés sans dérive, c'est à dire globalement cohérents. Cette information supplémentaire permettra donc de corriger la dérive du SLAM, qui se traduit généralement visuellement par l'écrasement de la reconstruction. Après avoir introduit les approches classiques permettant d'aligner différents ensembles de données (section 3.1), nous présenterons le modèle de transformations utilisé pour aligner la reconstruction et le modèle 3D (section 3.2). Nous décrirons alors le processus d'alignement (section 3.3).

Les travaux décrits dans ce chapitre ont donné lieu à deux publications (Lothe et al. (2009a,b)).

3.1 Méthodes d'alignement 3D

Aligner deux ensembles de données 3D est un domaine de recherche très actif, en particulier en modélisation 3D. La méthode fréquemment utilisée aujourd'hui est l'ICP (*Iterative Closest Point*). Un état de l'art sur l'ICP et ses variantes peut être trouvé dans l'article de Rusinkiewicz and Levoy (2001). L'idée directrice de cette approche est de décomposer le problème d'alignement en deux sous-problèmes :

- ▷ **Association des données.** Elle consiste à créer des paires entre les données des deux ensembles. Chacun des éléments du premier ensemble est associé à l'élément qui lui correspond dans le deuxième ensemble. Cette correspondance étant généralement inconnue, on associe chaque élément à l'élément qui lui est le plus proche dans l'autre ensemble. Il est donc nécessaire de définir une métrique permettant de mesurer la distance entre les éléments des deux ensembles. La notion souvent utilisée est la distance euclidienne. Néanmoins, des métriques plus complexes (probabilité, notion d'apparence, de forme, *etc.*) peuvent être mises en place de façon à améliorer la mise en correspondance des éléments et ainsi limiter les faux appariements (Rusinkiewicz and Levoy (2001)).

▷ **Minimisation de l'erreur.** Une fois les associations réalisées, la deuxième étape consiste à minimiser la distance entre les éléments associés. Il est donc nécessaire de définir à la fois une métrique (qui n'est pas nécessairement la même que celle utilisée pour l'association des données) et l'espace des transformations 3D dans lequel peuvent être modifiés les ensembles de données de façon à minimiser cette métrique. Généralement, les transformations utilisées sont les transformations euclidiennes (à savoir une rotation et une translation).

Ces deux étapes sont itérées jusqu'à convergence. Ce processus itératif permet de remettre en cause les associations de données après chaque minimisation. On espère ainsi converger vers la solution optimale. Notons cependant que rien n'assure la convergence de l'algorithme ICP.

Dans les cas particuliers où une carte de distance est calculable à l'avance, Fitzgibbon (2001) a montré que ces deux étapes peuvent être résolues simultanément. Néanmoins, cette approche semble peu appropriée dans notre cas, en particulier puisque l'environnement considéré possède 3 dimensions et est vaste, ce qui complexifie le calcul d'une telle carte de distance.

Dans la suite, nous allons définir les différents éléments qui caractérisent l'approche ICP, à savoir l'espace des transformations 3D considéré, l'association des données et la méthode de minimisation de la fonction d'erreur utilisée.

3.2 Espace de transformations utilisé

Nous avons vu précédemment que le cas le plus courant est de rechercher la transformation euclidienne (voire une similitude) entre les deux ensembles de données (par exemple dans les travaux de Kaminsky et al. (2009)). Cependant, dans notre cas de figure, les déformations induites par les dérives du SLAM sont très complexes et le problème ne peut donc pas se limiter à la recherche d'une rotation et d'une translation. Afin de pallier les limites des transformations euclidiennes, des méthodes d'alignement non-rigide ont été proposées. Le terme non-rigide est employé ici au sens large, c'est à dire dès lors que la transformation recherchée a plus de degrés de liberté qu'une rotation, une translation et un facteur d'échelle.

En raison de leur grand nombre de degrés de liberté, les transformations non-rigides nécessitent généralement d'être contraintes à une sous-classe de transformations afin d'assurer la bonne convergence de l'algorithme. Par exemple, Castellani et al. (2007) utilisent un terme de régularisation afin de limiter les déformations utilisées à celles physiquement possibles pour une feuille de papier. De la même manière, nous allons proposer un modèle simplifié de la dérive que le SLAM présente dans notre contexte. Ce modèle amènera à une classe de transformations non-rigides qui modélisent de manière relativement réaliste les déformations produites par cette dérive, tout en conservant un nombre de degrés de liberté suffisamment réduit pour assurer la convergence de l'algorithme d'optimisation.

3.2.1 Modélisation de la dérive du SLAM

La dérive du SLAM est un phénomène complexe et difficile à formaliser. La modélisation que nous en faisons dans cette partie est une forte approximation. Néanmoins, nous verrons dans la partie expérimentale (chapitre 5) que celle-ci est suffisamment bonne pour permettre un alignement correct entre la reconstruction 3D et le modèle de l'environnement.

Pour définir l'espace de transformations considéré, nous nous sommes appuyés sur les résultats expérimentaux obtenus avec la méthode de Mouragnon et al. (2006). Nous avons observé

que le facteur d'échelle, lorsque que la caméra regarde vers l'avant du véhicule, est quasi-constant sur les lignes droites alors qu'il a tendance à être fortement modifié dans les virages (voir par exemple la figure 2.9, page 39). Nous avons donc décidé d'utiliser une transformation affine par morceaux : les lignes droites de la trajectoire sont considérées comme des éléments extensibles et des articulations sont placées à chaque virage. Ainsi, les transformations retenues sont des *similitudes par morceaux avec contraintes de jointure aux extrémités*. Il est intéressant de noter qu'on retrouve ces transformations dans les travaux de Levin and Szeliski (2004) pour mettre en correspondance une trajectoire reconstruite par le processus SLAM avec une carte grossière de la trajectoire parcourue.

3.2.2 Fragmentation de la reconstruction

Les transformations obtenues étant des transformations par segments, il est nécessaire avant toute chose de *fragmenter* la reconstruction, c'est à dire de segmenter la trajectoire reconstruite de la caméra (section 3.2.2.1) et d'associer à chacun des segments de caméras obtenus les points 3D qui lui sont liés (section 3.2.2.2).

3.2.2.1 Approximation polygonale de la trajectoire

Afin de segmenter la trajectoire reconstruite, nous avons repris la méthode proposée par Lowe (1987) pour la segmentation de contours. Cette étape est également appelée *approximation polygonale*.

La figure 3.1 résume la méthode utilisée. Considérons la trajectoire reconstruite comme un ensemble de caméras temporellement ordonnées $\{\mathcal{C}_1 \ \dots \ \mathcal{C}_M\}$. Tout d'abord, nous considérons l'unique segment $\mathbf{l} = \mathcal{C}_1\mathcal{C}_M$. Nous cherchons alors la caméra la plus éloignée de ce segment :

$$\mathcal{C}_{i_m} = \underset{\mathcal{C}_i \in \{\mathcal{C}_1 \ \dots \ \mathcal{C}_M\}}{argmax} \ d(\mathbf{l}, \mathcal{C}_i) \tag{3.1}$$

où $d(\mathbf{l}, \mathcal{C}_i)$ est la distance orthogonale entre la droite $\mathbf{l}$ et la caméra $\mathcal{C}_i$. On note $d_{max} = d(\mathbf{l}, \mathcal{C}_{i_m})$ cette distance maximale.

Afin de mesurer la qualité d'approximation de l'ensemble $\{\mathcal{C}_1 \ \dots \ \mathcal{C}_M\}$ par la droite $\mathbf{l}$, on définit le critère Ψ :

$$\Psi(\{\mathcal{C}_1 \ \dots \ \mathcal{C}_M\}) = \frac{||\mathbf{l}||}{d_{max}} \tag{3.2}$$

Cette valeur est directement liée à la qualité d'approximation du segment. Ψ est élevé lorsque la longueur du segment est grande devant d_{max}. Autrement dit, plus Ψ est grand, plus l'approximation de $\{\mathcal{C}_1 \ \dots \ \mathcal{C}_M\}$ par $\mathbf{l}$ est bonne. Nous pouvons faire la même opération sur les sous-segments $\{\mathcal{C}_1 \ \dots \ \mathcal{C}_{i_m}\}$ et $\{\mathcal{C}_{i_m} \ \dots \ \mathcal{C}_M\}$:

$$\begin{cases} \Psi_r = \Psi(\{\mathcal{C}_1 \ \dots \ \mathcal{C}_M\}) \\ \Psi_g = \Psi(\{\mathcal{C}_1 \ \dots \ \mathcal{C}_{i_m}\}) \\ \Psi_d = \Psi(\{\mathcal{C}_{i_m} \ \dots \ \mathcal{C}_M\}) \end{cases} \tag{3.3}$$

On regarde alors si un des sous-segments est de meilleur qualité que le segment principal, c'est à dire si :

$$max(\Psi_g, \Psi_d) > \sigma \times \Psi_r \tag{3.4}$$

Si cette équation est vérifiée, la caméra $\mathcal{C}_{i_m}$ est conservée comme extrémité d'un segment de trajectoire et le processus est lancé récursivement sur les deux sous-segments. Dans le cas contraire, le segment $\mathbf{l}$ est conservé comme étant le segment minimal. Le facteur σ est un facteur à ajuster en fonction de la qualité de l'approximation polygonale recherchée. Des exemples de segmentation de trajectoire peuvent être trouvés dans la partie expérimentale (chapitre 5).

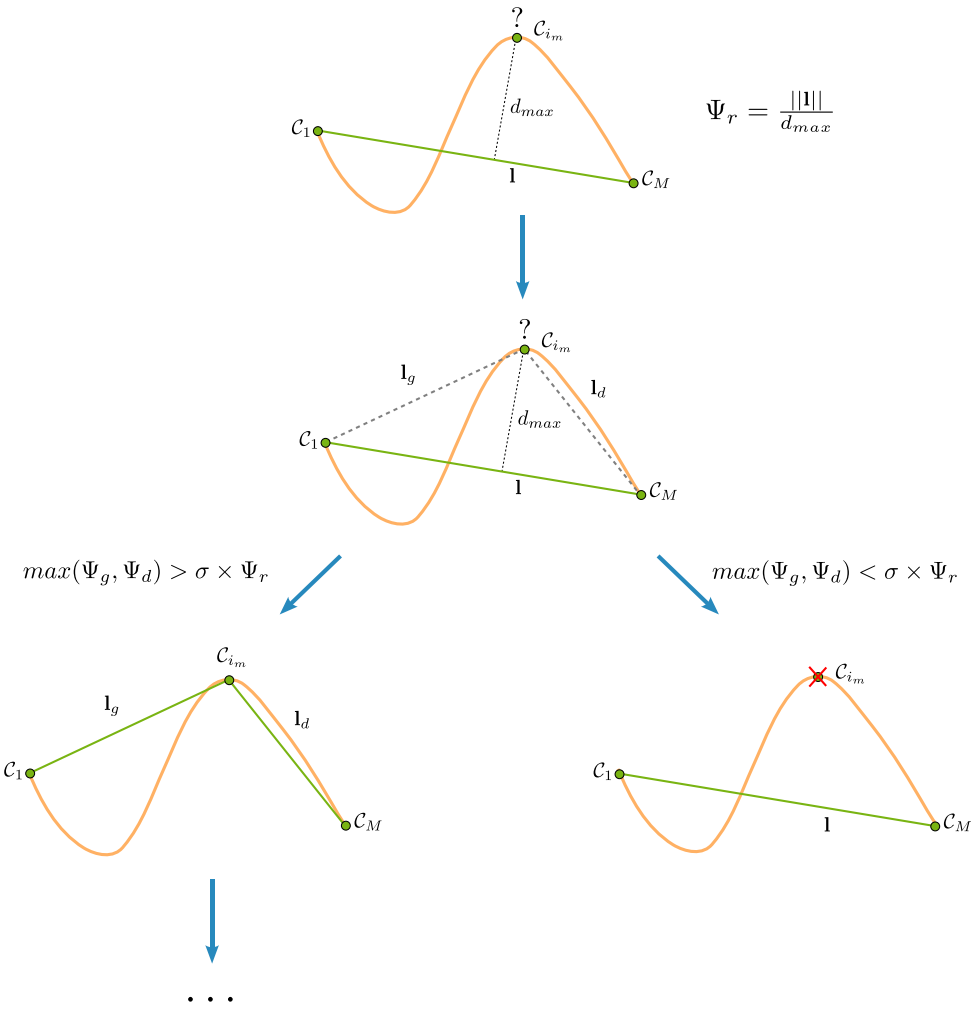


FIGURE 3.1 – **Approximation polygonale de la trajectoire du véhicule.** Cette méthode permet de découper en segments la trajectoire reconstruite.

3.2.2.2 Formation des fragments de la reconstruction

A la sortie de la segmentation de la trajectoire, nous disposons donc de m segments $(\mathcal{T}_i)_{1\leq i\leq m}$ dont les extrémités sont deux caméras notées $\mathbf{e}_i$ et $\mathbf{e}_{i+1}$. Une fois la trajectoire découpée, chacun des points 3D reconstruits doit être associé à un segment de caméras. Le point 3D subira alors la même transformation 3D que le segment de caméras auquel il est rattaché.

Afin de définir à quel segment appartient un point 3D, on définit la notion de visibilité. Nous dirons qu'un segment voit un point 3D si au moins une caméra de ce segment observe ce point. Deux cas de figure sont alors possibles. Le cas le plus simple apparaît quand seulement un segment voit le point : il est alors lié à ce segment. Dans l'autre cas, si le point est

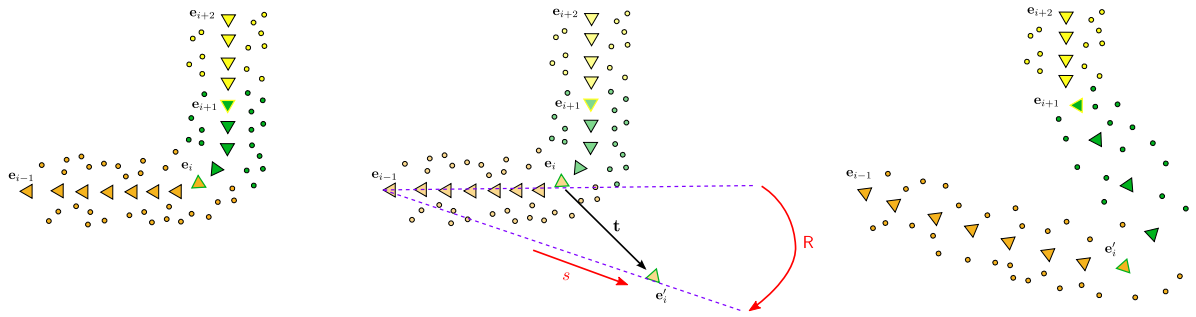
observé par plusieurs segments (dans les virages), plusieurs politiques sont possibles : associer ce point 3D au segment qui le voit en premier, en dernier, qui le voit le plus, *etc.* Nous avons testé expérimentalement ces différentes politiques et il s'avère qu'elles fournissent des résultats équivalents. Nous avons donc choisi arbitrairement d'associer le point au segment qui l'observe en dernier.

Dans la suite, nous appellerons $\mathcal{B}_i$ un *fragment* composé des caméras du segment $\mathcal{T}_i$ (c'est à dire incluses entre les extrémités $\mathbf{e}_i$ et $\mathbf{e}_{i+1}$) et des points 3D associés. Il est important de noter que pour $2 \leq i \leq m-1$, le fragment $\mathcal{B}_i$ partage ses extrémités avec ses fragments voisins $\mathcal{B}_{i-1}$ et $\mathcal{B}_{i+1}$.

3.2.3 Paramétrisation des transformations retenues

Nous avons vu précédemment que les transformations utilisées dans l'ICP sont des similitudes par morceaux avec contraintes de jointure aux extrémités. En pratique, ces transformations sont paramétrées par le déplacement des extrémités $(\mathbf{e}_i)_{1\leq i\leq m+1}$ de chacun des segments, c'est à dire leur translation 3D dans l'espace.

Prenons l'exemple d'un déplacement de l'extrémité $\mathbf{e}_i$ (figure 3.2). Dans cet exemple, l'extrémité $\mathbf{e}_i$ subit une translation $\mathbf{t}$. Pour le fragment $\mathcal{B}_{i-1}$, cette translation peut également être vue comme une similitude centrée en $\mathbf{e}_{i-1}$, de rotation $\mathcal{R}$ et de facteur s (figure 3.2(b)). C'est cette similitude, notée $\mathcal{S}(\mathbf{e}_{i-1}, s, \mathcal{R})$, qui est alors appliquée à l'ensemble des caméras et points 3D du fragment $\mathcal{B}_{i-1}$ (figure 3.2(c)). Le même raisonnement est réalisé sur le fragment $\mathcal{B}_i$ où la translation de $\mathbf{e}_i$ est vue comme la similitude $\mathcal{S}'(\mathbf{e}_{i+1}, s', \mathcal{R}')$.



(a) (b) (c)

FIGURE 3.2 – **Exemple d'une transformation par fragments sur une reconstruction SLAM.** (a) La segmentation de la reconstruction originale. (b) L'extrémité $\mathbf{e}_i$ subit une translation. La similitude $\mathcal{S}(\mathbf{e}_{i-1}, s, \mathcal{R})$ est déduite de ce déplacement et est appliquée au fragment $\mathcal{B}_{i-1}$. Un traitement équivalent est réalisé sur le fragment $\mathcal{B}_i$. (c) montre le résultat de cette transformation : les deux fragments liés à l'extrémité déplacée ont été modifiés.

Notons que cette paramétrisation des transformations ne permet pas de modifier l'angle de roulis des caméras, c'est à dire la rotation autour de l'axe optique. Le fait de ne pas optimiser l'angle de roulis est un choix que nous avons fait dans ce chapitre. En effet, nous avons remarqué que dans le cadre d'une caméra embarquée sur un véhicule, l'estimation de l'angle de roulis réalisée par le processus du SLAM est relativement correcte. Nos expériences nous ont également montré qu'optimiser l'angle de roulis pendant l'ICP non-rigide n'améliore pas les résultats obtenus. Au contraire, cela produit généralement une dégradation des résultats lorsque

beaucoup de points 3D reconstruits ne se situent pas sur les façades (voitures, arbres, *etc.*). Ce phénomène a également été récemment observé par Strecha et al. (2010).

Il est cependant nécessaire de fixer au départ l'angle de roulis global de la reconstruction par rapport au modèle 3D. Sourimant et al. (2007) ont montré qu'à partir d'un modèle 3D simple de l'environnement, il est possible d'obtenir facilement un modèle d'élévation du terrain en utilisant la triangulation de Delaunay (Delaunay (1934)). De plus, de tels modèles d'élévation sont désormais librement accessibles (*e.g.* sur le site de GeoNames [1]). La hauteur de la caméra sur le véhicule étant connue, il est alors possible d'ajuster le roulis de la reconstruction afin de minimiser l'erreur d'élévation des caméras. Pour chaque segment de trajectoire, l'erreur d'élévation résiduelle est alors corrigée en modifiant l'angle de roulis du segment précédent. Ceci nous fournit une approximation correcte de l'angle de roulis de chacun des segments par rapport au repère monde. Notons de plus que cet angle sera remis en cause dans la deuxième étape de correction (chapitre 4).

En pratique, l'altitude des caméras n'est pas optimisée. En effet, seuls les points reconstruits sur le sol permettent de fixer ce paramètre. Néanmoins, il apparaît qu'en proportion, très peu de points du sol sont reconstruits. Néanmoins, la caméra étant rigidement liée au véhicule, sa distance au sol est connue. Ainsi, l'altitude de chacune des caméras est donc fixée à l'aide d'une carte d'élévation du terrain (dans nos expériences, l'élévation est supposée constante sur l'ensemble du parcours).

3.3 Recherche de l'alignement optimal

Notre modèle de déformation étant défini, il est possible d'aligner le nuage de points 3D reconstruit avec le modèle 3D de l'environnement en utilisant un algorithme de type ICP. Pour cela, nous allons préciser ci-après les 2 étapes principales caractérisant ces algorithmes, à savoir l'association des données et la minimisation de l'erreur associée. Pour cela, nous considérons dans cette section que nous possédons en entrée de l'ICP le modèle 3D de l'environnement et la reconstruction SLAM fragmentée comme expliquée précédemment.

3.3.1 Métrique utilisée et association des données

Le but de nos travaux étant d'aligner le nuage de points reconstruit avec le modèle 3D, il nous faut définir une métrique permettant d'évaluer la qualité de cet alignement. La métrique ω choisie dans notre étude est la distance orthogonale d entre un point 3D et le plan du modèle 3D $\mathcal{M}$ auquel il appartient. Comme nous l'avons vu dans la description des méthodes ICP (section 3.1), les correspondances réelles entre les points 3D et les plans du modèle étant inconnues, la métrique est définie comme étant la distance entre le point 3D et le plan $\mathbf{\Pi}$ le plus proche :

$$\begin{aligned} \omega(\mathcal{Q}^i, \mathcal{M}) &= d(\mathcal{Q}^i, \mathcal{M}) \\ &= \min_{\mathbf{\Pi}_j \in \mathcal{M}} d(\mathcal{Q}^i, \mathbf{\Pi}_j) \end{aligned} \tag{3.5}$$

Rappelons que pendant chaque étape de minimisation de l'ICP, l'association des données est constante. Ainsi, nous noterons dans la suite $\mathbf{\Pi}_{h_i}$ le plan associé au point $\mathcal{Q}^i$. Il est important

1. www.geonames.org

de noter que la distance d prend en compte le fait que les plans 3D sont des plans finis : pour être associé au plan $\mathbf{\Pi}$, un point 3D $\mathcal{Q}$ doit avoir sa projection orthogonale à l'intérieur des limites de $\mathbf{\Pi}$. Les points qui ne sont associés à aucun plan sont alors simplement retirés de l'optimisation.

3.3.2 Minimisation robuste de la métrique

Dans notre étude, la métrique utilisée pour l'association des données et pour la minimisation de l'erreur est la même. Ainsi, dès lors que la fonction ω est définie, on peut chercher la similitude par morceaux qui minimise cette métrique pour l'ensemble des données associées. Nous avons vu à la section 3.2.3 que ces transformations sont paramétrées par la position des extrémités des fragments $(\mathbf{e}_1 \dots \mathbf{e}_m)$. Le problème revient donc à chercher la position de ces extrémités qui minimise la fonction de coût ϵ :

$$(\mathbf{e}_1 \dots \mathbf{e}_m) = \underset{\mathbf{e}_1 \dots \mathbf{e}_m}{argmin}\, \epsilon \tag{3.6}$$

avec :

$$\begin{aligned} \epsilon &= \sum_i \omega^2 \\ &= \sum_i d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})^2 \end{aligned} \tag{3.7}$$

Robustesse aux points aberrants. L'association $(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})$ étant réalisée au plus proche, elle peut être erronée. En effet, il est possible que le point $\mathcal{Q}^i$ ne soit pas dans la réalité sur le plan $\mathbf{\Pi}_{h_i}$. Ces mauvaises associations peuvent être liées à deux raisons principales : une initialisation trop éloignée de $\mathcal{Q}^i$ par rapport à sa position réelle ou le fait que ce point n'est pas positionné sur le modèle dans la réalité (*i.e.* n'appartient pas à une façade). Dans les deux cas, le terme $d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})$ peut être alors prépondérant et donc empêcher l'obtention du minimum recherché. Pour limiter cet effet, nous utilisons un M-estimateur robuste ρ dans l'équation (3.7) :

$$\epsilon = \sum_i \rho(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})) \tag{3.8}$$

Le M-estimateur utilisé est le M-estimateur de Tukey (Huber (1981)). Le seuil du M-estimateur peut être réglé automatiquement grâce au MAD lorsque la distribution des erreurs peut être assimilée à une gaussienne. Or, il s'avère que cette hypothèse n'est vérifiée que sur chacun des fragments (figure 3.3(b)) mais pas sur la reconstruction dans sa globalité (figure 3.3(a)). Le seuil ξ utilisé par le M-estimateur ne peut donc pas être global mais il doit être propre à chacun des fragments de la reconstruction.

Pondération des fragments. En l'état actuel, on notera alors que l'influence de chacun des fragments dans le processus d'optimisation n'est pas la même. En effet, l'influence d'un fragment $\mathcal{B}$ dépend :

- ▷ du seuil de Tukey ξ appliqué aux résidus de celui-ci
- ▷ du nombre de points 3D qu'il contient (noté $card_{\mathcal{Q}}(\mathcal{B})$)

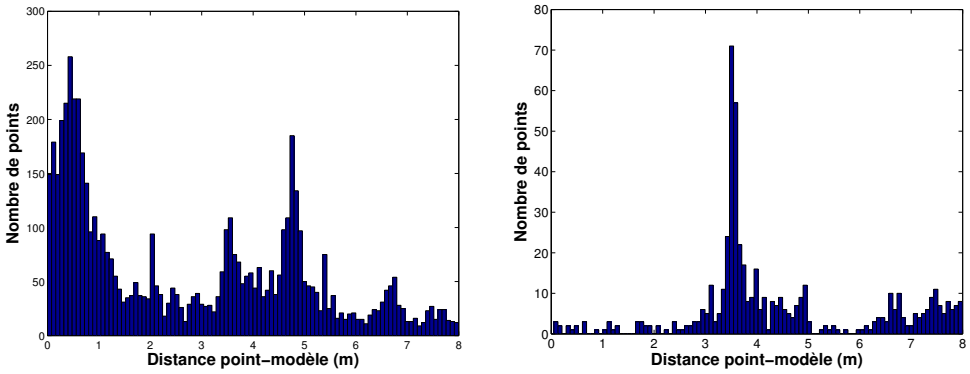
Pour assurer une meilleure convergence de notre algorithme, nous proposons donc de normaliser l'influence de chaque fragment vis à vis de ces deux paramètres. Tout d'abord, afin que

l'influence d'un fragment ne dépende plus du seuil du Tukey qui lui est associé, nous proposons de normaliser les résidus de chacun des fragments de la reconstruction. La fonction ϵ devient alors :

$$\epsilon = \sum_i \rho'_{l_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})) \tag{3.9}$$

où l_i est l'indice du fragment contenant le point $\mathcal{Q}^i$, et ρ'_{l_i} le M-estimateur de Tukey normalisé associé au seuil ξ_{l_i}:

$$\rho'_{l_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})) = \frac{\rho_{l_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i}))}{\max\limits_{\mathcal{Q}^j \in \mathcal{B}_{l_i}} \rho_{l_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i}))} \tag{3.10}$$



(a) (b)

FIGURE 3.3 – **Histogramme de la distance point-plan avant l'étape d'ICP non-rigide.** (a) est l'histogramme sur l'ensemble de la reconstruction et (b) est l'histogramme sur un seul fragment.

Pour éviter le second problème, c'est à dire que les fragments possédant peu de points 3D ne soient pas optimisés en faveur des fragments plus conséquents, nous avons décidé de normaliser les résidus des points 3D de chacun des fragments en fonction de leur cardinal. Le M-estimateur final $\rho^*_{l_i}$ s'écrit alors :

$$\rho^*_{l_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})) = \frac{\rho'_{l_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i}))}{card_{\mathcal{Q}}(\mathcal{B}_{l_i})} \tag{3.11}$$

et la fonction ϵ effectivement optimisée s'écrit :

$$\epsilon = \sum_i \rho^*_{l_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})) \tag{3.12}$$

Dans notre cas, c'est alors l'algorithme de Levenberg-Marquardt (Levenberg (1944)) qui est utilisé afin de minimiser cette erreur.

3.3.3 Initialisation de l'algorithme

L'association des données se faisant au sens de l'élément le plus proche, il est nécessaire que l'ICP non-rigide soit initialisé le plus proche possible de la solution. Dans notre cadre d'étude, cela revient à placer les extrémités des segments proches de l'endroit où elles se trouvent en

réalité. Cette initialisation peut par exemple être réalisée avec les données GPS si de telles données sont associées aux images. En effet, Kaminsky et al. (2009) ont montré que l'utilisation de ce type d'informations donne une initialisation qui est suffisamment correcte pour les méthodes d'alignement de modèles. N'ayant pas cette information à notre disposition dans notre cas, nous utiliserons dans nos expériences une interface graphique afin de simuler les données GPS.

3.4 Discussion

La figure 3.4 donne un exemple de reconstruction SLAM obtenue après l'ICP non-rigide sur une séquence de synthèse (cette séquence sera étudiée en détail dans le chapitre 5). Nous pouvons voir sur cette figure que la géométrie globale de la reconstruction est cohérente avec le modèle 3D. Néanmoins, deux limites principales peuvent d'ores et déjà être mises en avant.

Tout d'abord, la correction appliquée dans les virages n'est pas satisfaisante. En effet, la fragmentation de la reconstruction crée au niveau des virages une discontinuité sur le facteur d'échelle estimé. De plus, un point 3D n'étant associé qu'au fragment qui l'observe en dernier, la cohérence entre les caméras et les points 3D qu'elles observent n'est plus assurée dans les virages.

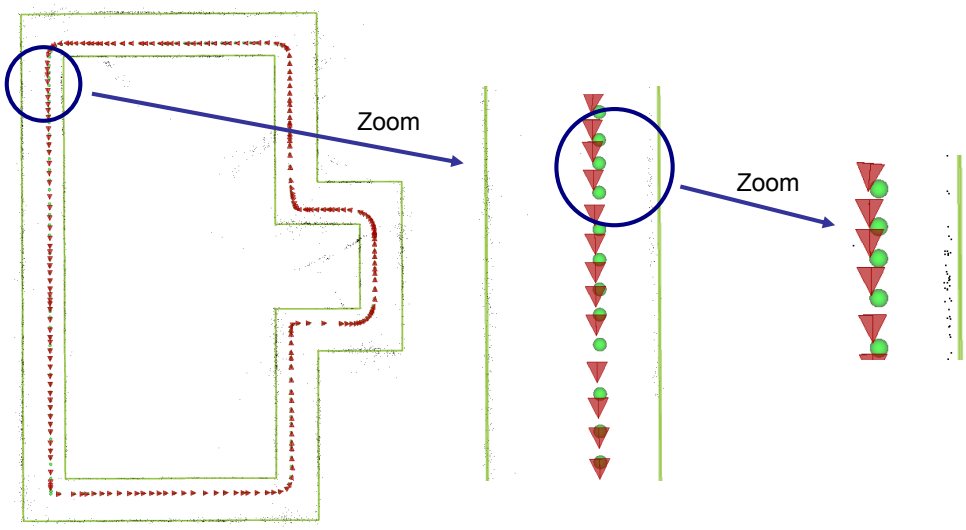


FIGURE 3.4 – **Résultat de l'alignement par ICP non-rigide.** Les pyramides rouges sont les caméras reconstruites et les sphères vertes représentent la vérité terrain. On peut observer que la position des caméras semble correcte, excepté dans le sens de la trajectoire, et que le nuage de points reconstruits n'est pas parfaitement aligné avec le modèle 3D.

De plus, au sein même des fragments, les erreurs de positionnement des caméras peuvent être encore localement importantes. En effet, la modélisation de la dérive du SLAM que nous avons choisie (section 3.2.1) n'est qu'une approximation grossière de la réalité puisque, même si elle est généralement plus faible, cette dérive apparaît également dans les lignes droites. Les transformations utilisées dans l'ICP non-rigide sont donc trop contraintes pour pouvoir corriger précisément la reconstruction SLAM. Ceci est confirmé par la figure 3.4 qui illustre les résultats obtenus sur la séquence Synthèse 1 (voir section B.1). Sur cette figure, on peut voir que l'erreur résiduelle sur le positionnement des caméras est encore importante dans les lignes droites, en

particulier dans la direction de la trajectoire de la caméra. Des résultats supplémentaires sur l'étape d'ICP non-rigide pourront être trouvés dans le chapitre 5.

Par ailleurs, dans des travaux récents, Pylvänäinen et al. (2010) ont repris l'idée d'ICP non-rigide présenté dans ce chapitre dans le but de corriger et de géoréférencer un nuage de points obtenu grâce à un LIDAR. Cependant, dans leur cas, la totalité de la reconstruction n'est pas optimisée simultanément. Pour chaque carrefour, ils choisissent de prendre en compte les points (et les positions de LIDAR qui s'y rapportent) placés à une certaine distance de cette intersection. Pour chaque carrefour, ils recherchent alors la similitude qui permet d'aligner au mieux la sous-reconstruction SLAM retenue avec le modèle 3D de ville. En conséquence, la position de chaque entité (point et dispositif LIDAR) peut être optimisée indépendamment pour différentes intersections. Au final, ces éléments ont donc plusieurs positions éventuelles. La position retenue est alors une interpolation entre toutes ces positions. Cela permet de limiter le problème de discontinuité que nous rencontrons à l'issue de l'ICP non-rigide et de lisser la correction appliquée à la reconstruction SLAM.

Dans le chapitre suivant, nous allons présenter le second processus de notre méthode. C'est ce second processus qui permet, dans notre approche, de remettre en cause les limites de l'ICP non-rigide et ainsi corriger les erreurs résiduelles observées à sa sortie.

CHAPITRE 4

Ajustements de faisceaux contraints par un SIG

Dans cette section, nous proposons une méthode permettant de corriger les erreurs locales résiduelles de la reconstruction SLAM à la sortie de l'ICP non-rigide. Ainsi, après avoir mis en évidence les limites des méthodes utilisées habituellement pour raffiner les reconstructions SLAM, nous présenterons une nouvelle approche permettant de prendre en compte à la fois les informations images et les contraintes apportées par le modèle 3D.

Les travaux décrits dans ce chapitre ont donné lieu à plusieurs publications (Lothe et al. (2009c,d, 2010a)).

4.1 Méthodes classiques et limites

Pour atteindre des corrections plus fines, il est nécessaire de relâcher les contraintes imposées dans l'ICP non-rigide. La méthode classique utilisée en vision par ordinateur pour optimiser la structure d'une reconstruction (c'est à dire la pose des caméras et la position des points 3D) est l'ajustement de faisceaux (Triggs et al. (2000)).

Dans cette section, nous allons tout d'abord montrer qu'un ajustement de faisceaux classique, c'est à dire n'incluant pas de contrainte relative au modèle 3D de la scène, ne permet pas de garantir une amélioration du résultat obtenu à l'issue de l'étape d'ICP. Nous présenterons ensuite une méthode classique permettant d'introduire le modèle 3D de la scène comme une contrainte supplémentaire dans l'ajustement de faisceaux.

4.1.1 Ajustement de faisceaux classique

L'ajustement de faisceaux classique optimise uniquement les relations géométriques qui existent entre les caméras et les points 3D. Ainsi, la fonction de coût de cet ajustement de faisceaux est l'erreur de reprojection des points 3D $(\mathcal{Q}^i)_i$ dans les caméras $(\mathcal{C}_j)_j$ (voir la section 2.3.2.5 pour plus de détails). Les paramètres optimisés sont la position des points 3D et les paramètres externes des caméras $(\mathsf{C}_j^E)_j$, le calibrage étant considéré connu :

$$\mathcal{F}(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E, \mathcal{Q}^1, \ldots, \mathcal{Q}^M) = \sum_{1 \leq j \leq N} \sum_{i \in \mathcal{A}_j} ||\mathbf{q}_j^i - \pi(\tilde{\mathsf{P}}_j \tilde{\mathcal{Q}}^i)||^2 \tag{4.1}$$

où $\tilde{\mathsf{P}}_j$ est la matrice de projection de la j^{eme} caméra.

Par exemple, c'est cet ajustement de faisceaux qui est utilisé par Levin and Szeliski (2004) afin de raffiner la géométrie de leur reconstruction. Leur idée est que la correction grossière qu'ils appliquent au préalable à la reconstruction grâce à la carte de la trajectoire permet d'améliorer l'initialisation de l'ajustement de faisceaux. Ils considèrent alors que celui-ci a une grande probabilité de converger vers la solution optimale.

Dans notre contexte, appliquer cet ajustement de faisceaux à l'issue de l'ICP va permettre de retrouver une cohérence entre les points 3D et les caméras qui les observent, en particulier dans les virages. Cependant, au cours de ce processus d'optimisation, aucune contrainte n'est introduite pour assurer la cohérence globale de la reconstruction avec le modèle 3D de la scène, et ainsi éviter une dérive du facteur d'échelle. Cette perte de cohérence globale est confirmée expérimentalement. En effet, si l'erreur de reprojection a bien diminué suite à l'ajustement de faisceaux, on observe une perte de cohérence aussi bien entre le nuage de points reconstruit et le modèle 3D qu'entre la trajectoire estimée de la caméra et la trajectoire réelle. Des exemples de perte de cohérence sont illustrés dans la figure 4.1 et le tableau 4.1.

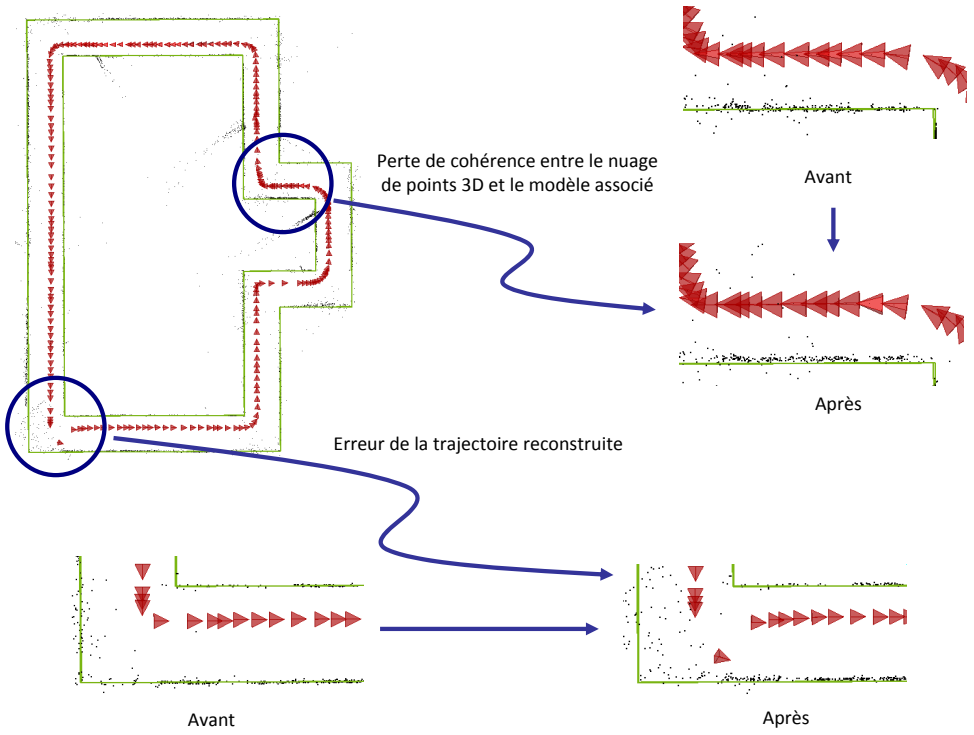


FIGURE 4.1 – **Comportement de l'ajustement de faisceaux classique sur la séquence Synthèse 1.** Appliquer une optimisation géométrique classique après l'ICP non-rigide peut amener à une mauvaise déformation de la reconstruction.

4.1.2 Combinaison linéaire de fonctions de coût

Comme l'a montré le paragraphe précédent, les seules contraintes de la géométrie multi-vue ne sont pas suffisantes pour assurer la convergence de l'ajustement de faisceaux vers la solution optimale (au sens géométrique). Nous proposons donc d'introduire des contraintes supplémentaires relatives au modèle 3D de la scène. Il est néanmoins intéressant de noter que

	Après ICP non-rigide	Après ICP non-rigide + ajustement de faisceaux classique
Distance moyenne entre les caméras et la vérité terrain (m)	**0,51**	0,57
Ecart-type (m)	**0,59**	0,46
Distance point-plan moyenne (m)	**0,11**	0,15
Ecart-type (m)	**0,08**	0,10
Seuil du Tukey	0,38	×

TABLE 4.1 – **Résultats numériques obtenus avec l'ajustement de faisceaux classique sur la séquence Synthèse 1.** On peut voir que l'ajustement de faisceaux classique a tendance à dégrader les résultats de l'ICP non-rigide.

certaines approches (*e.g.* Bartoli and Sturm (2003)), bien que n'utilisant pas de modèle 3D tel que nous le proposons, s'appuient sur l'hypothèse de la planéité par morceaux de la scène de sorte à améliorer la reconstruction 3D.

Dans notre cas, tout comme lors de l'étape d'ICP, nous proposons d'ajouter un terme favorisant l'appartenance des points 3D du SLAM au modèle 3D de la scène. Une méthode classiquement utilisée lorsque deux critères sont à minimiser simultanément est de minimiser leur somme (Horn and Schunck (1981); Chui and Rangarajan (2003); Modersitzki (2004); Pilet et al. (2005); Michot et al. (2010)):

$$\mathcal{F}_{CL}(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E, \mathcal{Q}^1, \ldots, \mathcal{Q}^M) = \mathcal{F}(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E, \mathcal{Q}^1, \ldots, \mathcal{Q}^M) + \alpha \times \mathcal{G}(\mathcal{Q}^1, \ldots, \mathcal{Q}^M) \quad (4.2)$$

où $\mathcal{F}$ est la fonction de reprojection classique et $\mathcal{G}$ une métrique entre les points reconstruits et le modèle 3D. Une telle fonction a été développée et testée dans le cadre de nos travaux. Sa description peut être trouvée dans l'annexe A.

Le facteur α est un facteur permettant de pondérer les deux critères. En effet, en plus de n'avoir pas forcément la même unité, les deux fonctions n'ont pas nécessairement le même ordre de grandeur. Il est donc nécessaire de pondérer ces fonctions afin d'éviter que le processus de minimisation privilégie uniquement un des deux critères.

La principale difficulté lorsqu'on utilise des combinaisons linéaires de fonctions de coût est de trouver le facteur α qui donne la solution optimale. De plus, la meilleure valeur du facteur α est généralement fortement dépendante de la séquence traitée (Bartoli et al. (2008)). Cette méthode a été évaluée dans notre contexte pour plusieurs valeurs de α. Les résultats obtenus sont consignés dans le tableau 4.2. Cette expérience souligne la sensibilité de cette méthode vis à vis du choix du paramètre α. En effet, les résultats obtenus ont montré qu'une faible variation de α peut avoir un impact notable sur la reconstruction SLAM obtenue. On notera même que certaines valeurs de α peuvent entraîner une dégradation de cette reconstruction. Dans la section suivante, nous allons donc présenter une solution permettant de s'affranchir du paramètre α.

4.2 Approche proposée : ST-CBA

L'approche que nous proposons est de définir une nouvelle fonction de coût prenant en compte, dans un unique terme, à la fois l'information image et l'information 3D liée au mo-

	Après ICP non-rigide	$\alpha = 0,01$	$\alpha = 0,1$	$\alpha = 0,5$	$\alpha = 1$	$\alpha = 1,5$	$\alpha = 2$
Distance moyenne entre les caméras et la vérité terrain (m)	0,51	0,52	0,48	**0,32**	0,35	0,42	0,53
Ecart-type (m)	0,59	0,89	0,67	**0,23**	**0,23**	0,35	0,77
Distance point-plan moyenne(m)	0.11	0,13	0,06	**0,05**	**0,05**	**0,05**	**0,05**
Ecart-type (m)	0,08	**0,04**	0,06	0,07	0,07	0,07	0,07
Seuil du Tukey	0,38	×	×	×	×	×	×

TABLE 4.2 – **Qualité de la reconstruction en fonction du facteur α sur la séquence Synthèse 1.** De faibles variations du facteur α entraînent des modifications importantes sur la reconstruction SLAM. Les meilleurs résultats sont en vert tandis que les résultats dégradés sont en rouge.

dèle de l'environnement. Dans la suite de ce mémoire, nous dénoterons cet ajustement de faisceaux par l'acronyme ST-CBA (pour *Single-Term Constrained Bundle Adjustment*). De plus, afin d'éviter les problèmes de pondération liés à la profondeur des points 3D (plus de détails concernant ce problème peuvent être trouvés dans l'annexe A), la fonction de coût utilisée se présente sous la forme d'une erreur de reprojection. Notons que la fonction retenue peut être mise en relation avec l'approche retenue par Vacchetti et al. (2004) pour le suivi d'objets 3D.

4.2.1 Fonction de coût proposée

L'idée de l'approche que nous proposons est de favoriser chacun des points $\mathcal{Q}^i$ à être sur leur plan le plus proche $\mathbf{\Pi}_{h_i}$ (figure 4.2(a)). Pour cela, nous allons passer par un point intermédiaire, noté $\mathcal{Q}'^i$, qui représentera la position la plus cohérente du point $\mathcal{Q}^i$ si on le considère comme étant sur le modèle. Ce point $\mathcal{Q}'^i$ est défini comme étant l'isobarycentre des rétroprojections de $(\mathbf{q}_j^i)_j$ sur le plan $\mathbf{\Pi}_{h_i}$ (figures 4.2(b) et 4.2(c)).

Puisque le point $\mathcal{Q}^i$ résulte de la triangulation de ces observations 2D $(\mathbf{q}_j^i)_j$, déplacer le point $\mathcal{Q}^i$ vers le point $\mathcal{Q}'^i$ équivaut à faire converger les rayons optiques issus des observations $(\mathbf{q}_j^i)_j$ sur le point $\mathcal{Q}'^i$. Ce résultat est également équivalent à minimiser l'erreur de reprojection entre le point $\mathcal{Q}'^i$ et les points d'intérêt qui lui sont associés $(\mathbf{q}_j^i)_j$ (figure 4.2(d)).

La fonction de coût proposée $\mathcal{F}_{ST}$ peut s'écrire sous cette forme :

$$\mathcal{F}_{ST}\left(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E\right) = \sum_{1 \leq i \leq M} \sum_{j \in \mathcal{D}_i} ||\mathbf{q}_j^i - \pi(\tilde{\mathsf{P}}_j \tilde{\mathcal{Q}'}^i(\{\mathsf{C}_k^E\}_{k \in \mathcal{D}_i}, \mathbf{\Pi}_{h_i}))||^2 \tag{4.3}$$

où $\mathcal{D}_i$ est l'ensemble des indices des caméras qui observent le point $\mathcal{Q}^i$. Tout comme dans l'ICP non-rigide (section 3.3), nous considérons que l'association point-plan est constante, c'est à dire que pour chacun des points $\mathcal{Q}'^i$, le plan associé $\mathbf{\Pi}_{h_i}$ est le même durant l'ensemble de

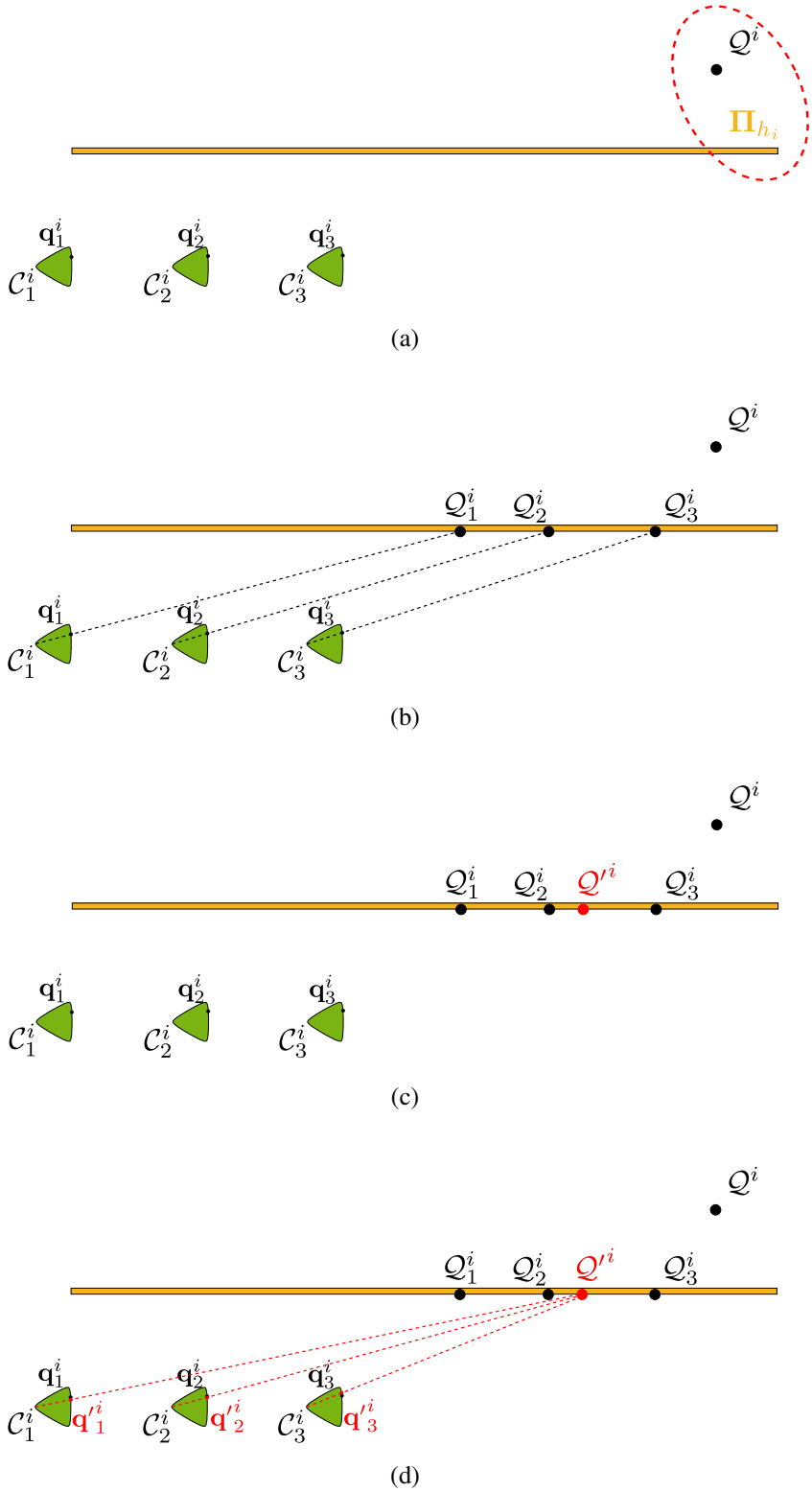


FIGURE 4.2 – **Fonction de coût proposée.** Exemple d'un point 3D $\mathcal{Q}^i$ observé par 3 caméras. Les étapes successives sont décrites par les sous-figures (a) à (d). Les résidus sont les distances 2D entre $(\mathbf{q}_j^i)_j$, les observations de $\mathcal{Q}^i$, et $(\mathbf{q}'^i_j)_j$, les projections de son point 3D associé $\mathcal{Q}'^i$.

la minimisation. Ainsi, la position du point $\mathcal{Q}'^i$ dépend uniquement des paramètres des caméras qui l'observent et de l'équation du plan $\mathbf{\Pi}_{h_i}$. Cela implique que pendant l'optimisation, le mouvement de $\mathcal{Q}'^i$ est cohérent avec le déplacement des caméras qui l'observent. Notons qu'en pratique, l'altitude des caméras n'est pas optimisée. Comme pour l'étape d'ICP non-rigide (section 3.2.3), elle est fixée grâce à un modèle d'élévation du terrain.

4.2.2 Optimisation robuste

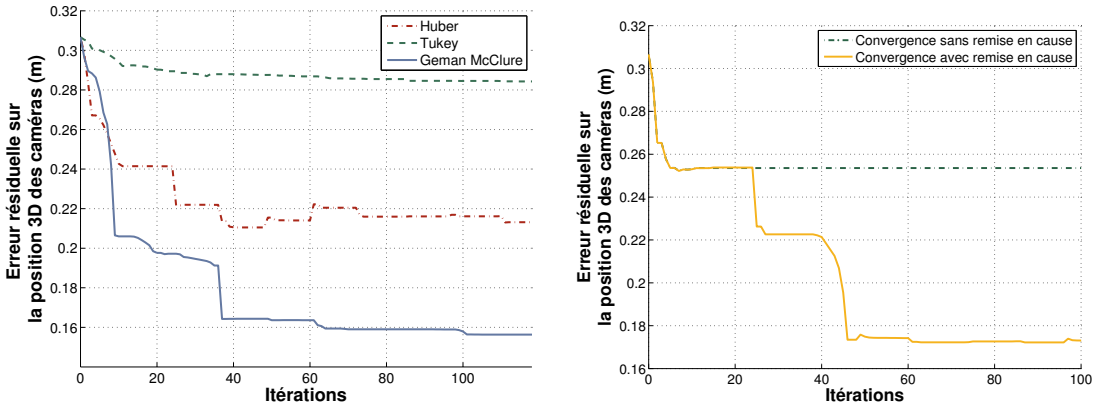
Afin de maximiser les chances d'atteindre la solution optimale de notre problème, il est nécessaire d'assurer la robustesse de la minimisation de l'équation 4.3 à la présence de données aberrantes. Ces données aberrantes ont plusieurs origines possibles : points reconstruits n'appartenant pas au modèle 3D dans la réalité, associations point-plan erronées, *etc.* Ainsi, plusieurs processus robustes ont été mis en place.

4.2.2.1 Robustesse aux points aberrants

Certains points 3D peuvent être mal reconstruits ou ne pas appartenir dans la réalité au modèle 3D (par exemple les points appartenant à un arbre situé devant une façade, à une voiture garée sur le trottoir, *etc.*). Afin d'être robuste à ces points aberrants, nous utilisons un M-estimateur dans la fonction $\mathcal{F}_{ST}$:

$$\mathcal{F}_{ST}\left(\mathsf{C}_1^E,\ldots,\mathsf{C}_N^E\right) = \sum_{1\leq i\leq M}\sum_{j\in\mathcal{D}_i}\rho_{GM}\left(||\mathbf{q}_j^i - \pi(\tilde{\mathsf{P}}_j\tilde{\mathcal{Q}}'^i(\{\mathsf{C}_k^E\}_{k\in\mathcal{D}_i},\mathbf{\Pi}_{h_i}))||\right) \tag{4.4}$$

Le M-estimateur ρ_{GM} retenu ici est le M-estimateur de Geman-McClure (Huber (1981)). En effet, nous avons observé expérimentalement que grâce à l'évolution asymptotique de sa valeur, ce M-estimateur donne une convergence plus rapide et précise que les autres M-estimateurs classiques, à savoir Tukey et Huber (figure 4.3(a)). Le seuil du M-estimateur est réglé de façon automatique grâce au MAD.



(a) (b)

FIGURE 4.3 – **Robustesse de l'ajustement de faisceaux.** (a) montre que le M-estimateur de Geman-McClure semble fournir la meilleure convergence. (b) affiche l'intérêt de remettre en cause l'association point-plan dans l'optimisation.

4.2.2.2 Remise en cause des associations point-plan

Dans la section précédente, nous avons précisé que l'association point-plan est constante au cours de la minimisation. Une mauvaise position initiale du point $\mathcal{Q}^i$ pouvant amener à la sélection d'un mauvais plan, il est possible que nombre d'associations point-plan soient erronées dans le processus d'optimisation. Ceci peut alors empêcher de converger vers la solution optimale (figure 4.3(b)). Tout comme pour l'ICP non-rigide, il est nécessaire de mettre à jour la position des points $(\mathcal{Q}^i)_i$ au cours de l'optimisation afin de remettre en cause les différentes associations point-plan.

Ainsi, la solution globale proposée pour minimiser l'équation 4.3 est un processus itératif semblable à celui de l'ICP non-rigide :

1. Associer les points $(\mathcal{Q}^i)_i$ aux plans les plus proches $(\mathbf{\Pi}_{h_i})_i$.
2. Trouver la pose optimale des caméras en minimisant la fonction de coût (4.3) grâce à l'algorithme de Levenberg-Marquardt (Levenberg (1944)).
3. Retrianguler les points $(\mathcal{Q}^i)_i$ à partir des nouvelles poses de caméras obtenues. Notons que la position de $\mathcal{Q}^i$ est obtenue par triangulation des observations $(\mathbf{q}^i_j)_j$ puisque cette position n'est pas nécessairement la position de $\mathcal{Q}'^i$.
4. Recommencer à l'étape 1.

Dans la pratique, nous effectuons 10 itérations du processus, avec 10 itérations pour chacun des Levenberg-Marquardt. La figure 4.4 montre brièvement les résultats obtenus après la méthode totale proposée (*i.e.* l'ICP non-rigide et l'ajustement de faisceaux ST-CBA) sur la même séquence de synthèse que la figure 3.4, page 59. En comparant les figures 4.4 et 3.4, nous pouvons observer que la précision des positions des caméras a été nettement améliorée. De même, après l'ajustement de faisceaux proposé, le nuage de points reconstruit et le modèle 3D sont plus cohérents.

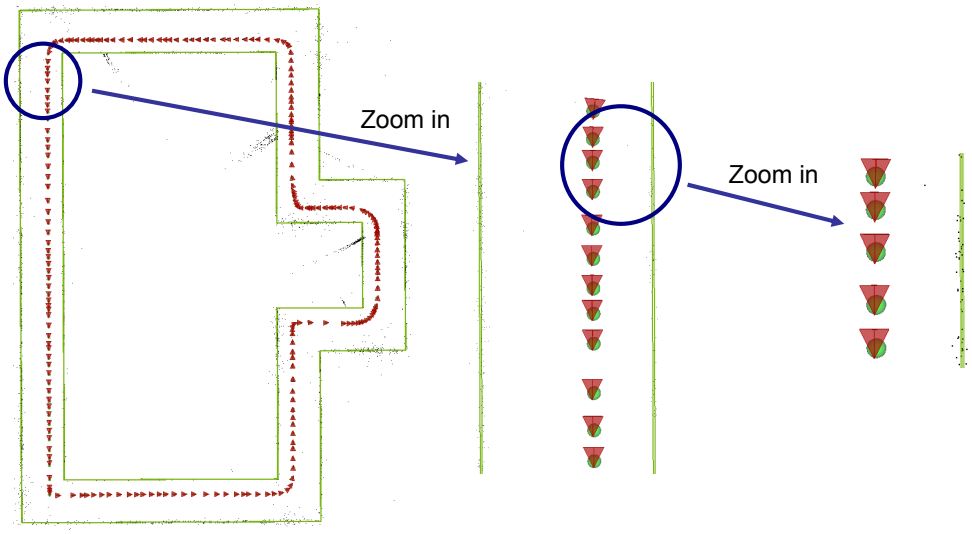


FIGURE 4.4 – **Résultat de la reconstruction après l'ajustement de faisceaux proposé.** Les pyramides rouges sont les caméras reconstruites et les sphères vertes représentent la vérité terrain. L'erreur résiduelle sur la position des caméras, même le long de la trajectoire, est faible.

Dans le chapitre suivant, nous présentons des résultats détaillés sur les deux étapes du processus proposé, à savoir l'ICP non-rigide et l'ajustement de faisceaux ST-CBA.

CHAPITRE 5

Résultats expérimentaux

Ce chapitre a pour but de valider expérimentalement la méthode de construction de base d'amers visuels proposée dans les deux chapitres précédents. Tout d'abord, des séquences de synthèse seront étudiées afin de mesurer quantitativement la précision des reconstructions obtenues ainsi que la robustesse de la méthode proposée. Ensuite, différentes séquences réelles seront traitées afin de vérifier la faisabilité de la méthode dans des conditions réalistes.

5.1 Evaluation quantitative sur des données de synthèse

Ne disposant pas de données (séquence vidéo, modèle 3D de ville) associées à une vérité terrain (trajectoire réelle de la caméra, modèle 3D précis de l'environnement), nous avons eu recours à des données de synthèse pour mener une évaluation quantitative de notre méthode. Dans la pratique, les séquences de synthèse sont des séquences vidéos 640×480 réalisées à partir du logiciel 3DS Max (figure 5.1). La séquence vidéo obtenue est alors utilisée en entrée de l'algorithme de SLAM.



(a) (b) (c)

FIGURE 5.1 – **Extraits d'une vidéo de synthèse.** Les séquences de synthèse sont des vidéos réalisées à partir d'un logiciel de modélisation 3D.

Deux séquences de synthèse différentes ont été créées (voir le tableau 5.1) afin de mesurer respectivement la précision et la robustesse de la méthode proposée.

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
Synthèse 1	420	218	6848
Synthèse 2	420	184	6258

TABLE 5.1 – **Statistiques sur les reconstructions de synthèse.**

5.1.1 Evaluation de la précision

La première séquence (appelée Synthèse 1, voir section B.1) a été réalisée à partir de l'environnement 3D décrit dans la figure 5.2(a). Le but de cette première expérience est de mesurer la qualité de la méthode dans un environnement parfait. De ce fait, le modèle 3D utilisé pour corriger la reconstruction SLAM est le modèle 3D exact de l'environnement. Il est ainsi possible de décorréler les problèmes liés à la méthode proposée de ceux liés à l'inexactitude du modèle 3D associé. En particulier, l'hypothèse que nous faisons sur la qualité du modèle 3D, à savoir que sa distance à la scène réelle est assimilable à une gaussienne centrée en 0 et de faible écart-type (section 2.6.2.3), est ici parfaitement respectée.

5.1.1.1 Reconstruction SLAM originale

La figure 5.2(b) affiche la reconstruction obtenue à la sortie de l'algorithme de SLAM. Rappelons que dans notre cas, nous utilisons l'algorithme d'odométrie visuelle proposé par Mouragnon et al. (2006). Il est possible d'observer sur cette figure la dérive inhérente au SLAM. En effet, nous pouvons voir que la trajectoire ne boucle pas, c'est à dire que la position de la première et de la dernière caméra sont différentes, alors qu'elles sont identiques dans la séquence vidéo. Ce résultat est corroboré par la figure 5.3, dans laquelle on voit nettement la dérive du facteur d'échelle (ici un écrasement) tout au long de la trajectoire. Pour cette figure, le facteur d'échelle est calculé comme étant le ratio de la distance inter-caméra entre la reconstruction et la vérité terrain.

5.1.1.2 Résultats de l'ICP non-rigide

La première étape à réaliser est de fragmenter la reconstruction SLAM. La figure 5.2(b) donne le résultat de cette fragmentation : les sphères rouges sont les extrémités des segments et les points 3D reconstruits sont colorés en fonction du segment de trajectoire auquel ils sont associés. La fragmentation obtenue avec la méthode automatique proposée semble cohérente avec celle qu'un utilisateur aurait pu réaliser manuellement.

Une fois la fragmentation effectuée, nous avons initialisé l'ICP non-rigide en simulant l'utilisation de données GPS (figure 5.2(c)) : chacune des extrémités des fragments a été placée autour du modèle 3D (via une interface graphique) avec une erreur proche de celle obtenue avec un système GPS classique. La figure 5.2(d) illustre la reconstruction obtenue suite à l'étape d'ICP non-rigide. Il est intéressant de noter que désormais, la boucle de trajectoire est reformée alors qu'aucune contrainte de boucle n'est intégrée à la méthode. Ceci valide le fait que l'ICP non-rigide permet de retrouver la cohérence globale de la géométrie de la reconstruction.

Le tableau 5.2 confirme numériquement cette amélioration. Les statistiques de ce tableau ont été calculées sur les 5591 points 3D retenus comme non-aberrants par l'ICP non-rigide, parmi les 6848 points reconstruits par le SLAM. En particulier, on peut noter que la distance

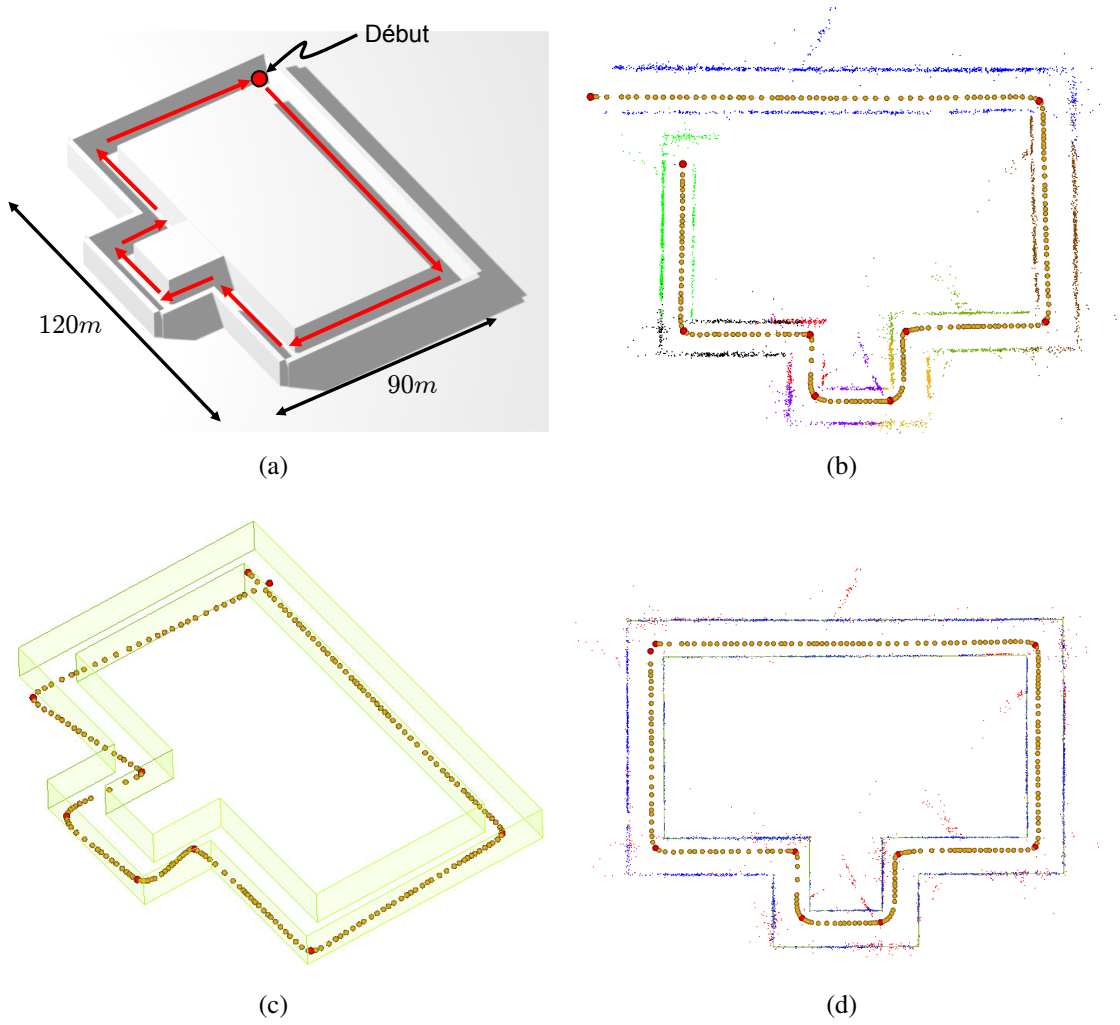


FIGURE 5.2 – **Déroulement de la méthode sur la séquence Synthèse 1.** (a) est le modèle 3D utilisé pour générer et traiter la séquence. (b) est la reconstruction à la sortie de l'algorithme de Mouragnon et al. (2006). La fragmentation de la reconstruction y est également représentée à l'aide d'un code couleur sur les points 3D reconstruits. Les extrémités des fragments sont représentées par les sphères rouges. (c) montre l'initialisation de la reconstruction avant l'ICP non-rigide. (d) affiche la reconstruction finalement obtenue : les points 3D bleus sont les points conservés par le M-estimateur et les points rouges sont les points aberrants.

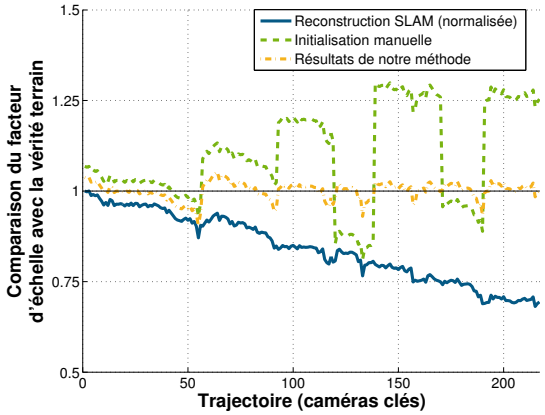


FIGURE 5.3 – **Evolution du facteur d'échelle sur la séquence Synthèse 1.** La dérive du facteur d'échelle est visible sur la reconstruction originale. Avec notre méthode, la facteur d'échelle obtenu est centré sur 1 (la courbe d'une reconstruction parfaite étant la droite $y = 1$).

moyenne entre la position de la caméra reconstruite et sa vérité terrain passe d'environ 4 mètres à 50 centimètres après l'ICP non-rigide.

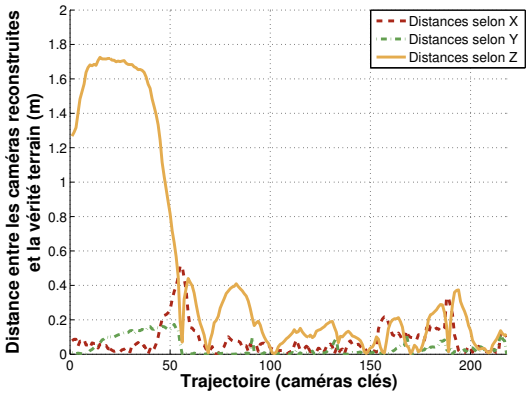
La figure 5.4(a) représente l'erreur de positionnement des caméras clés sur l'ensemble de la trajectoire. Les résultats qu'elle affiche nuancent ceux du tableau 5.2. En effet, on voit que l'erreur résiduelle sur la position des caméras peut être encore très importante pour certaines, en particulier dans la direction de leur axe focal. Comme nous l'avons souligné dans la section 3.4, cette erreur résiduelle est due à l'hypothèse de travail que nous faisons dans l'ICP non-rigide. En effet, nous observons que considérer le facteur d'échelle comme étant constant sur les lignes droites (section 3.2.1) est généralement uniquement une vision très approximative du phénomène réel.

	Avant ICP non-rigide	Après ICP non-rigide	Après ICP non-rigide + ajustement de faisceaux classique	Après ICP non-rigide + ajustement de faisceaux par combinaison linéaire ($\alpha = 0,5$)	Après ICP non-rigide + ajustement de faisceaux ST-CBA
Distance moyenne entre les caméras et la vérité terrain (m)	4,61	**0,51**	0,59	0,32	**0,14**
Ecart-type (m)	2,25	**0,59**	0,54	0,23	**0,10**
Distance médiane entre les caméras et la vérité terrain (m)	5,16	**0,22**	0,38	0,23	**0,10**
Distance point-plan moyenne (m)	3,37	**0,11**	0,15	**0,05**	0,08
Ecart-type (m)	3,9	**0,08**	0,10	**0,07**	0,08
Seuil du Tukey	×	0,38	×	×	×

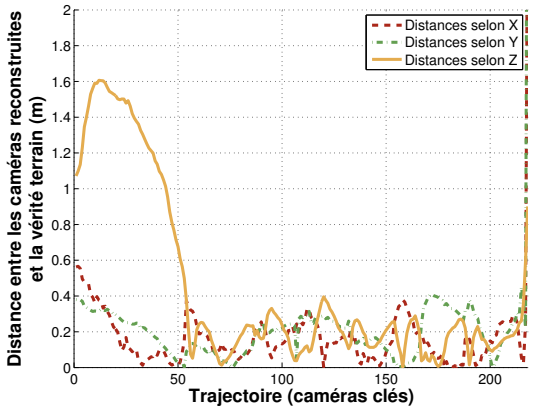
TABLE 5.2 – **Résultats numériques obtenus sur la séquence Synthèse 1.** Chaque valeur est une moyenne sur l'ensemble de la reconstruction.

5.1.1.3 Apport de l'ajustement de faisceaux ST-CBA

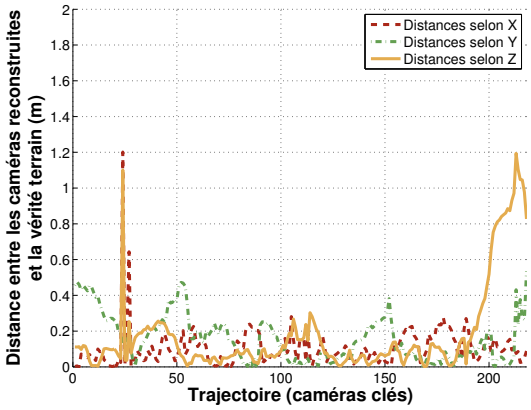
La figure 5.4(d) fournit les résultats de la reconstruction obtenue lorsqu'ont été appliqués à la fois l'ICP non-rigide puis l'ajustement de faisceaux proposé. Nous voyons dès lors que l'erreur résiduelle de positionnement est très nettement diminuée, pour obtenir finalement une erreur de positionnement moyenne de 14 centimètres.



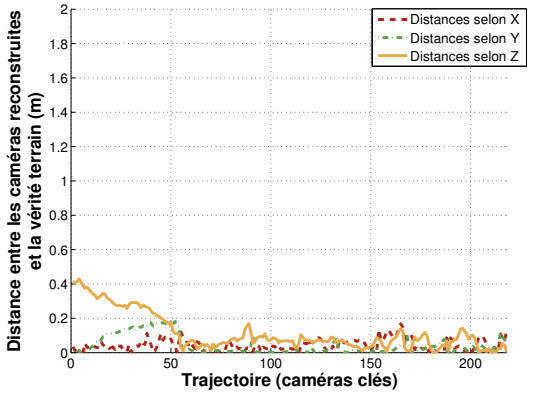
(a) Après l'ICP non-rigide



(b) Après l'ICP non-rigide + l'ajustement de faisceaux classique



(c) Après l'ICP non-rigide + l'ajustement de faisceaux par combinaison linéaire ($\alpha = 0,5$)



(d) Après l'ICP non-rigide + l'ajustement de faisceaux ST-CBA

FIGURE 5.4 – **Erreur de positionnement des caméras sur la séquence Synthèse 1.** Le repère (X, Y, Z) est relatif à chacune des caméras : Z correspond à l'axe optique, X la direction latérale et Y l'altitude.

Les figures 5.4(b), 5.4(c) ainsi que le tableau 5.2 permettent de comparer numériquement les résultats de l'ajustement de faisceaux proposé avec l'ajustement de faisceaux classique (section 4.1.1) ainsi que l'ajustement de faisceaux par combinaison linéaire des résidus (section 4.1.2). Notons que les résultats sur l'ajustement de faisceaux par combinaison linéaire ont été obtenus pour $\alpha = 0.5$, cette valeur donnant les meilleurs résultats sur cette séquence (voir l'annexe A). Nous retrouvons numériquement les observations faites dans le chapitre 4. En effet, du fait de la perte des contraintes liées au modèle 3D, l'ajustement de faisceaux classique tend à dégrader les résultats. De plus, même si la méthode basée sur la combinaison linéaire des

résidus permet de baisser l'erreur de positionnement des caméras, la méthode proposée fournit une précision deux fois meilleure sur cette séquence. Il est également intéressant de noter que la précision des positions des caméras est en effet meilleure pour notre méthode alors que la distance point-modèle moyenne est supérieure. Cela revient à dire que la précision sur la position des caméras est meilleure alors que la cohérence entre le nuage de points et le modèle est moins bonne. Cela tend à montrer que la méthode que nous proposons ne force pas les points 3D à être sur le modèle mais qu'elle prend intrinsèquement en compte la dispersion pouvant exister entre le nuage de points et le modèle.

5.1.2 Evaluation de la robustesse

Comme annoncé précédemment, une seconde séquence de synthèse (appelée Synthèse 2, voir section B.2) a été réalisée de façon à analyser la robustesse de la méthode à l'utilisation d'un modèle 3D erroné (car simplifié) de l'environnement. Etudier la robustesse revient dans ce cas à regarder si la méthode proposée conserve la géométrie locale du nuage de points, information qui n'apparaît pas sur le modèle 3D utilisé.

Pour cela, nous avons complexifié l'environnement réel dans lequel est tournée la vidéo (figure 5.5(a)), alors que le modèle 3D fourni à la méthode est uniquement une approximation de la géométrie réelle de la scène (figure 5.5(b)).

5.1.2.1 Robustesse de la reconstruction à un modèle 3D erroné

La reconstruction obtenue après notre méthode pour cette nouvelle séquence apparaît sur la figure 5.5(d). Trois tests différents ont été réalisés de façon à tester la robustesse de notre méthode dans différentes conditions. Pour rappel, notre hypothèse de travail est que la distribution de l'erreur entre le modèle 3D et l'environnement qu'il décrit peut être assimilée à une gaussienne centrée en 0 et de faible écart-type (voir la section 2.6.2.3 pour plus de détails).

Test 1 : hypothèse de travail respectée (caméras clés 0 à 45). Le premier test réalisé suit l'hypothèse de travail fixée. Le mur (annoté 1 sur la figure) n'est en réalité pas plan mais a la forme d'une vague. Nous pouvons voir que cette structure est bien reconstruite alors que l'information n'apparaît pas dans le modèle 3D.

Test 2 : hypothèse de travail non respectée localement (caméras clés 150 à 170). Le deuxième test a été de placer un coin incurvé dans la scène réelle (annoté 2). Ainsi, dans ce coin, notre hypothèse de travail n'est plus respectée. Néanmoins, nous voyons que cette erreur locale ne semble pas perturber la reconstruction. Cela peut s'expliquer par le fait que, l'erreur étant très locale, chaque caméra clé observe à la fois des points 3D qui sont sur la partie erronée du modèle mais également des points 3D sur la partie correcte. Ainsi, pour chacune des caméras, les points 3D qu'elles observent suivent majoritairement l'hypothèse de travail fixée.

Test 3 : hypothèse de travail non respectée à grande échelle (caméras clés 60 à 85). Le dernier test vise à analyser le comportement de la méthode lorsque notre hypothèse de travail n'est pas respectée sur une large zone (annotée 3.1 et 3.2). En pratique, un mur entier a été fortement incurvé. Précisément, la distance entre le modèle 3D et l'environnement au milieu du mur est de 2 mètres. Deux cas de figure sont à distinguer :

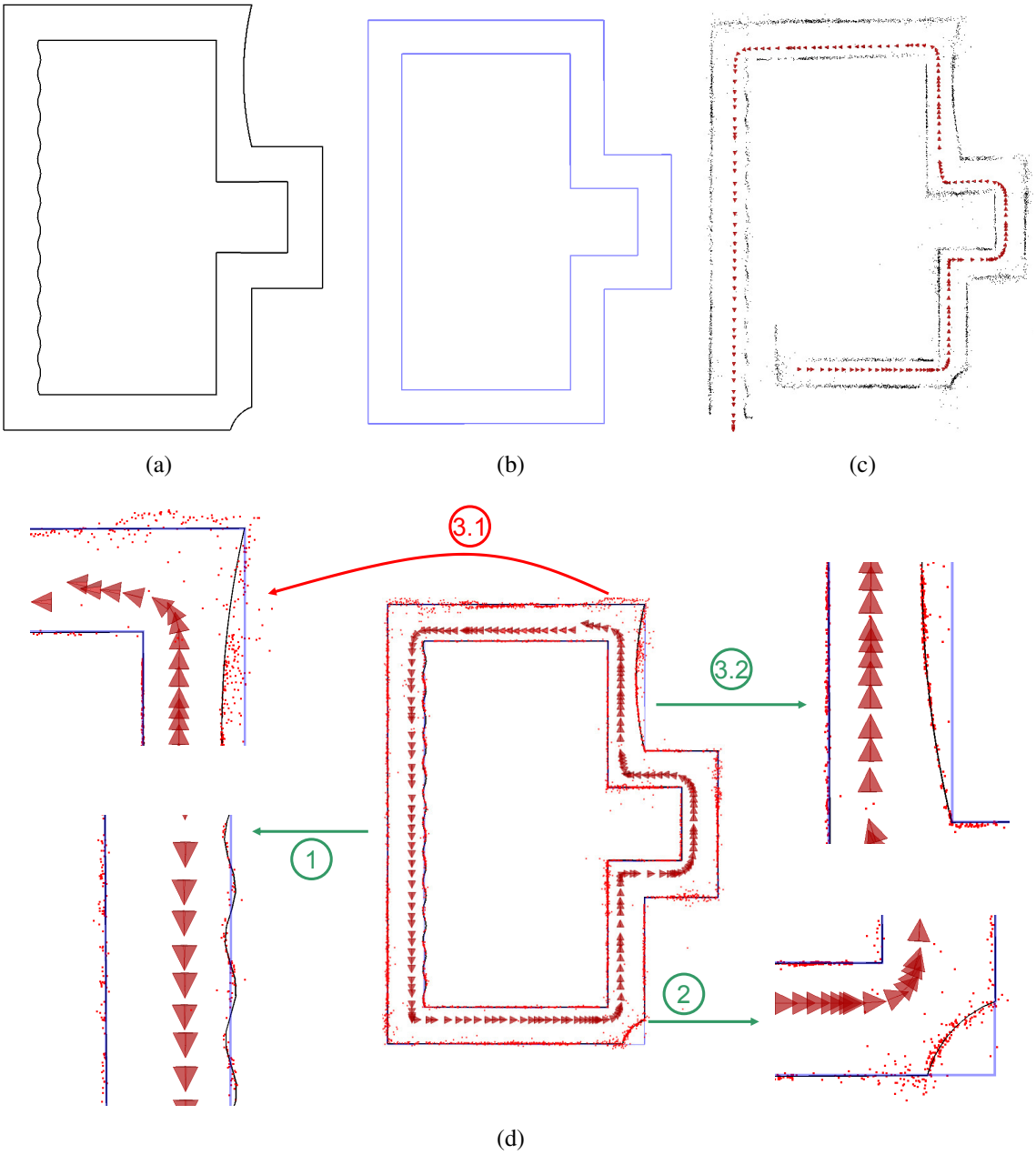


FIGURE 5.5 – **Robustesse à un modèle 3D erroné.** (a) est le modèle 3D utilisé pour créer la séquence vidéo. (b) est le modèle 3D simplifié fourni à la méthode. (c) est la reconstruction originale obtenue avec la méthode proposée par Mouragnon et al. (2006). (d) est la superposition du modèle réel (en gris), du modèle fourni (en bleu) et de la reconstruction finalement obtenue (en rouge). Même si le modèle 3D fourni est une forte approximation de la réalité en plusieurs endroits, la reconstruction obtenue est quasiment toujours en accord avec la scène réelle.

- ▷ Pour la première partie du mur (annotée 3.1), on remarque que les points reconstruits ont tendance à être plaqués sur le modèle 3D, ce qui entraîne localement une mauvaise reconstruction des points et des caméras. A cet endroit, les caméras n'observent que des points sur le mur erroné et la distribution de l'erreur n'est plus centrée en 0 mais possède un biais important. Notons cependant qu'afin de rentrer dans nos hypothèses de travail, et conformément à ce qui est généralement effectué dans la réalité, il suffirait de modéliser ce type de larges murs incurvés comme un ensemble de plans.

- ▷ Dans la seconde partie du mur (annotée 3.2), les caméras observent à nouveau une majorité de points 3D situés sur une partie correcte du modèle (sur le mur de l'autre côté de la route et en face). Nous sommes alors dans les mêmes conditions que lors du test 2. Dans ce cas, on peut s'apercevoir que la position des caméras est à nouveau cohérente et que le mur incurvé est bien reconstruit. De plus, toute la suite de la reconstruction est cohérente avec la scène réelle.

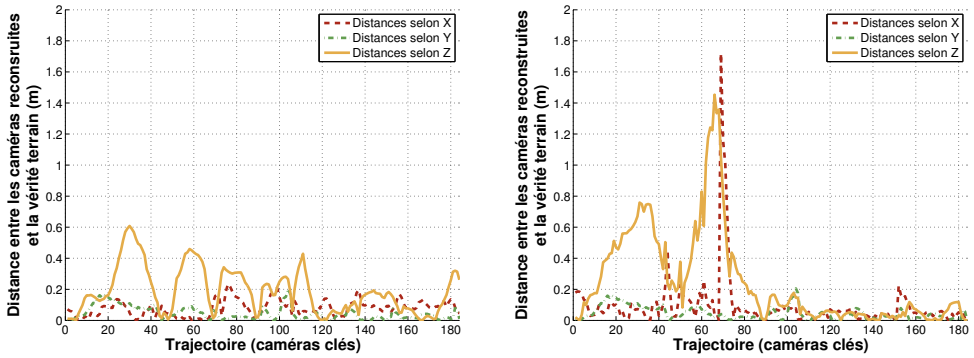
5.1.2.2 Précision obtenue

Si la robustesse (en terme de reconstruction locale) vient d'être montrée expérimentalement, il est cependant intéressant de regarder ce que devient alors la précision de la reconstruction obtenue.

Ces résultats sont mis en avant par la figure 5.6 ainsi que par le tableau 5.3. Trois remarques principales peuvent en être tirées. Tout d'abord, la figure 5.6(b) met clairement en avant l'erreur de reconstruction observée lors du test 3 : aux environs de la caméra clé 70, l'erreur de positionnement des caméras atteint jusqu'à 1,4 mètre d'erreur. De plus, on peut remarquer que sur le premier segment de trajectoire qui présente le mur en vague (entre les caméras clés 1 et 45), l'ajustement de faisceaux à tendance à dégrader légèrement les résultats (de l'ordre de 20 centimètres au maximum). Néanmoins, ce résultat est à nuancer vis à vis du résultat obtenu après l'ICP non-rigide sur cette portion de trajectoire. En effet, la figure 5.6(a) montre que la précision de la reconstruction après l'ICP non-rigide est très bonne. On peut d'ailleurs noter que la distance moyenne des caméras à la vérité terrain est de 18 centimètres après l'ICP non-rigide (tableau 5.3), ce qui est proche des résultats obtenus après l'ajustement de faisceaux lorsque le modèle est parfait. Ce bon résultat est équivalent à dire que le facteur d'échelle a été bien estimé sur cette séquence par l'algorithme de SLAM. Néanmoins, on note que pour la fin de la séquence (après la caméra 70), la précision des caméras est à nouveau améliorée par l'ajustement de faisceaux. Ceci est confirmé par le calcul de la médiane des distances entre les caméras et leur vérité terrain (tableau 5.3). En effet, la médiane peut être vue comme une moyenne robuste aux données aberrantes (ici la partie mal reconstruite de la trajectoire). En particulier, on peut voir que le coin incurvé (correspondant au test 2) ne semble pas influer sur la précision obtenue pour les caméras qui l'observent.

Notons que les statistiques sur la distance entre les points 3D reconstruits et le modèle 3D utilisé ne sont pas réalisées pour cette séquence. En effet, ce modèle 3D étant erroné dans cette expérience, la distance point-plan n'apporterait pas d'information sur la qualité de la reconstruction.

Cette expérience a permis de mettre en évidence que, tant que notre hypothèse de travail est respectée pour la majorité du modèle 3D, la méthode proposée est robuste aux erreurs et imprécisions du modèle 3D fourni en entrée de la méthode.



(a) Après l'ICP non-rigide

(b) Après l'ICP non-rigide + l'ajustement de faisceaux ST-CBA

FIGURE 5.6 – **Précision obtenue avec un modèle 3D erroné.** Le repère (X, Y, Z) est relatif à chacune des caméras : Z correspond à l'axe optique, X la direction latérale et Y l'altitude.

	Après l'ICP non-rigide	Après l'ICP non-rigide + l'ajustement de faisceaux ST-CBA
Distance moyenne entre les caméras et la vérité terrain (m)	**0,18**	0,24
Ecart-type (m)	**0,14**	0,30
Distance médiane entre les caméras et la vérité terrain (m)	0,16	**0,11**

TABLE 5.3 – **Statistiques sur la précision obtenue avec un modèle erroné.**

5.2 Evaluation qualitative sur des données réelles

Des séquences réelles ont été tournées dans le but de confronter la méthode proposée aux conditions rencontrées dans un environnement réel. Ces séquences, réalisées dans Versailles, mettent en avant des exemples d'occultation des bâtiments observés (arbres, voitures), de bâtiments approximatifs par rapport à la géométrie réelle, *etc.*

5.2.1 Données utilisées

Les séquences réelles ont été enregistrées avec une caméra perspective monochrome GUPPY F-046B (capteur 1/2"). L'optique utilisée est une optique à large champ de vue (focale de 4mm). Le calibrage de la caméra a été effectué à l'aide d'une méthode classique utilisant une cible 2D connue (Lavest et al. (1998)).

Cette caméra enregistre des images de résolution 640×480 à une fréquence de 30 images par seconde (figure 5.7). La courte focale induit des distorsions importantes dans l'image qu'il est nécessaire de traiter dans l'algorithme de reconstruction. Les paramètres de distorsion de la caméra sont également obtenus lors de son calibrage.

Le modèle 3D de l'environnement utilisé (figure 5.8(b)) est un modèle simple, uniquement composé de plans verticaux décrivant les différentes façades (comme décrit à la section 2.6.2.3). Ce modèle a été créé à partir des outils de mesure fournis sur le site du Géoportail. On considère



(a) (b) (c)

FIGURE 5.7 – **Extraits des séquences vidéos à Versailles.** Les séquences réelles sont des vidéos 640×480 qui ont été enregistrées avec une caméra perspective simple.

donc que ce modèle a une précision d'environ 2 mètres. Le fait que ce modèle ne décrive que grossièrement l'environnement réel sera mis en évidence par la figure 5.11(b), page 81.

5.2.2 Résultats obtenus

Les séquences réelles, appelées Versailles 1 (voir section B.3) et Versailles 2 (voir section B.4), sont des trajets de respectivement 1,5 et 2 kilomètres (voir tableau 5.4). Les différentes étapes de la méthode sur ces séquences sont représentées dans la figure 5.8. Les reconstructions obtenues à partir de l'algorithme de SLAM (dans notre cas celui de Mouragnon et al. (2006)) illustrent bien le phénomène de dérive inhérent à ce type de méthode (figures 5.8(d) et 5.8(g)). En effet, ces reconstructions ont été placées manuellement dans le même repère que le modèle 3D et le facteur d'échelle a été fixé de façon à ce que les deux premiers virages soient cohérents avec ce modèle. On peut alors observer que la trajectoire devient incohérente dès le virage suivant.

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
Versailles 1	1500	240	16761
Versailles 2	2000	313	14780

TABLE 5.4 – **Statistiques sur les séquences réelles.**

Comme pour les séquences de synthèse, l'initialisation de la position des extrémités des fragments de trajectoire a été réalisée en simulant les erreurs liées au système GPS classique grâce à une interface graphique (figures 5.8(e) et 5.8(h)). Les résultats obtenus suite à l'application de notre méthode illustrent bien que celle-ci permet de corriger la dérive du SLAM pour l'ensemble de la trajectoire. En particulier, la superposition des reconstructions finales avec l'image satellite de l'environnement parcouru (figure 5.9) met en avant le fait que ces reconstructions respectent la géométrie de la scène réelle : la trajectoire de la caméra est cohérente avec le tracé de la route sur l'ensemble du trajet et le nuage de points reconstruit semble décrire correctement le contour des différents bâtiments.

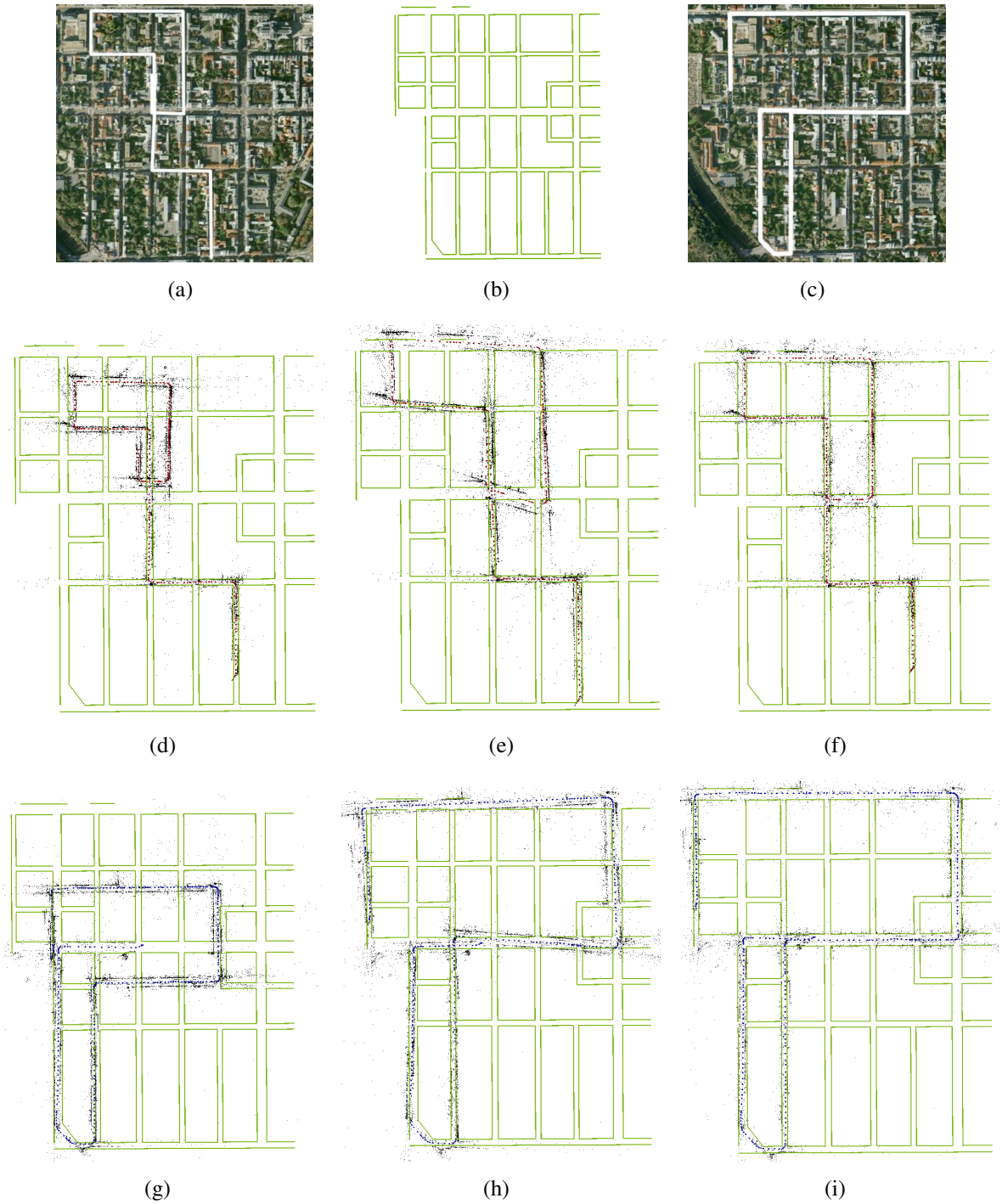


FIGURE 5.8 – **Séquences de Versailles.** La première ligne présente les informations sur les séquences réalisées : les deux trajectoires (a) et (c) et le modèle 3D utilisé (b). La deuxième et la troisième ligne affichent l'état des reconstructions à différents moments du processus respectivement pour Versailles 1 et Versailles 2 : la reconstruction initiale obtenue avec la méthode de Mouragnon et al. (2006) (d,g), l'initialisation avant l'ICP non-rigide (e,h) et le résultat après notre méthode (f,i).



(a) (b)

FIGURE 5.9 – **Recalage des reconstructions obtenues sur les images satellites.** Les reconstructions s'alignent correctement avec l'image satellite de la zone parcourue.

5.2.3 Evaluation qualitative de la précision obtenue

Ne disposant pas de vérité terrain pour les séquences réelles, il nous est impossible de quantifier numériquement la qualité de la reconstruction finale ainsi que de mesurer l'apport de l'ajustement de faisceaux par rapport à l'ICP non-rigide seul. Néanmoins, quelques expériences et résultats supplémentaires effectués permettent d'évaluer qualitativement la précision de la méthode proposée.

5.2.3.1 Apport de l'ajustement de faisceaux ST-CBA

Sans vérité terrain sur la position des caméras, il est particulièrement difficile de mesurer le gain de précision apporté par l'ajustement de faisceaux. Cependant, on peut mesurer visuellement ce gain en observant non pas les caméras mais la structure de la scène reconstruite (c'est à dire le nuage de points) puisque celle-ci est directement liée à la pose des caméras. La figure 5.10 donne un exemple d'amélioration de la structure observée sur la première séquence réelle. Cette amélioration concerne un pan de mur (entouré sur l'image) perpendiculaire à la façade du bâtiment, juste après un virage. Ce mur n'est pas modélisé sur le modèle 3D de l'environnement mais apparaît nettement sur la reconstruction SLAM obtenue.

Cependant, après l'ICP non-rigide, on peut voir que ce mur est mal reconstruit : les points qui le décrivent ont une dispersion importante et le mur n'est pas perpendiculaire à la façade du bâtiment. Cela peut s'expliquer par le fait que, lors de l'ICP non-rigide, la fragmentation de la reconstruction crée au niveau des virages une discontinuité à la fois sur le facteur d'échelle estimé et sur le lien entre les caméras et les points 3D observés. Or, les points 3D de ce mur ne dépendent pas tous du même fragment et sont donc optimisés indépendamment, ce qui implique l'absence de cohérence de cet ensemble de points.

Après l'étape supplémentaire d'ajustement de faisceaux, on peut alors observer que ce mur est mieux reconstruit : la dispersion des points est moindre et son orthogonalité avec la façade est retrouvée. Ceci tend à montrer que l'ajustement de faisceaux a permis de rétablir la continuité

dans les relations entre caméras et points et que, de plus, la position des caméras avant et après le virage a été corrigée.

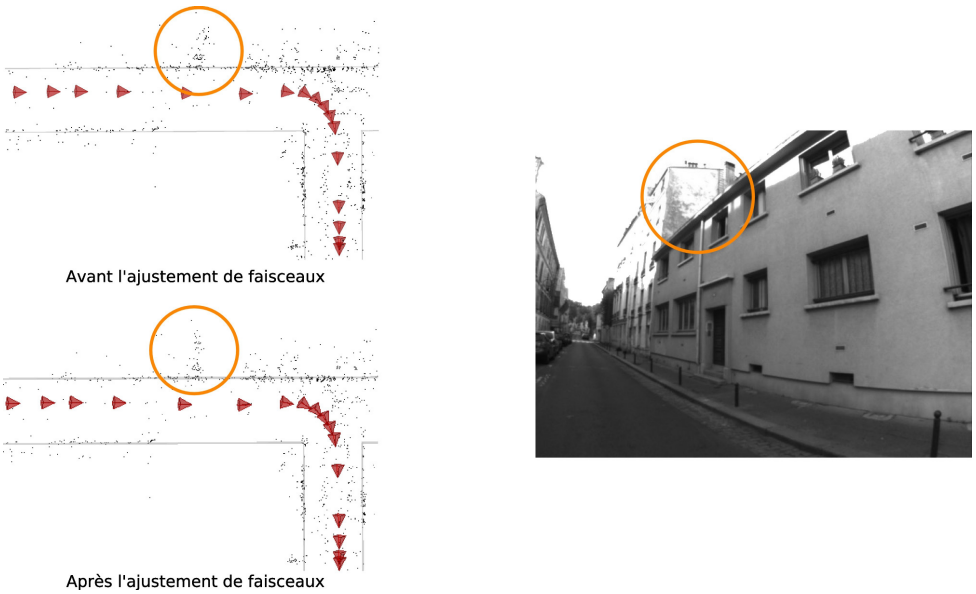


FIGURE 5.10 – **Exemple d'amélioration de la géométrie locale.** L'ajustement de faisceaux ST-CBA permet d'améliorer la géométrie locale de la reconstruction obtenue après l'ICP non-rigide.

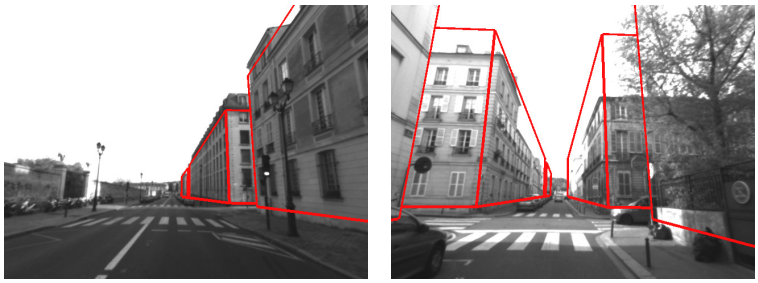
5.2.3.2 Précision obtenue

Comme nous l'avons vu précédemment, nous ne disposons pas des données de vérité terrain qui permettraient d'apprécier la précision de la position obtenue pour les caméras reconstruites. Néanmoins, une première expérience qui consiste à projeter le modèle 3D dans les images clés reconstruites (figure 5.11) met en avant que le recalage entre les images et ce modèle semble correct. Notons que cela ne permet pas de conclure que la position des caméras est très précise mais uniquement qu'elle l'est suffisamment pour projeter correctement le modèle 3D dans les images. Par ailleurs, c'est ce dernier aspect qui nous intéresse en particulier dans le cadre de nos travaux.

Pour apprécier visuellement la précision de la position 3D des caméras, nous avons superposé les deux reconstructions obtenues (figure 5.12), ces deux reconstructions ayant été réalisées dans le même quartier de Versailles. Le résultat obtenu permet de montrer que la précision des reconstructions est suffisante pour distinguer les différentes voies de circulation d'une même rue avec des voies de circulation de taille classique (entre 2 et 4 mètres de large).

5.3 Discussion

Au terme des différentes expérimentations réalisées, il est nécessaire de discuter la performance de l'approche proposée, en particulier par rapport à l'application de relocalisation visée, et de pointer les limites et difficultés rencontrées par la méthode dans son état actuel.



(a) (b)

FIGURE 5.11 – **Projection du modèle 3D de ville dans les caméras clés.** Le modèle 3D grossier de l'environnement peut être proche (a) ou éloigné (b) de la géométrie réelle de la scène parcourue.

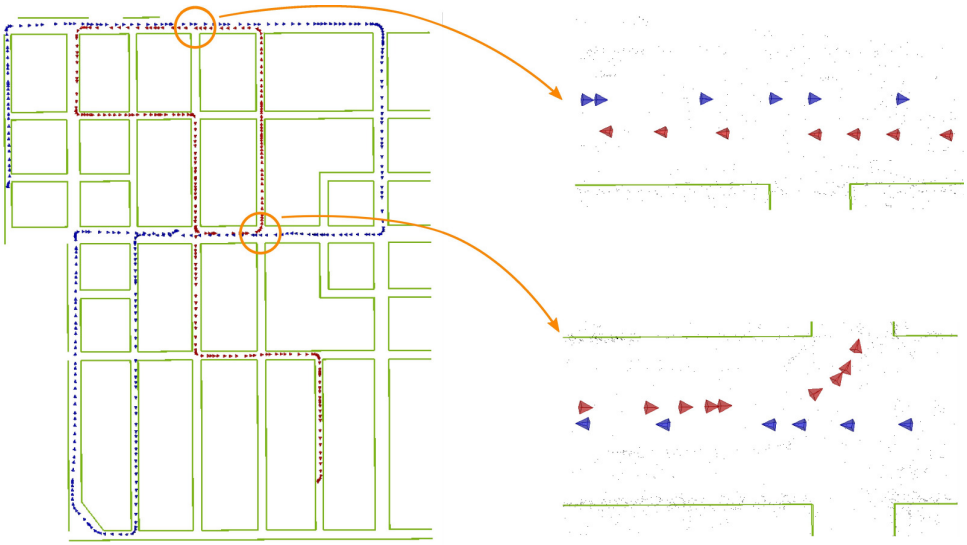


FIGURE 5.12 – **Fusion de plusieurs reconstructions.** Les deux reconstructions des séquences de Versailles ont été réalisées dans le même quartier. Les zooms montrent qu'il est possible de discerner les différentes voies.

5.3.1 Performance de l'approche proposée

La performance de la méthode est liée à la fois à la précision des reconstructions obtenues ainsi qu'au temps de traitement nécessaire à l'obtention de ces reconstructions.

5.3.1.1 Qualité des résultats obtenus

Les expériences réalisées dans ce chapitre ont permis de montrer que les reconstructions obtenues semblent correctes, même lorsque le modèle 3D de l'environnement comporte des informations simplifiées ou erronées. Cela est en particulier rendu possible par la fonction de coût proposée dans l'ajustement de faisceaux qui permet de prendre en compte l'incertitude sur la précision du modèle. En effet, cette fonction ne force pas les points à se plaquer sur les murs et l'optimisation globale (c'est à dire de l'ensemble des caméras et des points 3D) est suffisamment contrainte pour conserver la précision locale apportée par le nuage de points reconstruit.

Néanmoins, il est à noter que si la géométrie locale du modèle 3D peut être remise en cause par l'information apportée par la reconstruction SLAM, cela n'est plus vrai en ce qui concerne la géométrie globale. En effet, la dérive du facteur d'échelle ne pouvant être quantifiée, la reconstruction SLAM n'apporte pas d'information globale en laquelle on peut avoir confiance, en particulier sur les dimensions de la scène. La géométrie globale, qui se traduit en particulier par l'échelle de la scène dans notre problématique, est donc uniquement apportée par le modèle 3D (à travers la largeur et la longueur des routes par exemple). Cela implique qu'une erreur sur les dimensions du modèle 3D aura des répercussions directes sur la reconstruction finalement obtenue : celle-ci s'alignera au mieux avec le modèle 3D mais sera donc incohérente avec la scène réelle si le modèle l'est lui-même.

Cependant, ce comportement est à nuancer pour deux raisons principales. Tout d'abord, comme nous l'avons dit précédemment, les modèles issus des SIG possèdent généralement une erreur limitée (métrique) et leur précision tend à s'améliorer, des modèles décimétriques étant par exemple en train d'apparaître sur le Géoportail (section 2.6.2.3). De plus, l'application qui est visée dans nos travaux est l'aide à la navigation par réalité augmentée. Or, pour la réalité augmentée, la caméra devra être en priorité recalée par rapport au modèle puisque c'est sur ce modèle que seront ajoutées les informations additionnelles à fournir à l'utilisateur. Au contraire, la précision de la position de la caméra dans le monde n'est pas critique si on se limite à cette seule application.

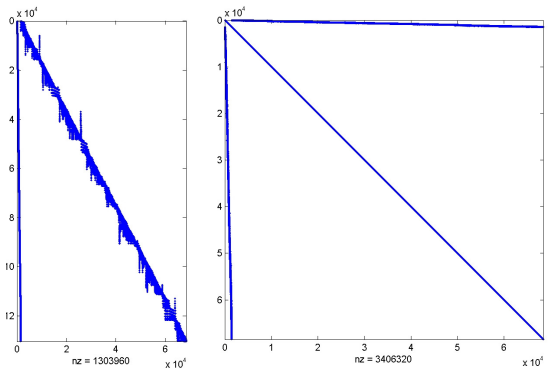
5.3.1.2 Temps de traitement nécessaires

Pour qu'une méthode soit exploitable en pratique, il est nécessaire que le temps de traitement relatif à son exécution soit raisonnable. Nous allons donc ici essayer de donner un ordre de grandeur du temps nécessaire au déroulement de notre méthode.

La première étape de notre approche (*i.e.* l'ICP non-rigide) a été entièrement codée en C++. Avec le code tel qu'il est aujourd'hui, l'ICP non-rigide sur la séquence Versailles 1 s'exécute en moins d'une minute pour 10 itérations des étapes d'association des données et de minimisation de la métrique. Notons cependant que les dérivées sont calculées numériquement dans l'algorithme de Levenberg-Marquardt. Le passage à un calcul analytique permettra donc d'améliorer nettement le temps nécessaire à l'exécution de cette étape.

Dans le cadre de nos expériences, les différents ajustements de faisceaux ont été développés et testés sous Matlab. Les temps de traitement obtenus sont par conséquent importants. En

pratique, l'optimisation sur la séquence Versailles 1 (composée de 10 itérations des étapes d'association des données et de minimisation de la métrique) prend environ 5 heures, le calcul des dérivées étant ici aussi numérique. A titre de comparaison, l'ajustement de faisceaux classique codé en Matlab nécessite environ 3 heures pour converger sur cette même séquence. Cette différence peut s'expliquer par le fait que l'implémentation de l'ajustement de faisceaux ST-CBA se prête moins facilement à une écriture matricielle. Or, c'est ce type d'écriture matricielle qui permet de réduire fortement les temps de calcul sous Matlab. Il est intéressant d'envisager les temps de traitement qu'il serait possible d'obtenir après un passage en C++. Des travaux (en particulier ceux de Mouragnon et al. (2006)) ont montré que l'ajustement de faisceaux classique peut être calculé très efficacement. Pour cela, l'idée est d'exploiter la structure creuse de la hessienne associée à ce problème (figure 5.13). La structure de la hessienne associée à l'ajustement de faisceaux spécifique est également creuse (figure 5.13). Il serait donc intéressant d'étudier la structure exacte de cette matrice et de l'exploiter afin d'accélérer l'inversion de la hessienne au sein du Levenberg-Marquardt. On peut dès lors penser que les temps de calcul nécessaires à la résolution de notre problème pourraient être raisonnablement courts, voire suffisamment faibles pour une utilisation temps-réel dans le cadre d'un ajustement de faisceaux local.



(a) (130396 × 68484) (b) (68484 × 68484)

(c) (55529 × 1440) (d) (1440 × 1440)

FIGURE 5.13 – **Comparaison des matrices jacobiennes et hessiennes.** (a) et (b) sont les matrices jacobiennes et hessiennes de l'ajustement de faisceaux classique. (c) et (d) sont les matrices jacobiennes et hessiennes de l'ajustement de faisceaux ST-CBA proposé.

5.3.2 Limites de la méthode actuelle

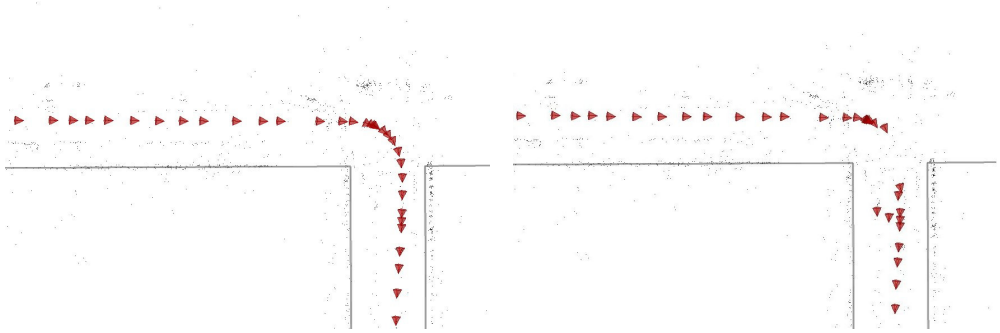
Comme nous l'avons vu, la méthode telle que présentée dans ce mémoire permet d'obtenir des reconstructions 3D d'environnements étendus avec une précision relativement correcte. Cependant, une des limites rencontrées lors des expérimentations est sa robustesse dans le cas où la majorité des points reconstruits ne se situe pas sur les bâtiments. Ce cas de figure peut se produire pour deux raisons principales.

Points 3D occultants. La présence de nombreux points non situés sur les murs peut s'expliquer par la présence d'objets occultants entre la caméra et le bâtiment. Ce phénomène est relativement fréquent dans le cadre automobile à cause par exemple des arbres et des véhicules stationnés. Actuellement, ces points sont filtrés automatiquement par le seuil du M-estimateur à la fois dans l'ICP non-rigide (en fonction de la distance orthogonale au modèle) et dans l'ajustement de faisceaux (en fonction de l'erreur de reprojection). Néanmoins, ce seuil est réglé à l'aide du MAD sous l'hypothèse que la distribution d'erreur est gaussienne, hypothèse qui n'est plus vérifiée lorsque les points occultants sont très nombreux. Cela peut entraîner un mauvais filtrage des points 3D qui ne sont pas sur les bâtiments dans la réalité et donc amener à une convergence de la méthode vers une solution erronée.

Il serait donc intéressant de classifier en amont les points reconstruits pour savoir s'ils sont situés ou non sur le bâtiment et ainsi assurer la convergence de la méthode. Pour cela, des contraintes géométriques pourraient être utilisées : reconstruction de patchs 3D orientés (ceci sera traité dans la section 8.3.2), recherche des plans principaux, *etc.* On peut également penser à utiliser des approches basées sur la segmentation des images afin de distinguer les zones liées à la route, aux bâtiments, aux voitures, *etc.*

Absence de bâtiments. Dans les scènes réelles, il est courant que certaines rues ou portions de rues soient dépourvues de bâtiments. Dans ce cas, puisque les points 3D qui ne sont associés à aucun plan sont retirés de l'optimisation (que ce soit dans l'ICP non-rigide ou l'ajustement de faisceaux), une importante partie de l'information liant les caméras entre elles est perdue. Ce manque d'information ne permet pas d'optimiser certaines caméras, les paramètres de celles-ci n'étant pas suffisamment contraints par la structure conservée. Des expériences ont par ailleurs montré qu'optimiser la pose de ces caméras entraîne des erreurs de reconstruction lors de l'ajustement de faisceaux (figure 5.14). Avec la méthode en l'état, il est donc nécessaire de retirer ces caméras de l'optimisation pour éviter ce phénomène et la pose de ces caméras n'est donc pas raffinée.

Il semble donc nécessaire d'intégrer dans l'ajustement de faisceaux les contraintes liées aux points reconstruits qui ne sont pas situés sur le modèle 3D. Pour cela, on peut penser à utiliser pour chacun des points 3D de la reconstruction le résidu correspondant à son cas : l'erreur de reprojection proposée si le point est sur un mur et l'erreur de reprojection classique sinon. Ces différents résidus pourraient alors être minimisés simultanément puisque les deux types d'erreur se rapportent à une erreur de reprojection. Néanmoins, des expérimentations ont été réalisées et ont montré que les deux types de résidus ont des ordres de grandeur différents, ce qui amène à des reconstructions considérablement erronées. Cela est dû en particulier à la simplicité du modèle 3D utilisé : l'erreur de reprojection spécifique proposée, étant directement liée à ce modèle, est généralement plus importante qu'une erreur de reprojection classique. Il est alors nécessaire de réfléchir à une approche permettant de pondérer automatiquement les deux types de résidus pour égaliser leur poids dans l'optimisation.



(a) Reconstruction après l'ICP non-rigide (b) Reconstruction après l'ajustement de faisceaux

FIGURE 5.14 – **Exemple de manque de contrainte dans l'ajustement de faisceaux.** Dans le virage, les caméras n'observent aucun point sur le modèle : leur pose n'est donc pas contrainte dans l'ajustement de faisceaux, ce qui amène localement à une mauvaise reconstruction.

Nous pensons qu'il sera important de résoudre ces problèmes en vue de traiter un nombre conséquent de séquences dans des environnements complexes et variés. En ce sens, les différents points précédemment cités semblent être des perspectives directes de nos travaux (page 158).

CHAPITRE 6

Relocalisation et réalité augmentée

Dans ce chapitre, nous étudions la possibilité de relocaliser une caméra mobile au sein de l'environnement préalablement appris. Nous nous attarderons tout d'abord à montrer que la base d'amers créée à partir de notre méthode se prête bien à ce type d'application. Nous présenterons alors une application complète d'aide à la navigation s'appuyant à la fois sur le SLAM de Mouragnon et al. (2006) et sur la base d'amers décrivant l'environnement.

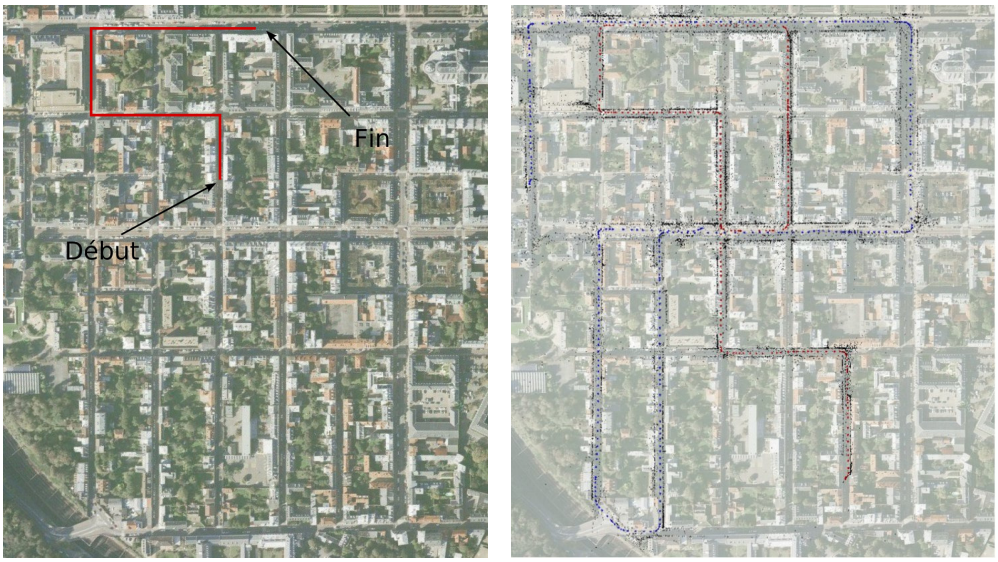
Les travaux décrits dans ce chapitre ont donné lieu à une publication (Gay-Bellile et al. (2010)).

6.1 Présentation de l'application visée

Le but de ce chapitre est de montrer que la base apprise avec la méthode proposée peut être utilisée avec succès pour des applications de relocalisation d'une caméra mobile. Nous souhaitons également montrer que la relocalisation est alors suffisamment précise pour des applications de réalité augmentée, en particulier dans le scénario d'aide à la navigation qui nous intéresse.

La base de données utilisée est le nuage de points reconstruit à l'aide de la méthode proposée dans cette partie. Chacun des points 3D de la base est associé aux descripteurs 2D, dans notre cas obtenus avec SURF (Bay et al. (2006)), de ses observations dans les images de la séquence d'apprentissage.

Dans la suite, nous tâcherons de localiser une caméra mobile dans cette base de données (figure 6.1(b)), c'est à dire de localiser chacune des images de la nouvelle séquence vidéo dans l'environnement préalablement appris. La caméra embarquée sur le véhicule est la même que celle utilisée pour l'apprentissage (section 5.2.1) et sa position sur le véhicule est identique. La trajectoire réelle suivie par le véhicule fait environ 500 mètres de long (figure 6.1(a)).



(a) (b)

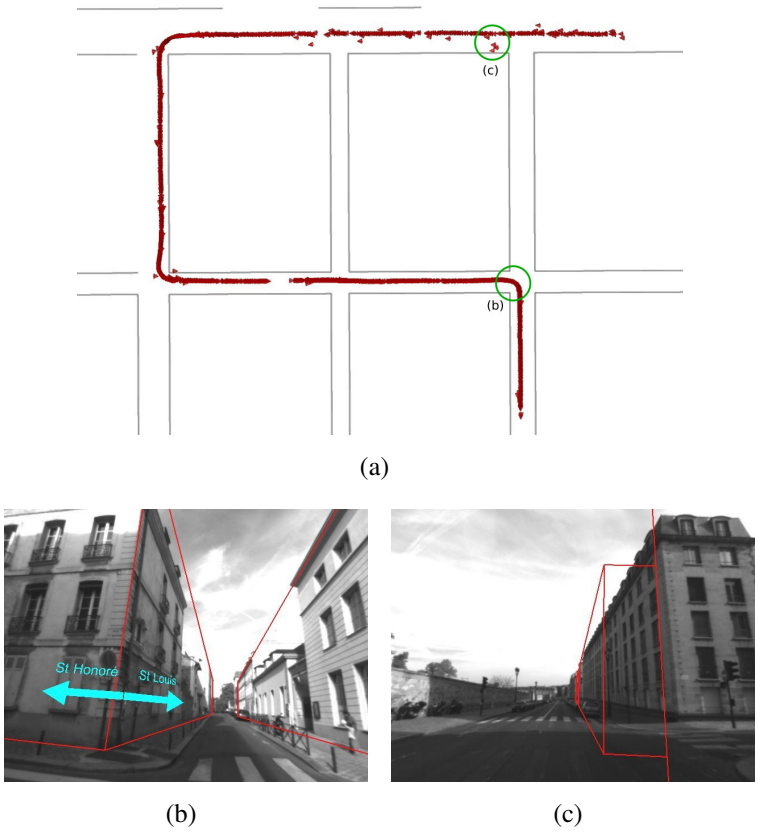
FIGURE 6.1 – **Données utilisées pour la relocalisation.** La relocalisation consiste à localiser l'ensemble des images de la nouvelle séquence vidéo (a) au sein de la base d'amers créée précédemment (b).

6.2 Localisation absolue dans la base d'amers

Dans cette section, chacune des images de la vidéo est localisée indépendamment des autres, c'est à dire qu'aucun filtrage temporel n'est utilisé entre les images successives. Le but de cette expérience est de montrer que la base créée permet très souvent de localiser avec précision une image sans information *a priori* sur sa localisation.

Ainsi, pour chaque image à localiser, nous cherchons tout d'abord à associer les points d'intérêt qui y sont détectés avec ceux de la base. Pour cela, nous cherchons pour chacun des points d'intérêt de l'image courante celui qui lui est le plus proche dans la base (en terme de distance associée au descripteur utilisé). Notons que la recherche du point d'intérêt le plus proche est exhaustive dans cette section. Cependant, afin de gagner en performance, plusieurs méthodes permettant de grouper en amont les descripteurs de la base par ressemblance photogrammétrique ont été proposées. On trouve parmi les approches les plus connues les *Randomized Trees* (Lepetit and Fua (2006)) et les *Vocabulary Trees* (Nister and Stewenius (2006)). Ces associations de points d'intérêt permettent de constituer des paires entre les points d'intérêt de l'image courante et les points 3D de la base. Ces associations 2D/3D permettent alors d'estimer la pose courante de la caméra (Haralick et al. (1994)), la robustesse de ce calcul étant assurée par l'utilisation d'un processus RANSAC (Fischler and Bolles (1981)).

Le résultat de la relocalisation de la caméra sur l'ensemble de la trajectoire est donné par la figure 6.2. Nous pouvons voir que globalement la trajectoire a été bien reconstruite, c'est à dire que la grande majorité des images ont été bien localisées. Néanmoins, certaines images ont un géoréférencement incorrect (figure 6.2(c)). Cette erreur de positionnement peut être due à différentes raisons. Généralement, cela est le résultat de mauvaises associations 2D/3D résultant de motifs répétitifs dans la structure de l'environnement par exemple. Un autre cas qui se produit souvent est un calcul de pose erroné à cause d'un mauvais conditionnement du problème, par exemple lorsque les points 3D utilisés sont coplanaires ou qu'ils sont mal répartis dans l'image.



(a) (b) (c)

FIGURE 6.2 – **Relocalisation d'une caméra dans un environnement connu.** Le fait de localiser toutes les images indépendamment (b-c) peut amener des erreurs de localisation importante (a).

Pour cette dernière situation, il est possible de calculer la covariance (c'est à dire l'incertitude) de la pose calculée (Royer et al. (2007)). Néanmoins, dans l'application de guidage qui nous intéresse, la pose se doit d'être précise à chaque image pour pouvoir superposer avec cohérence l'information voulue sur le flux vidéo. De plus, il apparaît nettement que localiser chacune des images indépendamment entraîne un *jittering* (c'est à dire un phénomène de tremblement) très important pour les éléments ajoutés en surimpression.

Afin de pallier ces problèmes, une méthode de relocalisation prenant en compte à la fois l'information temporelle et l'information géométrique a été proposée dans le laboratoire.

6.3 Vers une application d'aide à la navigation

Comme nous l'avons vu précédemment, afin de localiser une caméra mobile de façon précise, robuste et fluide, il est nécessaire d'intégrer dans le suivi deux types d'informations :

- ▷ une information temporelle, apportée dans cette étude par l'utilisation d'un algorithme de SLAM monoculaire, qui permettra à la fois de lisser la trajectoire reconstruite et d'assurer la cohérence entre les images successives
- ▷ une information géoréférencée, fournie par la base d'amers construite, qui évitera la dérive de la position de la caméra en fournissant une connaissance sur la position absolue de certains amers

6.3.1 Limites de l'existant

De nombreux travaux, combinant SLAM et information absolue, ont été récemment proposés. Par exemple, Castle and Murray (2009) s'appuient sur la méthode PTAM (*Parallel Tracking and Mapping*, Klein and Murray (2007)). PTAM est une méthode de SLAM utilisant un ajustement de faisceaux pour construire simultanément la trajectoire de la caméra ainsi qu'une carte de la zone parcourue. La principale différence avec la méthode de Mouragnon et al. (2006) est l'utilisation qui est faite de cette carte. En effet, dans la méthode de Mouragnon et al. (2006), l'association des points d'intérêt de l'image courante est faite uniquement avec les derniers points 3D reconstruits. Au contraire, faisant l'hypothèse que l'environnement parcouru est réduit, Klein and Murray (2007) proposent pour chaque nouvelle image d'y projeter l'ensemble de la carte déjà construite et de réaliser les associations entre les points d'intérêt de l'image courante et ces points projetés. Cela permet de toujours prendre en compte l'ensemble des informations de la carte et d'être donc moins sensible au problème de dérive. L'idée proposée par Castle and Murray (2009) est alors simplement de placer au sein de cette base des points connus et géoréférencés. L'ajustement de faisceaux prend alors en compte simultanément et de façon transparente l'information absolue apportée par ces points ainsi que l'information temporelle apportée par le reste de la carte. Notons que Vacchetti et al. (2004) proposent une approche similaire pour le suivi d'objet dont le modèle CAO est connu, l'information absolue étant dans ce cas des points du modèle dont la texture est connue préalablement.

Néanmoins, même si ces approches sont bien adaptées aux environnements réduits où l'information globale apportée peut être fournie très précisément, elles sont difficilement utilisables avec succès dans notre approche. En effet, il est nécessaire de prendre en compte le fait que, dans notre cas, la précision de la base d'amers est liée à celle des modèles 3D utilisés, précision qui reste limitée comme nous l'avons précisé à la section 2.6.2.3. Ainsi, on voit expérimentalement que les résidus liés aux points 3D de la base et ceux liés aux points 3D reconstruits en ligne par le SLAM n'ont pas le même ordre de grandeur, ce qui ne permet pas de les prendre en compte simultanément dans l'ajustement de faisceaux.

6.3.2 Approche proposée

Ne pouvant pas traiter les points géoréférencés et les points reconstruits en ligne simultanément, l'idée principale de la méthode que nous proposons (Gay-Bellile et al. (2010)) est d'utiliser chacune des données aux moments opportuns.

Le principe de la méthode développée est le suivant. Le flux vidéo courant (c'est à dire enregistré en ligne par la caméra mobile) est traité en temps-réel par la méthode de SLAM de 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 géoréférencées. Pour cela, un traitement spécifique est réalisé à chaque nouvelle image clé. Deux poses sont 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 visuels. Cette pose est supposée correcte et précise. Plusieurs filtres classiques peuvent être utilisés en pratique pour s'en assurer (répartition des points d'intérêt dans l'image, ratio d'inliers retenus, ...)

La transformation entre ces deux poses (figure 6.3(a)) définit complètement la dérive du SLAM : l'erreur de rotation, de translation ainsi que le facteur d'échelle (fourni par exemple par le rapport des distances entre les points reconstruits avec le SLAM et ces mêmes points dans la

base d'amers). 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 géoréférencées (figure 6.3(b)). Le facteur d'échelle est alors utilisé (figure 6.3(c)) pour corriger la norme du déplacement des 20 dernières caméras clés, c'est à dire celles utilisées dans l'ajustement de faisceaux local. Ainsi, le facteur d'échelle transmis à la suite de la reconstruction est correct.

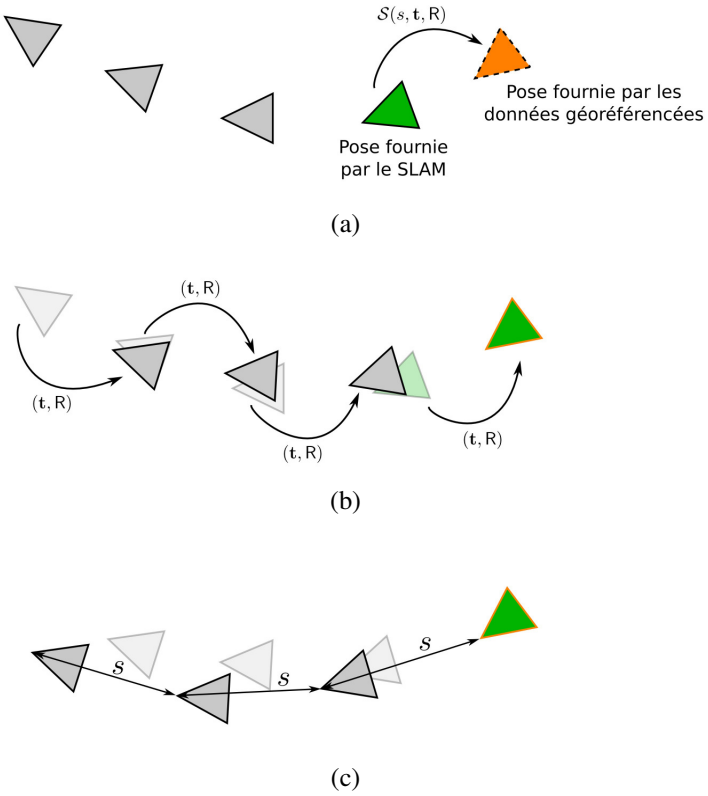


FIGURE 6.3 – **Couplage du SLAM avec des données géoréférencées.** La reconstruction SLAM est corrigée en utilisant les données géoréférencées (a) : une transformation euclidienne est appliquée pour corriger la pose de la caméra clé courante (b) et le facteur d'échelle est alors corrigé (c).

Le processus complet décrit dans cette section ne fonctionne que lorsque la pose calculée à partir des données géoréférencées est correcte. Or, nous avons vu précédemment (section 6.2) qu'il arrive 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 grâce à elle ; dans le cas contraire, aucune correction n'est appliquée et le processus de SLAM continue.

6.3.3 Résultats

La méthode proposée a été testée sur la séquence présentée à la section 6.1. La vitesse de traitement est environ de 40 images par seconde sur un PC de bureau classique. Nous pouvons voir sur la figure 6.4 que la trajectoire est globalement plus cohérente que lors de l'utilisation

unique des données géoréférencées (figure 6.2). Notons qu'ici, seules les images clés sont affichées puisque ce sont les seules qui sont remises en cause grâce aux données de la base d'amers visuels.

Nous pouvons également noter que la précision obtenue avec cette méthode permet d'ajouter avec une précision convenable des informations en surimpression. De plus, il apparaît très nettement que la vidéo résultante est beaucoup plus fluide et donc plus agréable pour l'utilisateur.

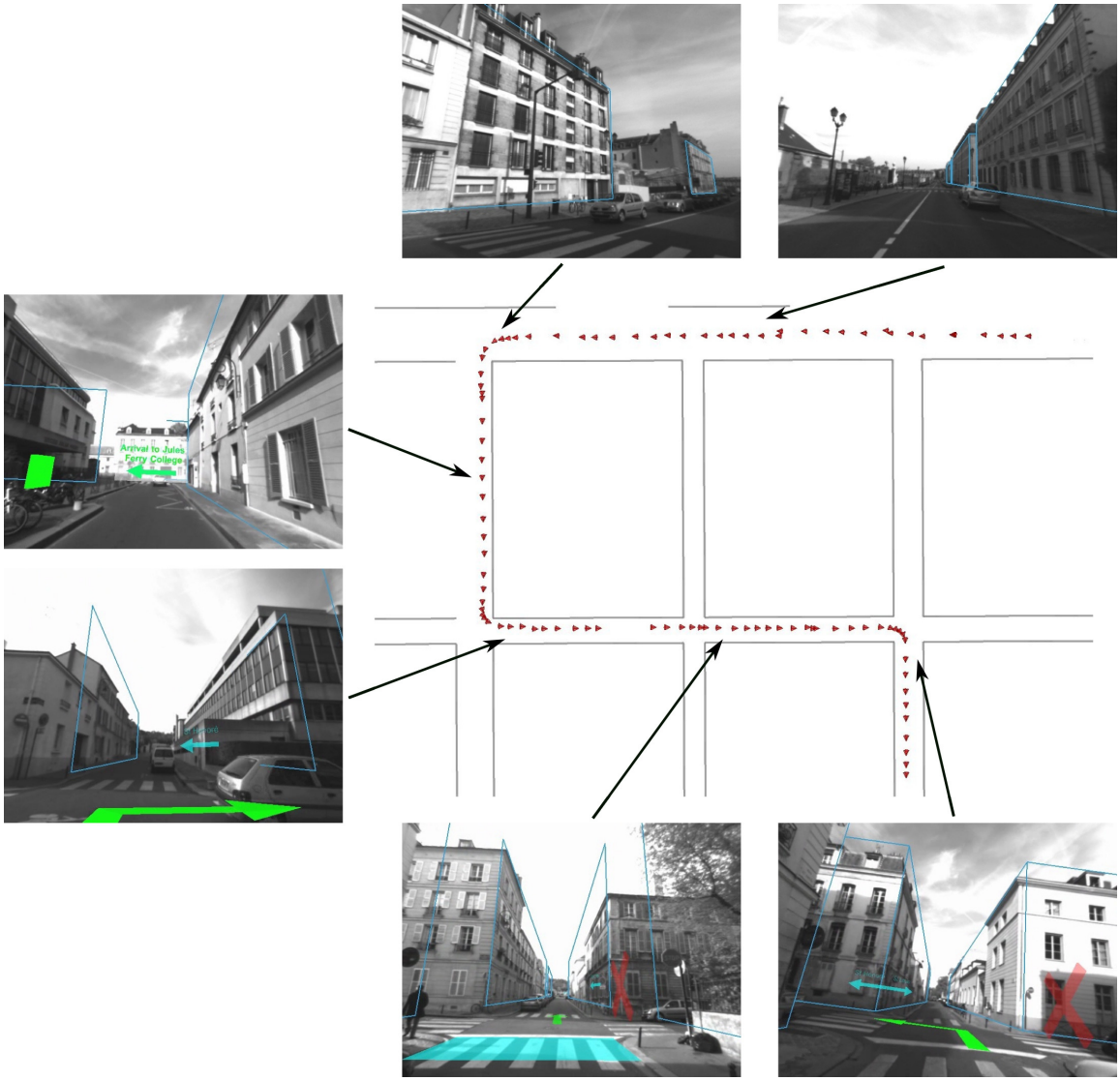


FIGURE 6.4 – **Relocalisation pour l'aide à la navigation.** La méthode proposée permet de relocaliser précisément une caméra mobile dans un environnement préalablement appris.

Dans le cadre des activités récentes du laboratoire, la méthode de relocalisation a été testée sur une autre séquence vidéo (en violet sur la figure 6.5, la trajectoire orange correspondant à la séquence présentée ci-avant). Cette séquence fait environ 650 mètres et a été tournée avec la même caméra que celle ayant servie à la construction de la base de données. Comme on peut le voir, la qualité visuelle de la restitution a également été améliorée.

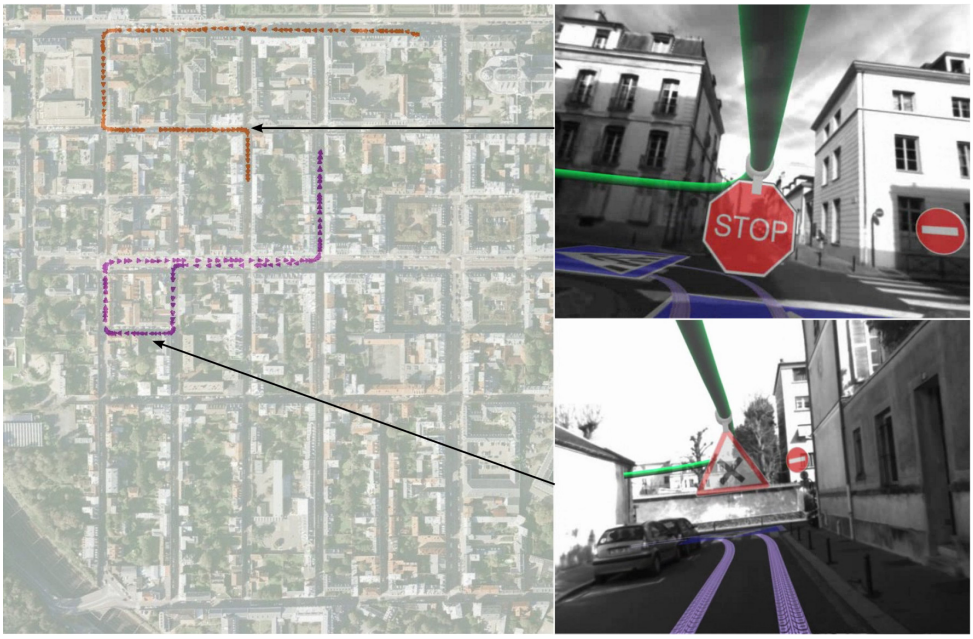


FIGURE 6.5 – **Résultats complémentaires en réalité augmentée.** La méthode de relocalisation proposée a été testée sur différentes séquences de plusieurs centaines de mètres.

6.4 Discussion

Au terme de cette partie, nous avons montré qu'il est possible, à partir d'un flux vidéo et d'un modèle 3D grossier, de construire une base d'amers d'un large environnement. Nous avons par la suite illustré que cette base d'amers peut être utilisée avec succès pour la relocalisation d'une caméra mobile dans l'environnement appris et que cette relocalisation est suffisamment précise pour être utilisée dans des applications de réalité augmentée.

Néanmoins, puisque la relocalisation s'appuie sur l'association de descripteurs de points d'intérêt, elle en partage les limites. En particulier, l'association des descripteurs est particulièrement sensible aux changements d'illumination et aux changements de points de vue. Or, ces conditions varient fortement dans le contexte extérieur qui nous intéresse. Pour assurer la robustesse de la relocalisation, plusieurs politiques sont dès lors envisageables. On peut par exemple penser à créer des bases de données dans différentes conditions. La base utilisée sera alors choisie en fonction de l'heure de la journée, des conditions climatiques, *etc.* On pourrait également penser à ce que les véhicules qui utilisent cette base puissent également la modifier : cela pourrait alors permettre qu'elle soit régulièrement mise à jour.

Dans la suite de ce mémoire, nous allons aborder une toute autre approche. En effet, afin d'éviter les problèmes liés aux descripteurs, nous allons proposer une méthode de localisation d'une caméra mobile sans apprentissage préalable. Afin de corriger le SLAM en ligne, cette méthode s'appuiera non pas sur des données photométriques mais uniquement sur la connaissance de la géométrie du SIG, donnée constante sur une grande échelle de temps.

Deuxième partie

Vers la correction en ligne d'une reconstruction SLAM

Contenu de la partie

Présentation de la méthode **99**

7 Méthode de correction du facteur d'échelle **103**
7.1 Etat de l'art 103
7.2 Contraintes disponibles et positionnement de nos travaux 105
7.3 Outils nécessaires à l'estimation du facteur d'échelle 106
7.4 Méthodes d'estimation du facteur d'échelle proposées 115
7.5 Validation expérimentale de l'estimation du facteur d'échelle 118
7.6 Intégration du facteur d'échelle dans la méthode SLAM 124
7.7 Résultats expérimentaux 129
7.8 Discussion 133

8 Méthode de correction de l'accumulation d'erreur **135**
8.1 Objectif de l'étude 135
8.2 Alignement et contraintes géométriques exploitables 136
8.3 Estimation de la dérive à l'aide d'un modèle 3D de ville 138
8.4 Estimation de la dérive à l'aide d'une carte de la route 144
8.5 Intégration de la nouvelle information 149
8.6 Résultats expérimentaux 149
8.7 Discussion 156

Présentation de la méthode

Dans cette deuxième partie, nous allons étudier la possibilité de localiser en temps-réel un véhicule à l'aide d'une unique caméra et d'un SIG simple de la scène. L'approche proposée s'appuie sur la méthode de Mouragnon et al. (2006) et consiste à lui apporter des informations supplémentaires permettant de corriger à la fois la dérive du facteur d'échelle et la dérive liée à l'accumulation d'erreur.

Une partie des travaux décrits dans cette partie ont donné lieu à une publication (Lothe et al. (2010c)).

Objectif détaillé de l'étude réalisée

Dans cette section, nous allons tout d'abord décrire la nouvelle problématique de notre étude. Nous expliquerons alors en quoi les outils précédemment proposés (partie I) ne peuvent être utilisés en l'état pour résoudre le nouveau problème posé.

Localisation en ligne d'un véhicule en milieu urbain

L'objectif de cette seconde partie est de pouvoir localiser en temps-réel un véhicule se déplaçant dans un milieu urbain. La méthode proposée dans la partie I permet d'obtenir une localisation fine de la caméra mais présente plusieurs inconvénients notables :

- ▷ **Sensibilité aux variations d'observation.** La méthode de relocalisation précédemment proposée se base sur la mise en correspondance d'amers visuels entre la base de données créée et l'observation courante de la caméra mobile. Ces appariements sont basés sur de l'information photométrique La méthode est par conséquent sensible aux variations d'illumination et de point de vue vis à vis de la séquence vidéo ayant servi à la construction de la base de données.

- ▷ **Apprentissage coûteux.** Cette approche nécessite la construction *a priori* de la base de données. Cette méthode, bien qu'automatique, deviendrait coûteuse en temps de traitement sur de très grands environnements. De plus, les données utiles dans cette base sont des données photométriques. Or, ce type de données n'est pas très stable dans le temps.

Ceci implique la nécessité de réenregistrer régulièrement des séquences vidéos afin de mettre à jour les bases de données.

Pour pallier ces différents problèmes, nous proposons dans cette partie d'exploiter uniquement un modèle géométrique de l'environnement. En effet, l'utilisation de la géométrie des bâtiments d'une ville présente plusieurs avantages :

- ▷ **Stabilité dans le temps.** L'information liée à la géométrie des bâtiments des villes peut être raisonnablement considérée comme étant stable dans le temps.

- ▷ **Invariance aux conditions d'observation.** Contrairement à l'information photométrique, l'information géométrique ne dépend pas du point de vue ni des conditions d'illumination.

- ▷ **Large distribution.** De tels modèles sont déjà largement distribués dans les applications de localisation sur internet et tendent à apparaître dans les systèmes de navigation. Ainsi, il n'est pas nécessaire de créer ces modèles pour pouvoir les utiliser. Il suffit au contraire d'exploiter des données qui sont pré-existantes dans de nombreux dispositifs.

Pour parvenir à localiser la caméra au sein de ce modèle 3D de ville, une méthode de SLAM classique est utilisée pour inférer en temps-réel la géométrie de l'environnement parcouru ainsi que la trajectoire du véhicule. Le processus de localisation proposé consiste alors à aligner la géométrie reconstruite par l'algorithme de SLAM et le SIG de la ville. On notera que l'information photométrique est ici uniquement utilisée pour reconstruire en ligne la géométrie de l'environnement parcouru. Par conséquent, l'apparence visuelle d'un amer est uniquement utilisée sur un court laps de temps et pour des points de vue proches les uns des autres. Dans ces conditions, l'information photométrique peut être considérée comme étant stable.

Limites de l'adaptation de la méthode précédente

Pour résoudre le nouveau problème fixé, il est naturel en premier lieu de penser à utiliser désormais en ligne les méthodes qui ont été proposées et utilisées précédemment hors ligne pour la construction de la base d'amers. En effet, dans les deux cas de figure le but est le même, c'est à dire aligner une reconstruction SLAM avec le SIG fourni. En particulier, il serait naturel de vouloir remplacer dans la méthode de SLAM de Mouragnon et al. (2006) l'ajustement de faisceaux classique par celui proposé dans le chapitre 4. Ceci permettrait en effet de prendre en compte le modèle 3D au fur et à mesure de la reconstruction (et donc de la localisation).

Néanmoins, une différence importante existe entre les approches hors ligne et en ligne. En effet, dans la partie I, l'alignement est réalisé *a posteriori*, sur l'ensemble de la reconstruction en même temps. Au contraire, si la correction est faite en ligne, nous disposons uniquement des informations récoltées par le SLAM jusqu'à l'instant courant. De plus, dans la méthode de Mouragnon et al. (2006), l'ajustement de faisceaux n'est réalisé que sur une fenêtre réduite de la reconstruction pour des raisons d'efficacité (section 2.6.1). En pratique, seuls les paramètres des 3 dernières caméras et des points 3D liés sont optimisés. Dans ce cas, la majorité des points optimisés ne sont pas nécessairement sur le modèle (par exemple lorsque le véhicule passe à côté d'une voiture, d'un arbre, *etc.*). Or, nous avons vu précédemment que l'ajustement de faisceaux proposé fournit de bons résultats lorsque la majorité des points reconstruits se situent effectivement sur le modèle 3D. Pour la construction de la base d'amers, l'ajustement de faisceaux

est global (c'est à dire que l'ensemble des caméras sont optimisées simultanément) si bien que cette hypothèse est très souvent globalement respectée. Expérimentalement, nous voyons alors qu'utiliser directement l'ajustement de faisceaux ST-CBA dans le SLAM a tendance à plaquer l'ensemble des points sur le modèle, ce qui amène très rapidement à une mauvaise reconstruction de la trajectoire.

Nous venons d'expliquer en quoi il est difficilement envisageable, de façon immédiate, d'intégrer la correction de la reconstruction SLAM à chaque image clé. Cette observation nous a donc amenés à proposer une approche alternative.

Approche proposée

L'idée principale de notre approche est d'exploiter des données simples sur la géométrie de l'environnement parcouru. Après avoir présenté ces données, nous montrerons quelles sont les contraintes qu'il est possible d'en extraire afin de corriger en ligne le processus de SLAM.

Données utilisées

Pour des raisons d'embarquabilité (développées à la section 7.1.1), nous ne souhaitons pas utiliser de capteur supplémentaire. Les données que nous exploitons dans cette partie sont donc uniquement des informations géométriques, facilement embarquables sur un véhicule :

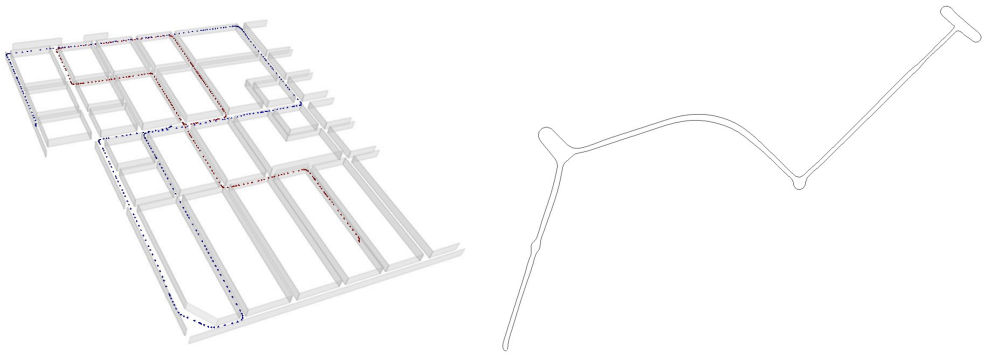
- ▷ **Equation du plan du sol.** Nous supposons que l'équation du plan du sol dans le repère caméra est connue. La caméra étant sur un véhicule terrestre, cette équation est par conséquent constante au cours du temps. Il suffit donc de réaliser, au préalable, une étape de calibrage externe entre la caméra et le plan de la route.

- ▷ **Système d'Information Géographique.** Nous considérons également que nous connaissons un Système d'Information Géographique lié à l'environnement parcouru. Parmi les données du SIG embarqué, deux types d'information seront utilisés dans notre approche. Tout d'abord, nous exploiterons les modèles de bâtiments 3D (figure 6.6(a)) tels que décrits dans la section 2.6.2.3. Nous montrerons de plus que lorsque ces modèles 3D ne sont pas disponibles (en particulier en dehors des villes), il est possible de les substituer avec succès par une carte simple de la route.

Exploitation de ces données

Pour rendre les méthodes SLAM monoculaires exploitables sur des séquences de grande échelle, il est nécessaire de corriger à la fois leur dérive en échelle et la dérive liée à l'accumulation des erreurs en position et en orientation. Dans les chapitres suivants, deux processus sont proposés afin de corriger respectivement ces deux types de dérive :

- ▷ **Correction du facteur d'échelle.** (chapitre 7) Le facteur d'échelle est corrigé en exploitant la connaissance de la position relative entre la caméra et le plan du sol. En effet, s'il est connu qu'il est possible d'extraire en particulier le facteur d'échelle à partir de l'homographie estimée du sol (section 2.4.2), nous montrerons que l'estimation du mouvement du véhicule fournie par le SLAM permet de résoudre ce problème de manière



(a) (b)

FIGURE 6.6 – **Données du Système d'Information Géographique 3D.** (a) est un exemple de modèle 3D. (b) est un exemple de carte, obtenue par sous-échantillonage d'un trajectomètre.

plus rapide et plus robuste que les approches habituelles. Nous présenterons alors également comment le facteur d'échelle ainsi estimé est utilisé pour corriger le processus de SLAM.

▷ **Correction de l'accumulation d'erreur.** (chapitre 8) La connaissance du SIG est exploitée afin de corriger l'accumulation d'erreur réalisée sur la position et l'orientation du véhicule. Cette correction est obtenue en alignant la reconstruction SLAM avec le SIG lorsque cela est possible.

Pour chacun de ces processus, nous présenterons tout d'abord les paramètres qu'il est possible d'estimer à partir de l'information supplémentaire utilisée. Nous présenterons alors comment sont estimés ces paramètres et la façon dont ils sont intégrés au processus SLAM.

Il est important de noter que ces deux processus sont indissociables. En effet, la correction du facteur d'échelle seule ne permet pas d'éviter la dérive liée à l'accumulation d'erreur. D'un autre côté, puisqu'il est basé sur une technique de recalage, le processus de correction de l'accumulation d'erreur nécessite une initialisation relativement proche de la solution. Cette condition ne peut être vérifiée que si le facteur d'échelle n'a pas trop dérivé.

CHAPITRE 7

Méthode de correction du facteur d'échelle

Dans ce chapitre, nous proposons une méthode robuste permettant la correction du facteur d'échelle (i.e. la norme du déplacement) au cours de la reconstruction SLAM. Nous allons tout d'abord montrer qu'en couplant le déplacement relatif estimé par le SLAM avec l'équation du plan du sol, il est possible d'exprimer le mouvement des points du sol sous la forme d'une homographie paramétrée par une seule inconnue : le facteur d'échelle. Une fois le facteur estimé, nous présenterons une méthode d'intégration de ce facteur d'échelle qui permettra ainsi de corriger en temps-réel la trajectoire reconstruite du véhicule.

7.1 Etat de l'art

Le problème de l'estimation du facteur d'échelle, qui est équivalent à l'estimation de la norme du déplacement de la caméra au cours du temps, est un problème majeur pour les méthodes de SLAM. Ainsi, de nombreuses approches ont déjà été proposées pour résoudre ce problème.

7.1.1 Utilisation d'un capteur tiers

Une première famille d'approches propose d'exploiter un capteur supplémentaire qui fournit directement ou indirectement la norme du déplacement. Par exemple, il est possible d'utiliser une centrale inertielle haut de gamme (Strelow and Singh (2004); Hol et al. (2007)) ou, plus naturellement dans le cadre du véhicule, un odomètre (Scaramuzza et al. (2009b); Kaess and Dellaert (2010)) ou un GPS (Agrawal and Konolige (2006); Ikeda et al. (2007)). Néanmoins, nous souhaitons que le dispositif proposé soit bas-coût (ce qui exclu l'utilisation d'une centrale inertielle haut de gamme), qu'il puisse être intégré aux véhicules déjà en circulation (ce qui exclu l'utilisation de l'odomètre) et qu'il soit robuste en centre ville, là où le système GPS est difficilement utilisable à cause du masquage des signaux par les bâtiments. C'est en ce sens que nous souhaitons éviter d'utiliser un capteur supplémentaire dans notre étude. Notons de plus qu'il est intéressant de chercher à tirer le maximum d'information du flux vidéo seul. En effet, cela permettra alors de rendre plus efficace la fusion éventuelle avec d'autres capteurs.

7.1.2 Utilisation d'une information tierce

Dans le cas où on se limite à l'utilisation du capteur caméra seul, de nombreuses méthodes ont été proposées pour calculer le mouvement et la norme du déplacement de la caméra mobile. Quelle que soit l'approche retenue, les méthodes reposent toutes sur une notion d'étalon. Cet étalon peut correspondre aux dimensions connues d'un élément de la scène directement observé ou à une distance relative au positionnement de la caméra montée sur le véhicule (cette mesure étant alors observée indirectement). Nous allons maintenant présenter les approches les plus utilisées dans le cadre du déplacement d'un véhicule, ces méthodes étant regroupées en fonction de l'information supplémentaire qu'elles utilisent.

7.1.2.1 Points 3D géoréférencés

Dans l'approche de Davison et al. (2007) par exemple, le facteur d'échelle est initialisé puis contraint grâce à l'observation d'une cible fournissant quatre points 3D dont la position dans le monde est connue. Le facteur d'échelle est donc observable à tout instant à travers la distance entre ces quatre points. Si cette approche est efficace dans les zones contrôlées (intérieur, milieu industriel, *etc.*), elle est difficilement utilisable dans le contexte qui nous intéresse puisqu'aucun point 3D géolocalisé avec précision n'est observable.

7.1.2.2 Stéréoscopie

La stéréoscopie consiste à utiliser non pas une seule caméra mais une paire de caméras rigidement liées (Nister et al. (2006); Lemaire et al. (2007); Comport et al. (2007)). Si le calibrage (*i.e.* le déplacement relatif ainsi que sa norme) entre ces deux capteurs est connu, la métrique est alors implicitement observée à tout instant. En particulier, une scène triangulée à partir d'une paire stéréo calibrée est reconstruite au bon facteur d'échelle. Néanmoins, comme nous l'avons évoqué précédemment, une des contraintes fixées dans nos travaux est de pouvoir localiser le véhicule à partir d'une unique caméra.

7.1.2.3 Homographie du plan de la route

Une approche classiquement utilisée pour calculer le déplacement d'un véhicule terrestre consiste à utiliser l'apparence du plan du sol (Simond and Rives (2004); Dumortier et al. (2006); Scaramuzza and Siegwart (2008)). En effet, la route étant considérée plane, les observations des points du sol dans les images successives suivent une homographie, cette homographie pouvant par exemple être estimée à l'aide de la méthode DLT (Hartley and Zisserman (2004)). Or, Faugeras (1993) a montré qu'il est possible d'extraire de la matrice d'homographie à la fois le déplacement relatif et l'équation du plan observé. Néanmoins, en vision monoculaire, il existe une ambiguïté entre la norme du déplacement de la caméra et la distance entre la caméra et le plan de la route. Dès lors, si la distance de la caméra à la route est connue, il est possible de déduire la norme du déplacement de la caméra. Cependant, cette approche présente deux problèmes majeurs. Tout d'abord, l'extraction des paramètres recherchés à partir de l'homographie est un processus numériquement instable. En particulier, une mauvaise répartition des points dans l'image peut fortement perturber les paramètres de mouvement estimés. De plus, dans cette approche, à la fois l'estimation du mouvement et celle du facteur d'échelle reposent sur le suivi de points d'intérêt sur le sol. Or, en pratique, le sol est peu texturé et peut être parfois

occulté (par exemple par la présence des voitures lorsque le trafic est dense). La robustesse de l'estimation du mouvement du véhicule n'est donc pas assurée tout au long de la séquence.

7.1.2.4 Mouvement non holonome

Dans une approche récente, Scaramuzza et al. (2009a) proposent d'exploiter le fait qu'un véhicule terrestre classique est non holonome. En particulier, cela implique que lorsqu'il tourne, le véhicule suit un arc de cercle théoriquement parfait ayant pour centre l'intersection des axes des roues avant et arrière. Scaramuzza et al. (2009a) montrent alors qu'en connaissant la distance entre la caméra et le centre du véhicule (*i.e.* le milieu de l'essieu arrière), il est possible de calculer le facteur d'échelle de la scène reconstruite durant les virages. Ce facteur n'étant calculé que dans les virages, il est nécessaire que la dérive soit négligeable dans les lignes droites. Dans leurs travaux, cette hypothèse est acceptable. En effet, la caméra utilisée est une caméra omnidirectionnelle et il a été montré que l'utilisation de ce type de capteur permet de réduire fortement la dérive observée dans les méthodes SLAM monoculaires (Tardif et al. (2008)). Cependant, cette dérive est importante avec une caméra perspective et il est donc nécessaire d'estimer plus régulièrement le facteur d'échelle dans notre cas.

Dans la suite, nous allons présenter une nouvelle approche permettant de conserver une estimation robuste du mouvement tout en permettant un calcul régulier du facteur d'échelle.

7.2 Contraintes disponibles et positionnement de nos travaux

Dans cette section, nous allons tout d'abord présenter les données que nous supposons connaître dans ce chapitre. Nous présenterons alors l'idée générale de l'approche retenue. Nous énumérerons enfin les différentes étapes nécessaires à la résolution de notre problème.

7.2.1 Données exploitées

La donnée dont nous disposons dans ce chapitre est la connaissance de la position de la caméra par rapport au sol. La connaissance de cette donnée dans notre cadre d'étude est réaliste. En effet, la caméra étant fixée sur un véhicule qui est lui-même lié au sol, la pose relative entre la caméra et le sol est constante au cours du temps. De plus, elle peut être aisément obtenue à l'aide d'une étape de calibrage préalable.

Dès lors, nous proposons d'utiliser cette information pour corriger le facteur d'échelle au cours de la reconstruction à partir de l'homographie du sol. Comme cela a été précisé dans la section 7.1.2.3, c'est en particulier la connaissance de la distance entre la route et la caméra qui nous permettra de retrouver l'échelle de la scène. Si on néglige pour l'instant les problèmes liés à l'occultation éventuelle du sol ou à son manque de texture, on peut noter que l'information liée à l'observation de la route est disponible pour chaque image de la séquence vidéo. Cependant, pour pouvoir estimer correctement et avec précision une homographie entre deux images, il est nécessaire que le déplacement effectué entre celles-ci ne soit pas trop faible. Au contraire, une trop grande variation dans les images peut empêcher d'estimer cette transformation. Dans le but de se placer dans des conditions optimales, nous n'estimerons donc l'homographie (et par conséquent le facteur d'échelle) qu'entre les images clés successives retenues par le processus de SLAM.

7.2.2 Approche proposée

Comme nous l'avons vu précédemment, nous souhaitons proposer une méthode permettant de calculer régulièrement (dans l'idéal à chaque nouvelle image clé) le facteur d'échelle de la reconstruction. Néanmoins, comme nous l'avons souligné, les données liées à l'observation du sol ne sont pas toujours disponibles en pratique. En effet, le sol est parfois peu texturé, il peut être occulté par exemple par d'autres véhicules, *etc.* L'estimation du facteur d'échelle ne sera donc pas forcément possible pour chaque image clé. Afin d'assurer la robustesse de la localisation du véhicule à tout instant, il est donc nécessaire que l'estimation du mouvement de la caméra ne dépende pas du succès de l'estimation du facteur d'échelle. Ainsi, deux différences majeures sont proposées dans ce chapitre pour pallier les limites des méthodes classiques d'odométrie visuelle basées sur l'observation du sol (par exemple Scaramuzza and Siegwart (2008)) :

- ▷ l'estimation du mouvement est réalisée avec la méthode de Mouragnon et al. (2006). On notera que le déplacement relatif fourni par cette méthode peut être considéré comme étant précis (à la norme près), en particulier puisqu'il est optimisé dans un ajustement de faisceaux.

- ▷ l'estimation du facteur d'échelle courant est essayée à chaque nouvelle image clé. Pour cela, la connaissance fournie par le SLAM du déplacement relatif entre les deux dernières caméras clés est utilisée de façon à contraindre le problème de recherche du facteur d'échelle. Ceci permettra en particulier d'améliorer la robustesse et donc la fréquence de cette estimation.

Le facteur d'échelle étant calculé en dehors de l'estimation du mouvement, il sera alors nécessaire de le réinjecter dans la méthode SLAM pour corriger effectivement au fur et à mesure la localisation de la caméra.

Après avoir décrit les différents outils mis en place pour répondre à notre problématique (section 7.3), nous présenterons et validerons la méthode proposée dans le but d'estimer le facteur d'échelle (sections 7.4 et 7.5). La méthode permettant d'intégrer l'information ainsi calculée sera alors présentée (section 7.6).

7.3 Outils nécessaires à l'estimation du facteur d'échelle

Dans cette section, nous identifierons tout d'abord les problèmes à résoudre pour permettre l'estimation du facteur d'échelle. Par la suite, les différents modules répondant à ces problématiques seront détaillés.

7.3.1 Aperçu de l'algorithme développé pour l'estimation du facteur d'échelle

Nous allons ici présenter les principes généraux des différents modules nécessaires à l'élaboration de la méthode proposée.

L'approche proposée repose sur quatre modules différents, dont trois d'entre eux sont des contributions de nos travaux (figure 7.1) :

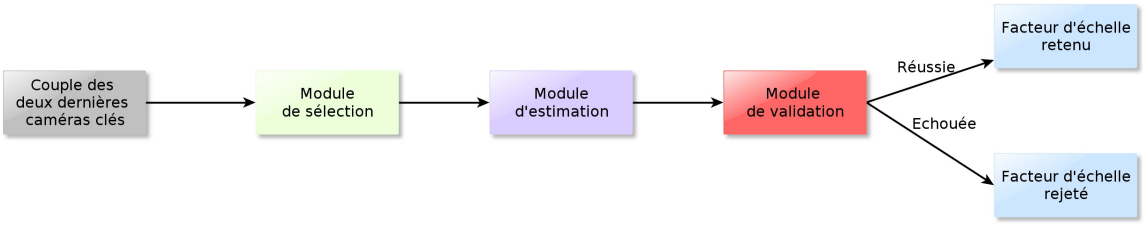


FIGURE 7.1 – **Modules nécessaires à l'estimation du facteur d'échelle.** Pour estimer le facteur d'échelle, trois problèmes sont à résoudre : la sélection des points sur la route, l'estimation du facteur d'échelle et la validation du facteur obtenu.

- ▷ **Appariement des points d'intérêt entre les images clés successives.** Il est tout d'abord nécessaire de détecter et d'apparier les points d'intérêt des deux dernières images clés. Néanmoins, cette étape est déjà réalisée par le processus SLAM et n'a donc pas à être effectuée à nouveau.

- ▷ **Sélection des points situés sur le sol.** L'estimation du facteur d'échelle se faisant à partir de l'homographie suivie par les points du sol, il est alors nécessaire de sélectionner parmi les points d'intérêt détectés et appariés ceux qui correspondent aux points 3D situés sur le sol dans la réalité.

- ▷ **Estimation robuste du facteur d'échelle.** Cette étape s'inspire des méthodes classiques qui consistent à estimer l'homographie suivie par les points du sol afin d'y extraire ensuite en particulier la norme du mouvement. L'originalité de notre méthode vient du fait qu'elle exploite le mouvement fourni par l'algorithme de SLAM pour contraindre le processus et ainsi le rendre plus robuste.

- ▷ **Validation du facteur d'échelle estimé.** L'étape de sélection des points situés sur le sol est une étape délicate, en particulier dans des conditions réelles de circulation. L'estimation du facteur d'échelle se faisant sur ces points, il est donc important de valider ce calcul *a posteriori*. Pour cela, deux critères de validation seront proposés. Notons dès à présent que cela accentue le fait que le facteur d'échelle ne sera pas disponible pour l'ensemble des couples d'images clés successives. Cependant, cela ne remet pas en cause la robustesse de la méthode d'estimation du mouvement puisque celle-ci est indépendante du calcul du facteur d'échelle. En pratique, lorsque le facteur d'échelle ne pourra pas être calculé, c'est celui propagé par la méthode SLAM qui sera conservé.

7.3.2 Module d'estimation du facteur d'échelle

Le premier module proposé consiste en une nouvelle formalisation du problème d'estimation du facteur d'échelle. Une méthode mathématique permettant de résoudre ce problème sera alors détaillée.

7.3.2.1 Le facteur d'échelle comme seul paramètre de l'homographie du sol

Pour estimer le facteur d'échelle, nous nous plaçons dans le contexte décrit par la figure 7.2 : les deux dernières caméras clés (notées $\mathcal{C}_1$ et $\mathcal{C}_2$) observent le plan du sol. L'équation du plan

du sol exprimée dans $\mathcal{C}_1$ (c'est à dire la position relative entre la caméra et sol) est supposée connue. Notons de plus qu'il est raisonnable de considérer que cette équation est la même pour toutes les caméras clés puisque le véhicule (et donc la caméra) est rigidement lié au sol.

Nous considérons dans cette partie que nous avons préalablement sélectionné un sous-ensemble de m points 3D $(\mathcal{Q}^i)_{1\leq i\leq m}$ (associés aux observations $(\mathbf{q}_1^i, \mathbf{q}_2^i)_{1\leq i\leq m}$) qui ont été filtrés comme étant sur le sol. Cette étape de sélection sera décrite à la section 7.3.3.

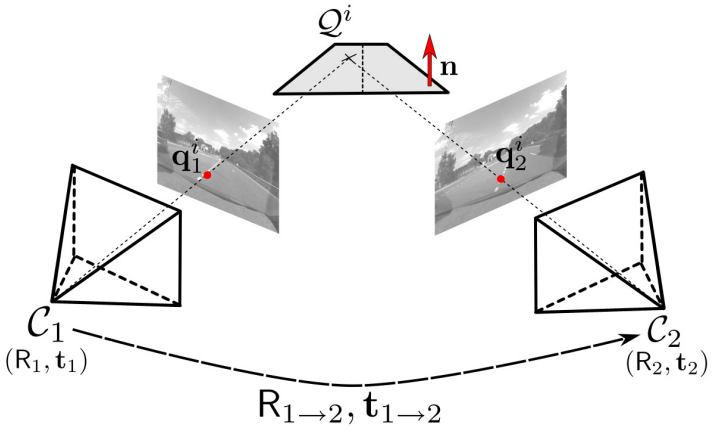


FIGURE 7.2 – **Contexte de la recherche du facteur d'échelle.** Pour estimer le facteur d'échelle, nous nous plaçons dans le cadre de deux caméras clés observant le même plan connu (le plan du sol).

Les points $(\mathcal{Q}^i)_{1\leq i\leq m}$ appartenant tous au même plan de l'espace, leurs couples d'observations sont reliés par une homographie $\mathcal{H}_{1\to2}$ (représentée matriciellement par $\mathsf{H}_{1\to2}$) :

$$\tilde{\mathbf{q}}_2^i \sim \mathsf{H}_{1\to2}\tilde{\mathbf{q}}_1^i \tag{7.1}$$

avec

$$\mathsf{H}_{1\to2} \sim \mathsf{K}(\mathsf{R}_{1\to2} - \frac{\mathbf{t}_{1\to2}\mathbf{n}^\mathsf{T}}{d})\mathsf{K}^{-1} \tag{7.2}$$

où K est la matrice de calibrage de la caméra, $(\mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$ le déplacement relatif de la caméra, $\mathbf{n}$ la normale du plan et d la distance caméra-plan (toutes deux exprimées dans le repère de la caméra $\mathcal{C}_1$).

Dans notre cas, le déplacement relatif entre les caméras a été préalablement estimé par le processus de SLAM. Les résultats des travaux de Mouragnon et al. (2006) ont montré que, au facteur d'échelle près, ce déplacement relatif est calculé avec précision. Ceci est directement lié au fait que la pose des N dernières caméras clés est optimisée par un ajustement de faisceaux local à chaque création d'une nouvelle image clé (voir le descriptif de la méthode à la section 2.6.1). Il est donc possible de considérer qu'en plus de l'équation du sol $(\mathbf{n}, d)$, les données $\mathsf{R}_{1\to2}$ et $\frac{\mathbf{t}_{1\to2}}{||\mathbf{t}_{1\to2}||}$ sont connues. Ainsi, dans notre contexte, l'homographie $\mathcal{H}$ est connue à un paramètre près qui est le facteur d'échelle λ :

$$\mathsf{H}_{1\to2}(\lambda) \sim \mathsf{K}(\mathsf{R}_{1\to2} - \lambda\frac{\mathbf{t}_{1\to2}\mathbf{n}^\mathsf{T}}{d})\mathsf{K}^{-1} \tag{7.3}$$

Nous pouvons déduire de cette équation que la recherche du facteur d'échelle peut être exprimée comme étant la recherche du λ qui minimise l'erreur de transfert liée à l'homographie

$\mathcal{H}_{1\to2}(\lambda)$. Cependant, optimiser uniquement la valeur de λ n'est pas optimale. En effet, les valeurs de $(\mathbf{n}, d, \mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$ n'étant pas parfaites, il serait nécessaire de les remettre en cause au cours de l'optimisation. Néanmoins, nous avons observé expérimentalement que le fait de relâcher ces paramètres induit des problèmes de convergence importants du fait par exemple de la mauvaise distribution des points dans l'image, d'une mauvaise sélection des points sur la route, *etc.* C'est pourquoi nous avons pris la décision de n'optimiser que la valeur de λ. Remarquons que Scaramuzza et al. (2009b) ont récemment tiré des conclusions similaires dans leur problème. En particulier, ils ont également été amenés à limiter les degrés de liberté du mouvement recherché du véhicule.

7.3.2.2 Résolution numérique du problème

Le problème étant désormais formalisé, nous allons étudier ici la méthode permettant d'estimer le facteur d'échelle à partir d'un ensemble de couples de points d'intérêt qui sont supposés être sur le sol.

Estimation linéaire. L'idée de la première étape de l'optimisation est de calculer une première estimation linéaire de λ qui sera alors utilisée comme initialisation pour l'optimisation non-linéaire.

L'équation 7.1 est équivalente à dire que les vecteurs $\tilde{\mathbf{q}}_2^i$ et $\mathsf{H}_{1\to2}\tilde{\mathbf{q}}_1^i$ sont colinéaires. En particulier, leur produit vectoriel est donc nul :

$$\tilde{\mathbf{q}}_2^i \times \mathsf{H}_{1\to2}\tilde{\mathbf{q}}_1^i = 0 \tag{7.4}$$

En développant cette relation pour l'ensemble des m points du sol, on peut en déduire la relation linéaire suivante :

$$\lambda\mathsf{A} = \mathsf{B} \tag{7.5}$$

où A et B sont deux matrices de dimensions $(3 \times m)$. On peut alors résoudre cette équation au sens des moindres carrés, à savoir :

$$\lambda = \mathsf{A}^{+}\mathsf{B} \tag{7.6}$$

où A^{+} est la matrice pseudo-inverse de A. Des données aberrantes pouvant se trouver parmi les données utilisées (mauvais appariements, *etc.*), cette estimation linéaire est rendue robuste par l'utilisation du consensus RANSAC (Fischler and Bolles (1981)).

Optimisation non-linéaire. La valeur obtenue pour λ peut alors être raffinée grâce à un processus classique de minimisation non-linéaire. En pratique, l'algorithme de Levenberg-Marquardt (Levenberg (1944)) est utilisé afin de minimiser l'erreur de transfert symétrique $\mathcal{E}$ associée à l'homographie $\mathcal{H}_{1\to2}(\lambda)$ pour l'ensemble des couples sélectionnés comme étant sur la route :

$$\mathcal{E}(\lambda) = \sum_i ||\mathbf{q}_2^i - \pi(\mathsf{H}(\lambda)\tilde{\mathbf{q}}_1^i)||^2 + ||\mathbf{q}_1^i - \pi(\mathsf{H}(\lambda)^{-1}\tilde{\mathbf{q}}_2^i)||^2 \tag{7.7}$$

où π est la fonction permettant de passer des notations homogènes aux coordonnées euclidiennes (voir la section 2.1). A l'instar de l'estimation linéaire, l'optimisation non-linéaire est rendue robuste par l'utilisation du M-estimateur de Tukey ρ_T dont le seuil est réglé automatiquement grâce au MAD. La fonction $\mathcal{F}$ effectivement minimisée est alors :

$$\mathcal{F}(\lambda) = \sum_i \rho_T(||\mathbf{q}_2^i - \pi(\mathsf{H}(\lambda)\tilde{\mathbf{q}}_1^i)|| + ||\mathbf{q}_1^i - \pi(\mathsf{H}(\lambda)^{-1}\tilde{\mathbf{q}}_2^i)||) \tag{7.8}$$

7.3.3 Modules de sélection des points d'intérêt situés sur le sol

Pour pouvoir calculer λ à partir de la méthode décrite ci-avant, il est nécessaire de pouvoir sélectionner parmi les points d'intérêt ceux qui correspondent aux points 3D étant sur le sol. Une approche classique (Adams et al. (2002); Scaramuzza and Siegwart (2008)) pour résoudre ce problème est de rechercher l'homographie principale existante entre les deux images clés, c'est à dire l'homographie qui relie le plus de points d'intérêt de ces deux images. Cette homographie est recherchée grâce au consensus RANSAC (Fischler and Bolles (1981)), le but de ce consensus étant de déterminer le plus grand sous-ensemble de points satisfaisant la même homographie. L'hypothèse faite alors est que le plan de l'image contenant le plus de points 3D observés est le plan du sol. Néanmoins, comme le montre la figure 7.3, cette hypothèse est souvent mise à mal dans un contexte urbain réel. En effet, de nombreux éléments hors du sol peuvent appartenir à un même plan de l'espace (coffre de voiture, ensemble de véhicules stationnés, façades des bâtiments, *etc.*). Dès lors, le plan le plus large observé dans les images n'est plus nécessairement le plan du sol.

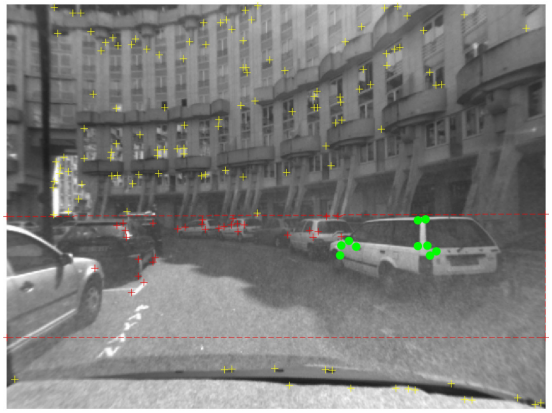


FIGURE 7.3 – **Recherche de l'homographie principale d'un couple d'images.** Dans un contexte urbain complexe, la recherche de l'homographie principale à l'aide du consensus RANSAC ne sélectionne pas toujours les points de la route. Les croix rouges sont les points d'intérêt de la zone route (*i.e.* sous l'horizon) et les ronds verts sont ceux retenus par le processus RANSAC.

Pour éviter ce problème, il a été proposé (Simond and Rives (2004); Dumortier et al. (2006)) de travailler sur une sous-zone de l'image correspondant à la route, par exemple en détectant *a priori* cette zone à partir des lignes de fuite ou des marquages au sol. Néanmoins, notons que ce prétraitement ne permet pas d'éviter les problèmes liés à la présence des véhicules situés sur la route, qu'ils soient stationnés ou en circulation.

Dans notre étude, nous limiterons la recherche des points d'intérêt situés sur le sol dans la zone de l'image située en dessous de l'horizon, celui-ci pouvant être calculé automatiquement dans notre cas puisque l'équation du plan du sol est connue dans le repère caméra. En pratique, pour définir *la zone route*, nous appliquons un décalage de 20 pixels en dessous de l'horizon de façon à filtrer les points d'intérêt détectés à l'infini, ces points pouvant perturber les différents calculs numériques réalisés dans cette partie.

Dans la suite, nous allons présenter deux nouvelles approches complémentaires visant à sélectionner parmi les points détectés dans les images ceux étant situés sur le sol.

7.3.3.1 Sélection globale

La première approche proposée est appelée sélection globale car elle n'utilise aucun *a priori* sur la norme du déplacement entre les deux caméras considérées. Ceci lui permet en particulier d'être robuste à la dérive du facteur d'échelle.

Approche proposée. L'idée générale de l'approche globale est que si un point est sur le sol, il est nécessairement le plus bas dans la scène reconstruite (ceci étant vrai dans un milieu urbain classique). Néanmoins, dans la pratique, cela n'est pas toujours respecté. En effet, les erreurs numériques, les objets mouvants dans la scène, *etc.* sont souvent à l'origine de la présence de points aberrants pouvant être reconstruits sous le sol. Dans le sens contraire, le point reconstruit le plus bas n'est pas nécessairement sur le sol. En effet, dans le cas où le sol est très peu texturé, les points reconstruits les plus bas correspondent par exemple souvent à des points situés sur les véhicules stationnés sur le bas-côté.

Pour chacun des points 3D reconstruits, il est donc nécessaire d'obtenir des critères permettant de fournir sa hauteur mais également sa probabilité d'être un point du sol. Pour cela, nous proposons de nous appuyer sur la paramétrisation de l'homographie du sol décrite à la section 7.3.2.1. En effet, l'homographie étant définie par le seul paramètre λ, il est possible d'obtenir pour chaque couple de points d'intérêt appariés $(\mathbf{q}_1^i, \mathbf{q}_2^i)$ deux informations complémentaires :

- ▷ sa *hauteur* par rapport aux autres points 3D. En effet, le facteur d'échelle λ_i calculé pour ce couple de points peut être traduit comme une information sur son altitude. Ainsi, le facteur d'échelle calculé peut être perçu comme la quantité de mouvement à appliquer entre les deux caméras afin de plaquer au sol le point 3D considéré. Dès lors, plus λ_i est petit, plus le point 3D est bas dans la scène.

- ▷ sa *probabilité* d'être sur le sol. En effet, l'estimation de λ_i est associé au résidu r_i. Ce résidu correspond à l'erreur de transfert symétrique associée à l'homographie $\mathcal{H}_{1\to2}(\lambda_i)$ calculée uniquement sur le couple $(\mathbf{q}_1^i, \mathbf{q}_2^i)$. Ce résidu peut être vu comme un critère de qualité mesurant si le couple de points d'intérêt respecte ou non le modèle de transformation 2D fixé par le plan du sol.

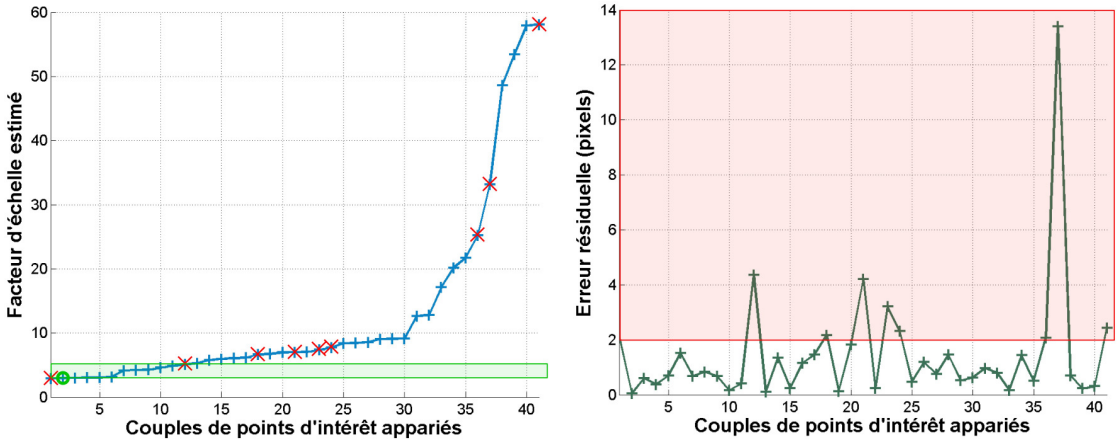
A partir de ces deux données, il est possible de définir un filtre qui sélectionne, parmi les couples de points d'intérêt, ceux qui ont la plus grande probabilité d'être sur le sol. L'idée de ce filtre est que pour être sur le sol, un point doit être le plus bas de la scène (*i.e.* associé au λ_i le plus faible) et suivre la contrainte imposée par le modèle de transformation du sol (*i.e.* avoir un résidu r_i peu élevé). L'idée est alors de considérer que le sol est associé au facteur λ_i le plus faible parmi ceux dont le résidu r_i est inférieur à un seuil ϵ donné. Dès lors, l'ensemble des points de l'image retenus comme étant sur le sol sont tous les points de faible résidu (*i.e.* avec $r_i < \epsilon$) dont le facteur d'échelle associé est proche de celui du sol (*i.e.* avec une faible variation par rapport au λ_i retenu).

Description de l'algorithme utilisé. En pratique, nous commençons donc par calculer les données (λ_i, r_i) pour l'ensemble des couples de points d'intérêt $(\mathbf{q}_1^i, \mathbf{q}_2^i)_i$. Ces données sont alors rangées dans l'ordre des λ croissants (figure 7.4). Parmi ces données, nous commençons par retirer les couples de points pour lesquels le résidu r_i est supérieur à ϵ (figure 7.4(b)). En pratique, nous fixons $\epsilon = 2$ pixels. Parmi les données restantes, nous notons $\lambda_{i_{min}}$ la valeur la

plus faible des facteurs d'échelle. Dès lors, l'ensemble des couples de points retenus $\mathcal{E}$ comme étant sur le sol sont les couples dont la valeur λ_i a une variation inférieure à γ % par rapport à $\lambda_{i_{min}}$ (figure 7.4(a)), c'est à dire :

$$\mathcal{E} = \left\{ (\mathbf{q}_1^i, \mathbf{q}_2^i) \;/\; 100 \times \frac{|\lambda_i - \lambda_{i_{min}}|}{\lambda_{i_{min}}} \leq \gamma \right\}_i \tag{7.9}$$

où γ est le seuil de variation à régler. Dans nos expériences, nous avons fixé $\gamma = 20\%$.



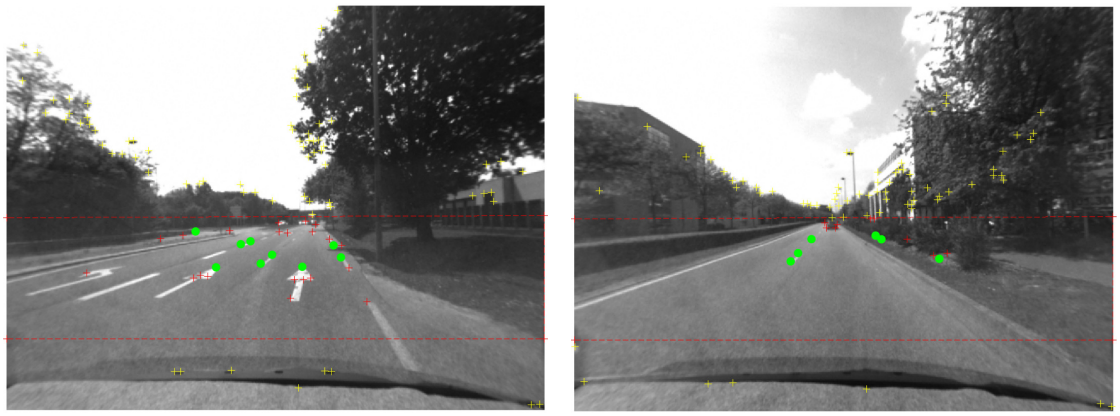
(a) (b)

FIGURE 7.4 – **Données utilisées pour la sélection globale des points du sol.** Pour chaque couple de points d'intérêt, on peut calculer (a) le facteur d'échelle associé et (b) l'erreur résiduelle obtenue pour ce facteur d'échelle sur ce couple de points d'intérêt. Les résidus filtrés comme aberrants sont dans l'encadré rouge. Les facteurs d'échelle correspondant aux points retenus comme étant sur le sol sont dans l'encadré vert.

Dès lors, le facteur d'échelle peut être calculé à partir de l'ensemble des points de $\mathcal{E}$ grâce à la méthode de résolution décrite à la section 7.3.2.

Discussion. Le plus grand intérêt de la sélection globale est qu'elle n'utilise aucun *a priori* sur la norme du déplacement inter-caméra. Cette particularité lui assure d'être robuste à la dérive du facteur d'échelle. Expérimentalement, comme le montre la figure 7.5, nous avons observé que cette méthode donne de bons résultats lorsque l'environnement parcouru est relativement simple (c'est à dire avec peu d'objets qui occultent la route).

Néanmoins, dans des environnements plus complexes, cette méthode échoue souvent : puisqu'aucun *a priori* sur λ n'est utilisé ici, la méthode doit s'appuyer sur des hypothèses fortes. En particulier, la sélection du facteur d'échelle le plus bas comme étant celui du sol et les seuils (ϵ, γ) à régler amènent généralement à éliminer certains points qui sont pourtant sur le sol dans la réalité. Cette méthode est donc qualifiée comme étant peu *sélective*. De plus, cette approche est sensible à l'incertitude liée aux données $(d, \mathbf{n}, \mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$ qui définissent l'homographie. En effet, lorsque ces données sont peu précises, le modèle d'homographie qu'elles définissent ne correspond pas exactement au modèle de mouvement des points du sol. L'homographie $\mathcal{H}_{1\to2}(\lambda)$ étant alors plus ou moins imprécise, les couples (λ_i, r_i) estimés peuvent être erronés. Rappelons de plus que dans cette partie, nous faisons l'hypothèse que le point associé au λ_i le plus faible est sur la route. Néanmoins, comme nous l'avons souligné, la seule information que nous avons réellement est qu'il s'agit du point 3D reconstruit le plus bas, mais rien



(a) (b)

FIGURE 7.5 – **Résultats obtenus avec la sélection globale.** Les croix rouges sont les points d'intérêt détectés dans la zone route et les ronds verts sont ceux retenus par la sélection globale.

n'assure qu'il appartienne à la route. Ces limites nécessiteront de valider *a posteriori* le facteur calculé, ce point étant étudié à la section 7.3.4.

7.3.3.2 Sélection locale

Dans cette section, nous allons présenter une approche dite locale qui utilise la dernière estimation du facteur d'échelle pour sélectionner avec robustesse les points situés sur le sol.

Description. La principale idée de la sélection locale consiste à utiliser d'une part que la hauteur entre la caméra et le sol est connue et d'autre part que localement (c'est à dire sur moins de 5 caméras clés) la norme du déplacement fournie par le SLAM ne dérive que faiblement. Ainsi, si le facteur d'échelle a pu être calculé récemment, la norme du déplacement courant issue du SLAM remise à l'échelle grâce à cette dernière estimation de λ fournit une bonne approximation de la norme réelle du déplacement courant.

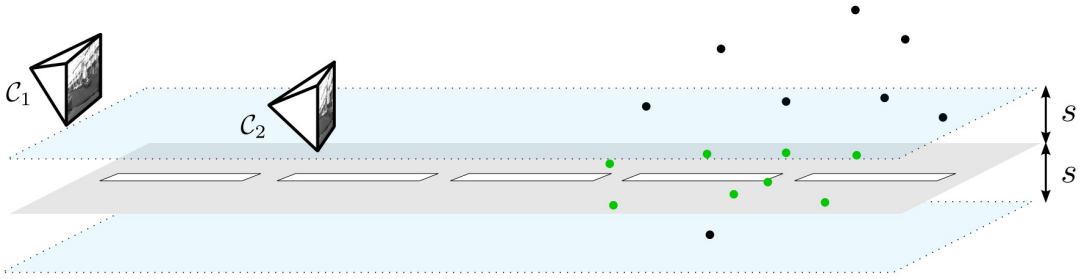


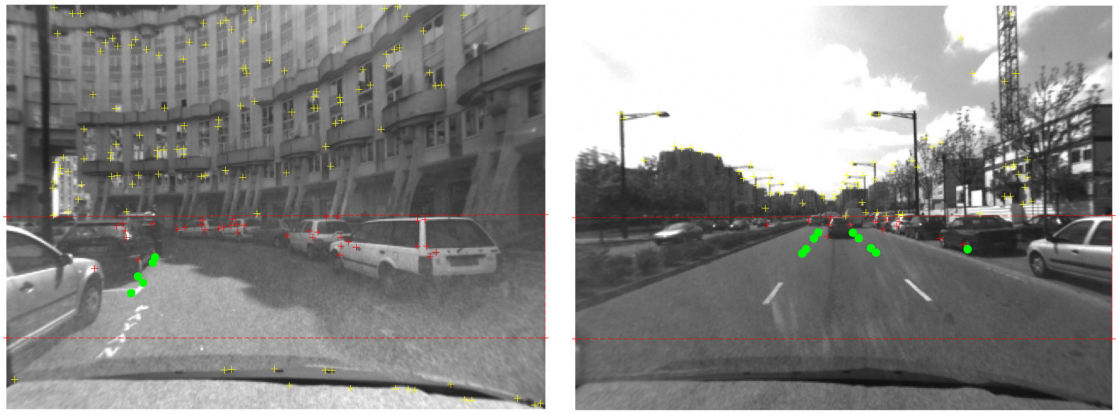
FIGURE 7.6 – **Méthode de sélection locale des points du sol.** En utilisant la dernière estimation de λ, il est localement possible de filtrer les points 3D en fonction de leur distance au sol.

A partir de cette information, et puisque l'équation du plan dans la caméra est connue, il est possible de filtrer les points 3D reconstruits par rapport à leur position 3D (obtenue par triangulation dans le processus de SLAM). En pratique (figure 7.6), la norme du déplacement estimée n'étant pas parfaite, nous sélectionnons tous les points situés à une distance du sol

inférieure à un seuil s (15 centimètres dans nos expériences). Cela permet de prendre en compte la variation possible de λ depuis sa dernière estimation.

Discussion. L'utilisation du calibrage entre la caméra et le plan de la route rend la méthode de sélection locale très sélective, en particulier dans les environnements complexes (figure 7.7). En particulier, la comparaison des figures 7.7(a) et 7.3 (page 110) met en avant la robustesse de la méthode proposée dans un environnement réaliste.

Néanmoins, la principale limite de cette méthode est son manque de robustesse face à la dérive du SLAM. En effet, le filtrage des points nécessite une bonne approximation *a priori* de λ, ce qui n'est plus le cas dès lors que son estimation n'a pu être réalisée pendant un nombre important de caméras clés (par exemple dans le cas d'une zone de trafic important ou d'une large zone avec un sol trop peu texturé).



(a) (b)

FIGURE 7.7 – **Exemples de résultats obtenus avec la sélection locale.** Les croix rouges sont les points d'intérêt détectés dans la zone route et les ronds verts sont ceux retenus par la méthode locale.

Nous allons maintenant présenter comment il est possible de valider le facteur d'échelle estimé à partir des points du sol sélectionnés.

7.3.4 Modules de validation du facteur estimé

Comme nous venons de le voir, la sélection des points du sol parmi tous les points reconstruits est une étape difficile, en particulier lorsque la scène parcourue est complexe. Le facteur d'échelle étant calculé à partir de ces points, il est donc nécessaire de qualifier *a posteriori* la confiance associée à son estimation. En ce sens, nous allons maintenant présenter les modules permettant de valider le facteur d'échelle estimé. Deux critères sont proposés dans la suite : le critère souple visant à vérifier que la résolution numérique s'est bien déroulée et le critère strict qui a pour but de vérifier le facteur d'échelle calculé lorsque celui-ci a été estimé à partir de données sur lesquelles nous n'avons pas entière confiance.

7.3.4.1 Critère souple

Le critère souple est composé de deux mesures dont les buts sont complémentaires. La première mesure vise à vérifier que la méthode d'estimation s'est effectuée dans un contexte per-

mettant une estimation robuste. En pratique, il est vérifié qu'au moins 4 couples de points d'intérêt ont été retenus comme non-aberrants (*i.e.* avec une erreur résiduelle inférieure au MAD) durant l'optimisation non-linéaire de λ (section 7.3.2.2). La deuxième mesure quant à elle vérifie que le λ obtenu est bien en accord avec les données utilisées. Pour cela, il est nécessaire que le RMS obtenu sur les points non-aberrants soit inférieur à 2 pixels.

7.3.4.2 Critère strict

Comme son nom l'indique, le but du critère strict est de proposer des contraintes fortes sur le λ estimé. Cela est utile en particulier lorsque la confiance qu'on peut avoir dans les points 3D utilisés pour son estimation est faible. L'idée principale de la validation stricte est de faire l'hypothèse que parmi tous les points d'intérêt pouvant être détectés dans la zone route (c'est à dire sous l'horizon), au moins 20% sont réellement situés sur la route. Le critère strict revient alors à vérifier qu'au moins 20% des couples de points d'intérêt de la zone route sont en accord avec le λ estimé. Pour cela, la validation stricte est composée de quatres étapes successives :

- ▷ **Détection de nouveaux points d'intérêt.** Dans chacune des deux dernières images clés, nous lançons une détection dense (*i.e.* avec un seuil lâche) de points d'intérêt de Harris (figure 7.8(a)). Notons qu'aucun descripteur n'est ici calculé, ce qui permet d'assurer un faible temps de traitement.

- ▷ **Transfert des points d'intérêt de l'image 1 dans l'image 2.** Tous les points détectés dans la première image clé sont transférés dans la deuxième image clé en utilisant l'homographie $\mathcal{H}_{1\to 2}(\lambda)$ associée à l'estimation courante du facteur d'échelle (figure 7.8(a)).

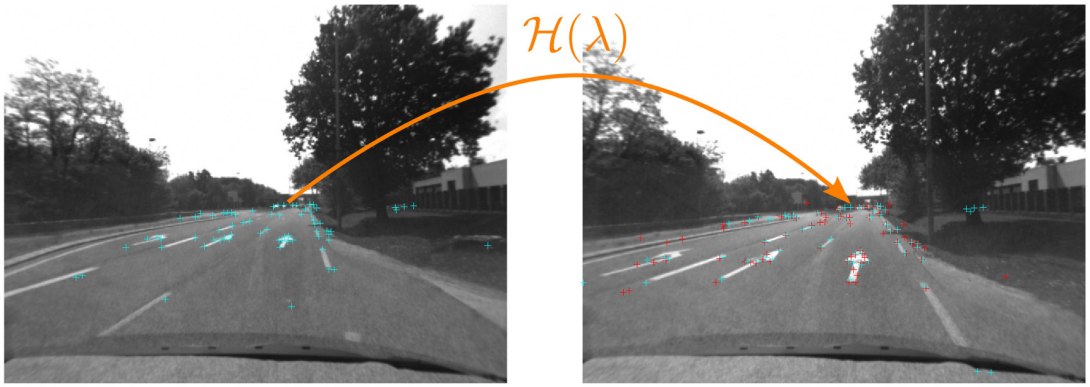
- ▷ **Association au plus proche.** Chacun des points d'intérêt détectés dans la deuxième image est associé à un point transféré depuis la première image (figure 7.8(b)). Le facteur d'échelle estimé étant supposé correct, cette association est réalisée au plus proche (en terme de distance euclidienne). En particulier, l'association au plus proche permet de gagner en temps de traitement par rapport à une association basée sur les descripteurs.

- ▷ **Mesure de la qualité.** Nous vérifions alors qu'au moins 20% des couples ainsi formés présentent une distance inférieure à 2 pixels. Si c'est le cas, le facteur d'échelle est conservé. Dans le cas contraire, il est rejeté.

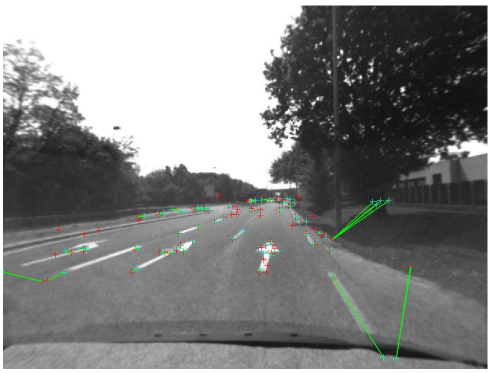
Dans la section suivante, nous allons présenter la façon dont les différents modules détaillés ci-avant sont agencés de façon à obtenir une méthode robuste d'estimation du facteur d'échelle.

7.4 Méthodes d'estimation du facteur d'échelle proposées

Dans cette section seront présentées différentes méthodes permettant d'estimer le facteur d'échelle. Ces différentes méthodes se différencient par la façon dont les modules (sélection, estimation et validation) sont reliés entre eux. Après avoir décrit des méthodes s'appuyant respectivement sur la sélection locale et globale, la méthode hybride retenue sera détaillée.



(a)



(b)

FIGURE 7.8 – **Aperçu de la validation stricte.** (a) représente la détection dense de points d'intérêt dans la zone route dans les deux images clés et le transfert des points de la première image dans la deuxième. (b) illustre la mise en correspondance de ces points au plus proche.

7.4.1 Méthode d'estimation globale

La première méthode est appelée *méthode globale* dans le sens où elle s'appuie sur la sélection globale des points sur le sol (figure 7.9). Après avoir sélectionné les points par la méthode de sélection globale (section 7.3.3.1), le facteur est estimé par résolution linéaire puis non-linéaire (section 7.3.2.2). Le facteur ainsi estimé est alors validé par le critère souple (section 7.3.4.1), de façon à s'assurer que la résolution du problème s'est effectuée dans de bonnes conditions. Si la validation par le critère souple réussit, une validation par critère strict est finalement lancée. En effet, comme indiqué dans la section 7.3.3.1, la sélection globale est peu sélective et la confiance qu'on peut avoir dans les données sélectionnées est par conséquent faible.

La méthode globale s'appuyant sur la méthode de sélection globale, elle en partage en conséquence les avantages et inconvénients. Ainsi, cette méthode est robuste à la dérive du facteur d'échelle. Néanmoins, de part son manque de sélectivité, l'estimation du facteur ne peut pas être réalisée très régulièrement.

7.4.2 Méthode d'estimation locale

Dans la *méthode d'estimation locale* (figure 7.10), les points du sol sont tout d'abord sélectionnés grâce à la méthode de sélection locale (section 7.3.3.2). Ces points sont alors exploités

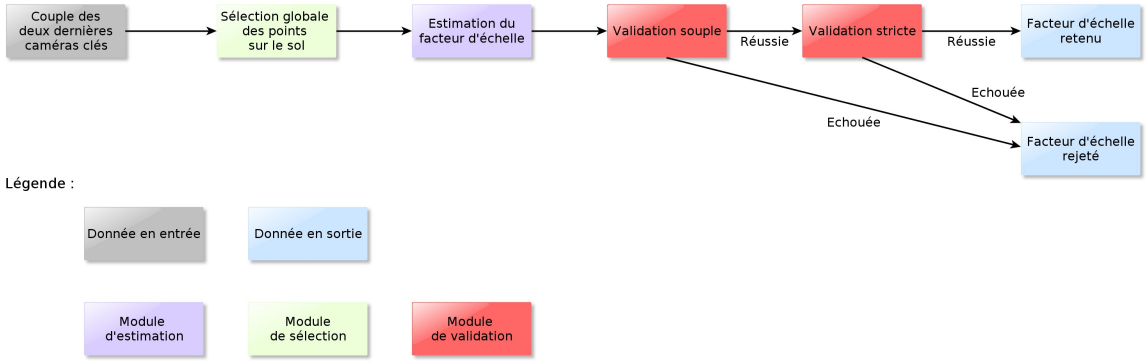


FIGURE 7.9 – **Description de la méthode globale.**

pour estimer la valeur du facteur d'échelle (section 7.3.2.2). Le facteur ainsi estimé est alors contrôlé par le critère de validation souple (section 7.3.4.1). Dès lors, si le facteur d'échelle courant a une évolution importante par rapport au dernier facteur d'échelle estimé (en pratique plus de 20%), il est nécessaire de valider le facteur obtenu par le critère strict. En effet, la méthode de sélection locale s'appuie sur l'hypothèse que le facteur d'échelle a peu évolué depuis sa dernière estimation. Si le résultat obtenu est non-conforme à cette hypothèse, il est donc préférable de vérifier que la valeur obtenue soit correcte.

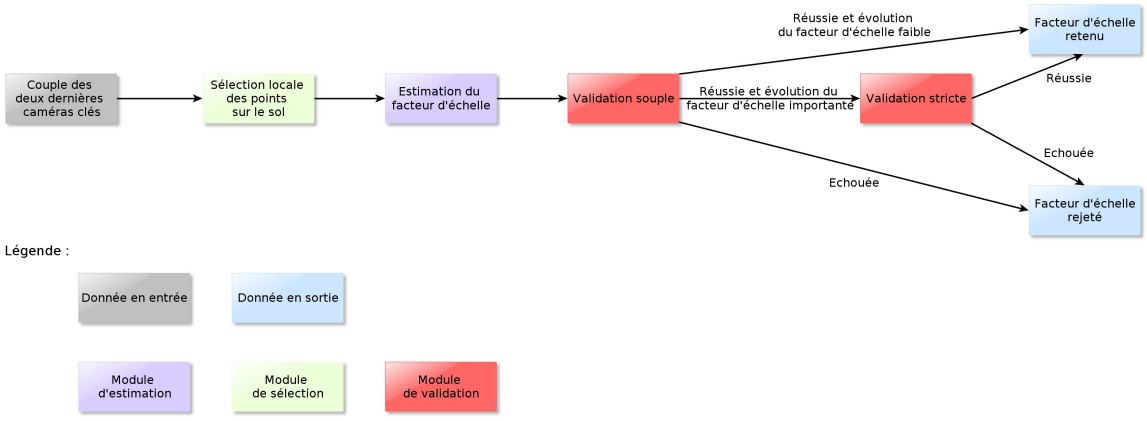


FIGURE 7.10 – **Description de la méthode locale.**

Tout comme le module de sélection locale sur lequel elle se base, cette méthode a l'avantage de souvent sélectionner l'ensemble des points 3D situés sur le sol. Cependant, elle reste très sensible à la dérive du facteur d'échelle.

7.4.3 Méthode retenue : une approche hybride

Nous l'avons vu précédemment, les méthodes d'estimation locale et globale sont complémentaires : la méthode locale est très sélective mais peu robuste à la dérive du facteur d'échelle alors que la méthode globale est peu sélective mais est nullement influencée par la dérive du facteur d'échelle.

La méthode finale retenue est une méthode hybride visant à prendre en compte les avantages de chacune des méthodes tout en limitant leurs inconvénients. Ainsi, comme le montre

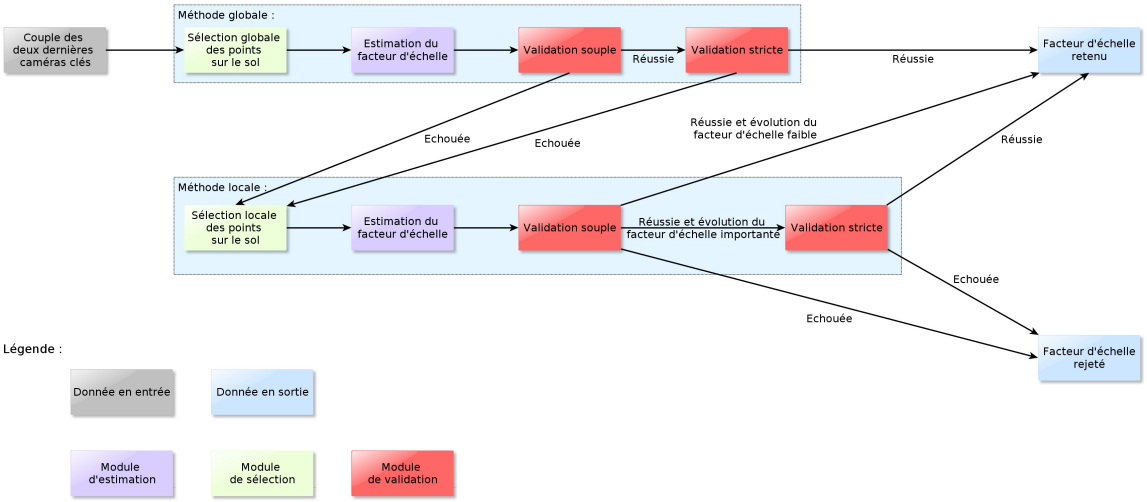


FIGURE 7.11 – **Résumé de la méthode d'estimation du facteur d'échelle hybride.** La méthode proposée comporte trois nouveaux modules : la détection des points sur la route, l'estimation du facteur d'échelle et la validation du facteur obtenu.

la figure 7.11, à chaque nouvelle caméra clé, nous testons tout d'abord la méthode globale. En effet, cette méthode ne demandant aucun *a priori* sur le facteur d'échelle, elle permet de pallier une éventuelle mauvaise estimation antérieure. Si la méthode globale échoue (*i.e.* qu'un des critères de validation n'est pas respecté), la méthode locale est alors essayée. Dès lors, si une des deux méthodes réussit, le facteur d'échelle estimé est conservé et utilisé pour corriger la norme du déplacement courant du SLAM (section 7.6). Dans la cas contraire, le facteur d'échelle est rejeté et la norme fournie par le SLAM n'est pas remise en cause.

7.5 Validation expérimentale de l'estimation du facteur d'échelle

Dans cette section, nous proposons une validation expérimentale de la méthode d'estimation du facteur d'échelle présentée ci-avant. Ainsi, après avoir décrit le protocole expérimental utilisé, nous détaillerons les résultats obtenus, en particulier en comparant la méthode de sélection hybride aux méthodes locale et globale seules. Nous discuterons alors des limites de la méthode et présenterons des perspectives permettant de les pallier.

7.5.1 Protocole expérimental

Après avoir détaillé la séquence vidéo sur laquelle sera testée la méthode d'estimation du facteur d'échelle proposée, les résultats de la méthode SLAM classique seront présentés.

7.5.1.1 Données utilisées

Dans cette section, nous travaillons sur une nouvelle séquence vidéo obtenue grâce au projet ODIAAC (ANR 06-PDIT-016-01). Les informations de cette séquence sont regroupées dans

les figures 7.12 et 7.13. La vidéo, tournée dans Saint-Quentin-en-Yvelines (France), retranscrit le parcours d'un véhicule sur une trajectoire d'environ 4,5 kilomètres dans des conditions réelles de circulation. La caméra embarquée sur le véhicule est une caméra perspective MARLIN F-033B qui fournit des images 640×480 à raison de 30 images par seconde.

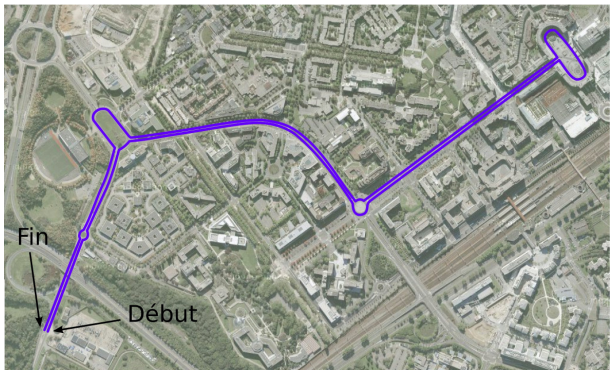


FIGURE 7.12 – **Trajectoire de la séquence ODIAAC.** Cette séquence est un parcours de 4,5 kilomètres dans Saint-Quentin-en-Yvelines (France).



(a) (b) (c)

FIGURE 7.13 – **Extraits de la séquence ODIAAC.** La séquence ODIAAC est une vidéo 640×480 enregistrée avec une caméra perspective simple.

L'intérêt majeur de cette séquence vidéo est qu'elle a été synchronisée avec des données provenant d'un trajectomètre (IXSEA LANDINS). Les données de ce capteur sont précises pour la majorité du parcours (inférieures au mètre en absolu), même si des erreurs de l'ordre de 2 mètres sont parfois observées. Nous nous servirons donc de ce capteur pour établir la vérité terrain de la position de chacune des caméras clés reconstruites (figure 7.14).

Dans notre algorithme, nous faisons l'hypothèse que l'équation du sol (à savoir la normale $\mathbf{n}$ et la distance caméra-sol d) est connue dans le repère de la caméra. En pratique, ne disposant pas de cette information, nous avons dû l'estimer grossièrement. Nous avons donc fixé une valeur grossière pour la distance d à partir des données constructeurs du véhicule. Pour sa part, la normale n a été calculée à l'aide d'une image de la séquence présentant un passage piéton. En effet, une fois les points d'intérêt correspondant aux coins du passage piéton triangulés, il est possible d'estimer de façon robuste la valeur de la normale.

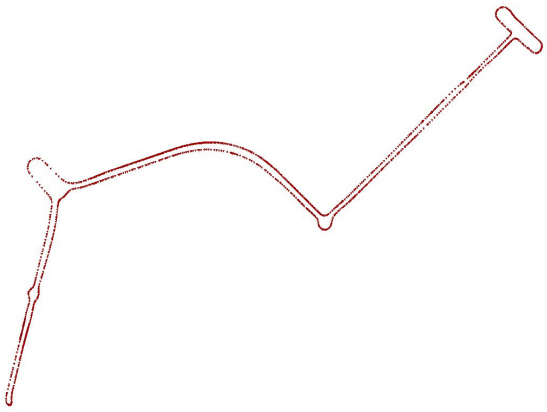


FIGURE 7.14 – **Vérité terrain de la séquence ODIAAC.** Le trajectomètre embarqué est utilisé pour créer la vérité terrain en position de chacune des caméras clés.

7.5.1.2 Reconstruction SLAM originale

La reconstruction SLAM obtenue sur cette séquence avec la méthode originale de Mouragnon et al. (2006) est décrite dans le tableau 7.1 et la figure 7.15. La reconstruction finale contient 1296 caméras clés et 39304 points 3D reconstruits. Cela revient à dire qu'en moyenne une image clé est créée tous les 3,5 mètres.

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
ODIAAC	4500	1296	39304

TABLE 7.1 – **Statistiques sur la reconstruction de la séquence ODIAAC.**

De plus, il est important de noter que la dérive du facteur d'échelle est particulièrement importante sur cette séquence. En effet, on aperçoit sur la figure 7.15 qu'après la place ovale (située en haut à droite de la trajectoire), le facteur d'échelle se réduit considérablement, ce qui rend la reconstruction globalement incohérente.

7.5.2 Résultats obtenus

Dans un premier temps, nous allons ici présenter les résultats obtenus sur l'estimation du facteur d'échelle pour la séquence ODIAAC. Nous nous placerons ensuite dans un scénario spécifique qui permettra de mettre en avant l'intérêt de la méthode hybride.

7.5.2.1 Calcul du facteur d'échelle

Afin de quantifier nos résultats, nous allons comparer les valeurs obtenues sur l'estimation de la norme du déplacement inter-caméra clé avec celles fournies par la vérité terrain. Nous allons en particulier comparer la méthode finale proposée (*i.e.* la méthode hybride) avec les méthodes locale et globale seules. Notons que pour pouvoir fonctionner, la méthode locale seule nécessite une initialisation correcte du facteur d'échelle pour les deux premières caméras clés. Dès lors, les trois méthodes seront testées avec la même initialisation du facteur d'échelle dans le but de pouvoir comparer leurs résultats.

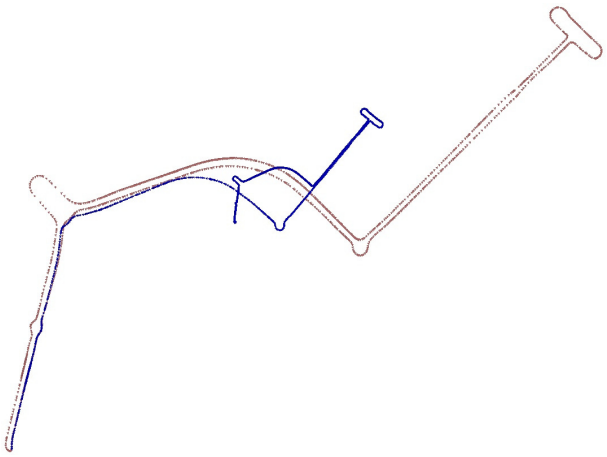


FIGURE 7.15 – **Reconstruction SLAM originale obtenue.** La superposition de la reconstruction SLAM originale (en bleu) et la vérité terrain (en rouge) met en avant que la dérive de la méthode de Mouragnon et al. (2006) peut être très importante sur de grandes distances.

Il est important de rappeler que, de part les étapes de validation *a posteriori* de chacune des méthodes, le facteur d'échelle n'est pas calculé pour tous les couples d'images clés successives. Toutes les statistiques présentées ci-après sont donc calculées uniquement sur le sous-ensemble de couples de caméras clés sur lesquels l'estimation de λ a réussi. Ces résultats sont consignés dans le tableau 7.2.

	Méthode locale	Méthode globale	Méthode hybride
Réussite de l'estimation (%)	33,9	13,7	**56,3**
Erreur résiduelle moyenne sur la distance inter-caméra (m)	**0,11**	0,15	0,17
Ecart-type (m)	**0,13**	0,17	0,19
Erreur résiduelle médiane sur la distance inter-caméra (m)	**0,06**	0,09	0,10
Erreur résiduelle moyenne sur la distance inter-caméra (%)	**6,19**	6,32	6,82
Ecart-type (%)	**5,79**	6,11	6,66
Erreur résiduelle médiane sur la distance inter-caméra (%)	**4,27**	4,53	4,82

TABLE 7.2 – **Statistiques sur l'estimation du facteur d'échelle.** Les statistiques sont uniquement calculées sur les couples d'images clés pour lesquels le calcul du facteur d'échelle a réussi.

Rappelons que le critère définissant si une caméra clé doit être créée ou non est uniquement basée sur l'information image (section 2.6.1). Dès lors, la distance réelle entre deux caméras clés varie d'un couple de caméras clés successives à l'autre. Pour mesurer la qualité de l'estimation du facteur d'échelle, il est donc important de prendre en compte à la fois l'erreur absolue (en mètre) mais également l'erreur relative (en pourcentage).

La première observation importante qu'il est possible de faire est que les résultats sur la précision de l'estimation de λ sont quasiment équivalents pour les trois approches. En effet, l'erreur résiduelle moyenne sur la distance inter-caméra clé (où le facteur a pu être calculé) est d'environ 15 centimètres (soit 6,32%) pour toutes les méthodes. Les variations obtenues d'une méthode à l'autre sont au maximum de 6 centimètres (entre la méthode locale et la méthode hybride). Cependant, cette mesure n'est pas significative puisqu'elle rentre dans l'incertitude de la vérité terrain.

La différence importante entre les différentes méthodes se situe sur le taux d'estimations réussies. La méthode hybride a un taux de réussite de 56,3% (c'est à dire en moyenne une estimation réussie tous les 6,3 mètres) alors que la méthode locale et globale parviennent à calculer λ dans respectivement 33,9% et 13,7% des cas (soit respectivement en moyenne tous les 10,6 mètres et 26,9 mètres). Le faible taux obtenu par la méthode globale s'explique à la fois de part le fait que cette méthode n'utilise aucun *a priori* sur le facteur d'échelle, ce qui rend la sélection des points sur le sol plus délicate, et de part la validation stricte qui peut dans certains cas éliminer des estimations correctes.

Sur cette séquence, les résultats obtenus par la méthode locale sont très corrects et relativement proches de ceux obtenus par la méthode hybride, même en terme de taux de calculs réussis. En effet, sur cette séquence vidéo, l'hypothèse que le facteur d'échelle est suffisamment souvent calculé correctement pour que la méthode locale fonctionne est constamment respectée. En vue de montrer l'intérêt majeur de l'approche hybride pour la robustesse de l'estimation de λ, nous allons désormais nous placer dans un scénario spécifique qui reflète une situation courante dans les conditions réelles de circulation.

7.5.2.2 Scénario de trafic dense

Le scénario dans lequel vont être comparées les méthodes locale et hybride est détaillé dans la figure 7.16. Dans ce scénario, nous simulons un trafic dense sur une large zone du trajet (400 mètres) qui empêche de pouvoir calculer le facteur d'échelle sur l'ensemble de cette zone.

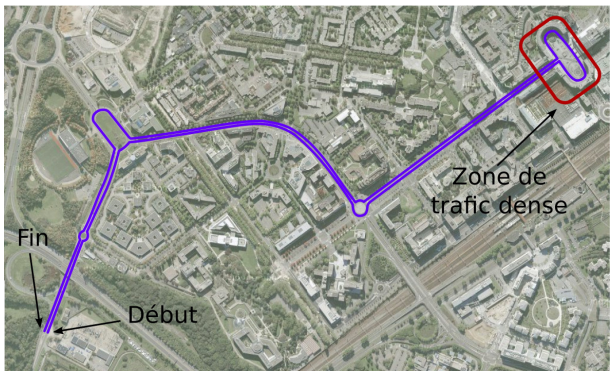


FIGURE 7.16 – **Scénario d'un trafic dense.** Dans ce scénario, on fait l'hypothèse que le trafic est trop dense pour permettre l'estimation du facteur d'échelle sur une large période.

En pratique, nous lançons donc les deux méthodes d'estimation à comparer en invalidant toutes les estimations de λ faites sur cette zone. Le tableau 7.3 rassemble les statistiques alors obtenues.

Ces résultats mettent en avant les résultats que nous pouvions attendre. La méthode locale voit son taux de calculs réussis passer de 33,9% à 16,6%. Ceci indique que la méthode n'a

	Méthode locale	Méthode hybride
Réussite de l'estimation (%)	16,6	**44**
Erreur résiduelle moyenne sur la distance inter-caméra (m)	**0,12**	0,15
Ecart-type (m)	**0,15**	0,18
Erreur résiduelle médiane sur la distance inter-caméra (m)	**0,08**	0,09
Erreur résiduelle moyenne sur la distance inter-caméra (%)	**6,22**	6,67
Ecart-type (%)	**5,98**	6,54
Erreur résiduelle médiane sur la distance inter-caméra (%)	**4,47**	4,83

TABLE 7.3 – **Statistiques sur l'estimation du facteur d'échelle pour le scénario proposé.** Les statistiques sont uniquement calculées sur les couples d'images clés pour lesquels le calcul du facteur d'échelle a réussi.

pas pu reprendre l'estimation de λ après la zone de trafic. En effet, l'estimation n'ayant pas pu s'effectuer sur 400 mètres, la dérive du SLAM est trop importante pour que la dernière estimation du facteur d'échelle soit une approximation correcte du facteur d'échelle courant. Au contraire, on voit que la méthode hybride a un taux de réussite beaucoup plus élevé. Cette méthode est donc bien capable d'estimer le facteur d'échelle sans aucun *a priori* sur sa valeur. L'estimation du facteur d'échelle a donc pu reprendre normalement après la zone de trafic.

7.5.2.3 Temps de traitement

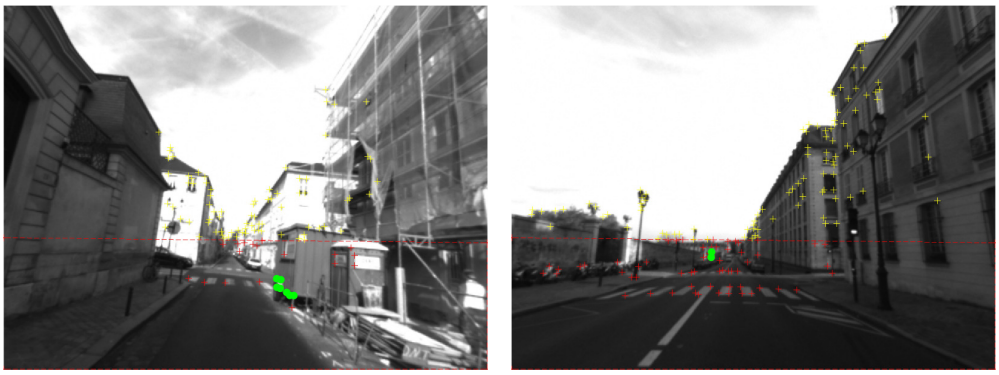
La méthode d'estimation hybride du facteur d'échelle est actuellement prototypée en Matlab, avec un calcul des dérivées en numérique lors de l'optimisation non-linéaire. L'estimation du facteur d'échelle entre deux images clés nécessite entre 0,4 et 0,9 secondes, en fonction des étapes nécessaires à cette estimation (estimation globale suffisante ou pas, nécessité d'une validation stricte, *etc.*). Il est donc raisonnable de penser que le passage en C++ rendra possible cette estimation sans réduire les performances du processus de SLAM. Cela est d'autant plus vrai que l'estimation du facteur d'échelle est indépendante du processus de SLAM. Il serait alors naturel que cette estimation soit réalisée dans un thread séparé.

7.5.3 Discussion

L'ensemble des résultats précédents montrent que la méthode hybride proposée permet de conjuguer les avantages des méthodes locale et globale. De plus, les statistiques obtenues tendent à montrer que l'approche proposée permet d'estimer le facteur d'échelle régulièrement (de l'ordre d'une caméra clé sur deux en moyenne) et de façon relativement précise (avec une erreur moyenne d'environ 7%).

Néanmoins, les résultats obtenus sur la séquence Versailles 1 (section 5.2.1) viennent pondérer ces résultats. En effet, même si aucune vérité terrain ne permet de quantifier les résultats obtenus sur cette séquence, il apparaît clairement que le facteur d'échelle est calculé moins fréquemment (de l'ordre d'une caméra clé sur quatre) et avec beaucoup moins de précision. Cette

baisse de résultats s'explique par le fait que dans cette séquence, l'hypothèse que la normale au sol soit constante par rapport à la caméra n'est pas respectée. En effet, les routes empruntées sont de petites routes bombées et abîmées, ce qui peut entraîner une variation de la normale. Ceci entraîne alors une mauvaise détection des points sur le sol et donc une mauvaise estimation de λ (ces deux processus étant directement liés à la connaissance de $\mathbf{n}$ et d). La figure 7.17 illustre ce type de problème. La mauvaise connaissance de la normale se traduit visuellement par une mauvaise estimation de la ligne d'horizon.



(a) (b)

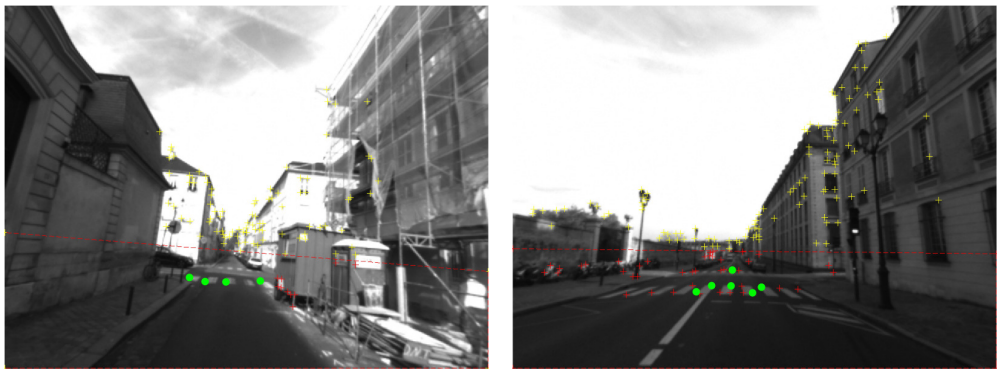
FIGURE 7.17 – **Illustration de la mauvaise estimation de la normale du sol.** Dans le cas où la surface de la route n'est pas parfaitement plane, l'hypothèse de constance de la normale du sol n'est plus vérifiée. La sélection des points sur la route échoue alors. Les croix rouges sont les points d'intérêt détectés dans la zone route et les ronds verts sont ceux retenus par la méthode proposée.

Cependant, pour plusieurs endroits délicats de la séquence, nous avons essayé de fournir à la méthode une estimation correcte de la normale (cette estimation ayant été faite manuellement). La figure 7.18 illustre alors le fait que, dès lors que la donnée $\mathbf{n}$ est correcte, la méthode proposée permet de calculer avec réussite λ là même où elle échouait précédemment. Ainsi, remettre en cause la normale de la route permettrait d'améliorer la robustesse de la méthode proposée. Pour cela, nous pouvons par exemple penser à utiliser un capteur tiers (*e.g.* une centrale inertielle) ou une méthode d'estimation des points de fuite basée uniquement sur la vision (par exemple en utilisant les résultats récents proposés par Tardif (2009)).

Cette validation expérimentale a montré qu'il était possible d'estimer efficacement le facteur d'échelle du SLAM uniquement à partir des données images. Néanmoins, ce calcul étant réalisé dans un processus extérieur, il est nécessaire de réinjecter cette nouvelle donnée dans le processus de SLAM afin de corriger la localisation du véhicule au fur et à mesure de son parcours. Dans la section suivante, nous allons en ce sens présenter la méthode d'intégration du facteur d'échelle proposée et les résultats de localisation alors obtenus.

7.6 Intégration du facteur d'échelle dans la méthode SLAM

Dans cette section, nous allons poser les notations liées à l'intégration du facteur d'échelle. Après avoir décrit les méthodes existantes et leurs limites, nous proposerons une solution permettant de résoudre notre problème en temps-réel.



(a) (b)

FIGURE 7.18 – **Résultat obtenu avec une meilleure estimation de la normale.** Dès lors que l'estimation de la normale est correcte, la méthode d'estimation du facteur d'échelle proposée fournit à nouveau des résultats corrects. Les croix rouges sont les points d'intérêt détectés dans la zone route et les ronds verts sont ceux retenus par la méthode proposée.

7.6.1 Contexte du problème étudié

La méthode proposée dans la section 7.4 permet d'estimer le facteur d'échelle entre deux caméras clés successives. Néanmoins, nous avons vu précédemment que les informations extraites des images ne sont pas toujours suffisantes pour rendre l'estimation du facteur d'échelle possible. L'estimation du facteur d'échelle n'est donc pas disponible pour tous les couples de caméras clés successives.

Le contexte de la problématique liée à l'intégration du facteur d'échelle est donc celui décrit dans la figure 7.19. Nous considérons que le facteur d'échelle λ_i a pu être calculé entre les caméras $\mathcal{C}_{i-1}$ et $\mathcal{C}_i$ et que cette donnée a été intégrée à la méthode SLAM. La distance entre les caméras $\mathcal{C}_{i-1}$ et $\mathcal{C}_i$ a donc été corrigée en fonction de la valeur λ_i. Pour les caméras suivantes (entre $\mathcal{C}_i$ et $\mathcal{C}_{j-1}$), nous considérons que le facteur d'échelle n'a pas pu être estimé. La distance inter-caméra pour toutes ces caméras est donc celle qui est fournie par le processus de SLAM. Nous nous plaçons alors dans le cas où une nouvelle caméra clé $\mathcal{C}_j$ est créée et que le facteur d'échelle λ_j entre les caméras $\mathcal{C}_{j-1}$ et $\mathcal{C}_j$ a pu être estimé. Le problème est alors de savoir comment intégrer cette nouvelle donnée dans la méthode de SLAM.

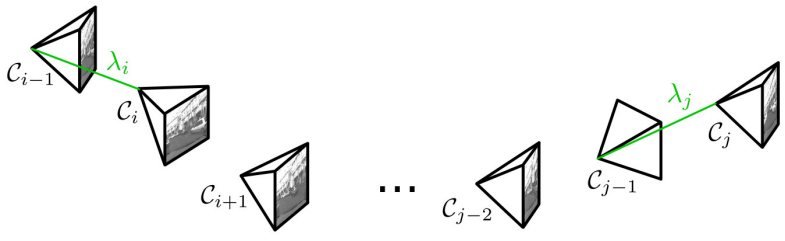


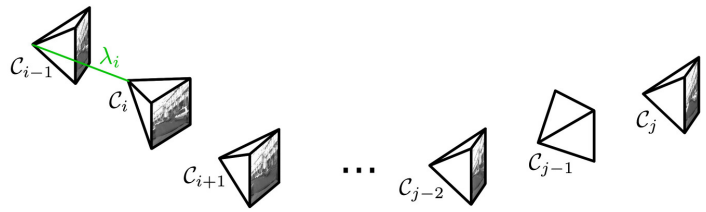
FIGURE 7.19 – **Contexte de l'intégration du facteur d'échelle.** Le facteur d'échelle n'a pu être estimé que pour les couples de caméras $(\mathcal{C}_{i-1}, \mathcal{C}_i)$ et $(\mathcal{C}_{j-1}, \mathcal{C}_j)$.

Puisque le facteur d'échelle entre les caméras $\mathcal{C}_{j-1}$ et $\mathcal{C}_j$ a pu être estimé, nous disposons de deux hypothèses différentes sur la position réelle de la caméra $\mathcal{C}_j$. En se rapportant aux notions

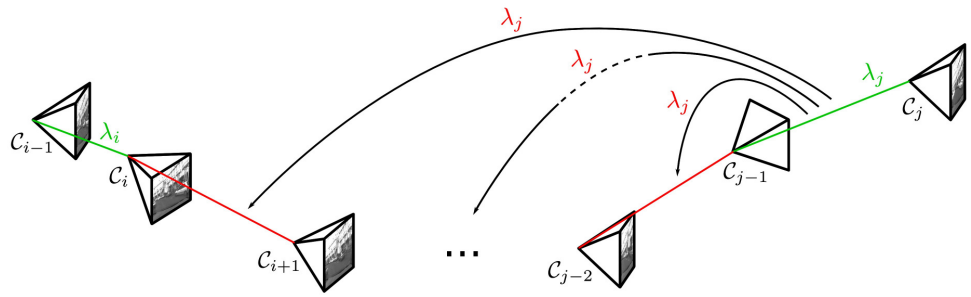
généralement utilisées en fusion de données (*e.g.* Konolige et al. (2007)), nous avons, pour la position réelle de $\mathcal{C}_j$, deux observations différentes (figure 7.20), à savoir :

▷ **Observation 1.** La première observation sur la position réelle est la position fournie par la méthode de SLAM seule (figure 7.20(a)).

▷ **Observation 2.** La valeur de λ_j permet d'obtenir une autre position probable de la caméra clé courante. Cette deuxième position peut par exemple être obtenue en corrigeant la distance entre toutes les paires de caméras clés successives entre les caméras $\mathcal{C}_i$ et $\mathcal{C}_j$ à l'aide du facteur λ_j (figure 7.20(b)).



(a) Observation 1



(b) Observation 2

FIGURE 7.20 – **Observations de la position de la caméra clé courante.** Le SLAM (a) et l'estimation du facteur d'échelle (b) fournissent deux positions probables de la caméra clé courante.

Le problème est alors de savoir comment fusionner ces deux informations afin d'obtenir la meilleure estimation possible de la position de la caméra $\mathcal{C}_j$.

7.6.2 Limites des méthodes existantes

Les problématiques de fusion d'informations représentent un domaine très actif de la recherche, en particulier en robotique. Pour résoudre le problème auquel nous sommes confrontés, deux méthodes sont couramment utilisées : le *filtre de Kalman* et l'ajout d'une contrainte dans l'ajustement de faisceaux. Ces deux méthodes se différencient par le fait qu'elles utilisent une approche statistique ou géométrique du problème.

7.6.2.1 Filtre de Kalman

La méthode la plus classique en fusion de données est le filtre de Kalman (Kalman (1960)). L'idée de cet algorithme est de mettre à jour la position courante estimée de la caméra (*i.e.* la

prédiction) à l'aide des nouvelles données observées (*i.e.* les observations). Cette étape, appelée mise à jour, est réalisée à partir de formules mathématiques définies par le filtre de Kalman.

Néanmoins, pour que cette étape de mise à jour soit efficace, il est nécessaire de connaître les covariances des données de prédiction et d'observation, c'est à dire une information sur l'incertitude associée à ces données. Cependant, dans notre étude, nous n'avons accès à aucune information de ce type. En effet, il est très difficile de pouvoir fournir une incertitude relative à l'estimation du facteur d'échelle. De plus, même si des méthodes récentes proposent un calcul de covariance pour la méthode de SLAM utilisée (Eudes and Lhuillier (2009)), cette covariance ne prend pas en compte la dérive du facteur d'échelle. Cela signifie que si le facteur d'échelle n'est pas corrigé durant une période importante, la covariance calculée peut devenir incohérente avec la position de la caméra clé associée.

Dès lors, il est déconseillé d'utiliser le filtre de Kalman. En effet, Mittu and Segaria (2000) ont montré que l'efficacité de ce filtre est directement liée à la qualité des covariances associées aux différentes données.

7.6.2.2 Contrainte de l'ajustement de faisceaux

Une autre approche parfois utilisée est d'ajouter une information géométrique dans l'ajustement de faisceaux du SLAM de façon à contraindre la convergence lors de l'optimisation. En pratique, l'idée consiste à ajouter à la fonction de reprojection classique une autre fonction de coût relative à la contrainte additionnelle. Dans notre étude, cette fonction (notée $\mathcal{J}_i$) pourrait par exemple être la différence de distance inter-caméra entre les caméras $\mathcal{C}_{i-1}$ et $\mathcal{C}_i$ et la distance fournie par l'estimation du facteur d'échelle. La fonction de coût finale $\mathcal{G}$ à optimiser est alors du type :

$$\mathcal{G} = (\mathcal{F}_{i+1} + \alpha_{i+1}\mathcal{J}_{i+1}) + \cdots + (\mathcal{F}_j + \alpha_j \mathcal{J}_j) \tag{7.10}$$

où $(\alpha_i)_i$ sont des paramètres qui permettent de pondérer l'intervention de la contrainte supplémentaire dans la minimisation et $\mathcal{F}_i$ la fonction de reprojection classique liée à la caméra $\mathcal{C}_i$. L'optimisation de la fonction $\mathcal{G}$ est très dépendante de la valeur des $(\alpha_i)_i$. Il est donc nécessaire de calculer précisément les $(\alpha_i)_i$ afin d'obtenir le minimum recherché. Dans le cas où on cherche à estimer le maximum de vraisemblance sous l'hypothèse que la distribution de l'erreur est gaussienne, la valeur des $(\alpha_i)_i$ peut être directement déduite des covariances. Comme nous l'avons vu précédemment, les covariances (et donc les $(\alpha_i)_i$) sont inconnues dans notre cadre d'étude. Dans ce contexte, des méthodes ont été proposées afin d'estimer automatiquement la valeur des $(\alpha_i)_i$. En particulier, la validation croisée a déjà été utilisée dans des contextes de SLAM (Farenzena et al. (2008)) et plus récemment, une méthode basée sur les L-Curve a été proposée par Michot et al. (2010) pour fusionner la méthode de SLAM de Mouragnon et al. (2006) avec une donnée provenant d'un odomètre.

Néanmoins, ces méthodes sont très coûteuses en temps de calcul et sont donc classiquement utilisées sur l'optimisation d'une seule caméra, la recherche du paramètre α étant alors un problème à une seule dimension. Dans notre cas, il est nécessaire d'optimiser $(j - i)$ caméras simultanément. Le problème de recherche des critères $(\alpha_i)_i$ devient alors un problème à $(j - i)$ dimensions. Le temps de calcul alors nécessaire devient trop important pour envisager un traitement temps-réel.

7.6.3 Approche retenue

Comme nous avons pu le voir, il est particulièrement difficile de fusionner en temps-réel les différentes informations que nous possédons, pour deux raisons en particulier :

- ▷ la dérive du facteur d'échelle n'étant pas quantifiable, il n'est rapidement plus possible de faire confiance à la position fournie par le SLAM.
- ▷ le facteur d'échelle n'étant pas disponible à chaque caméra clé dans notre cas, il est nécessaire de corriger un ensemble parfois important de caméras. Les temps de calcul nécessaires à l'utilisation d'une optimisation simultanée de toutes ces caméras sont alors trop importants pour des applications temps-réel.

De part ces différentes remarques, nous avons décidé d'utiliser une méthode très simple mais qui nous permettra de tester rapidement l'intégration du facteur d'échelle calculé dans la méthode de SLAM. L'idée que nous utilisons est alors de faire entièrement confiance à notre observation, c'est à dire au facteur d'échelle calculé. Notons dès à présent que le même type d'approche a été employée par Scaramuzza et al. (2009b). En effet, dans leurs travaux, la norme du déplacement de la caméra est fixée comme étant celle fournie par l'odomètre et n'est jamais remise en cause par l'information issue du flux vidéo.

Nous avons montré précédemment (figure 7.20(b)) qu'à partir de la valeur λ_j, il est possible de corriger l'historique de la reconstruction et ainsi d'en déduire la position de la caméra courante $\mathcal{C}_j$. Néanmoins, en propageant de cette façon le facteur d'échelle λ_j entre les caméras $\mathcal{C}_i$ et $\mathcal{C}_j$, nous faisons l'hypothèse que le facteur d'échelle est constant entre ces caméras. Cette hypothèse n'étant que très approximative, il serait alors préférable de réaliser suite à cela un ajustement de faisceaux sur l'ensemble des caméras entre $\mathcal{C}_{i+1}$ et $\mathcal{C}_{j-1}$, processus qui augmenterait alors fortement les temps de calcul et qui empêcherait donc le fonctionnement temps-réel de la méthode SLAM.

L'idée que nous utiliserons est donc de propager entre les caméras $\mathcal{C}_i$ et $\mathcal{C}_j$ un facteur d'échelle qui correspond le plus possible à la réalité. Ainsi, plutôt que de propager λ_j sur tous les couples de caméras, nous réalisons une interpolation linéaire entre λ_i et λ_j (figure 7.21). En effet, nos expériences sur l'analyse du facteur d'échelle (figure 5.3, page 72) nous ont montré que la dérive du facteur d'échelle peut généralement être modélisée grossièrement comme étant localement linéaire.

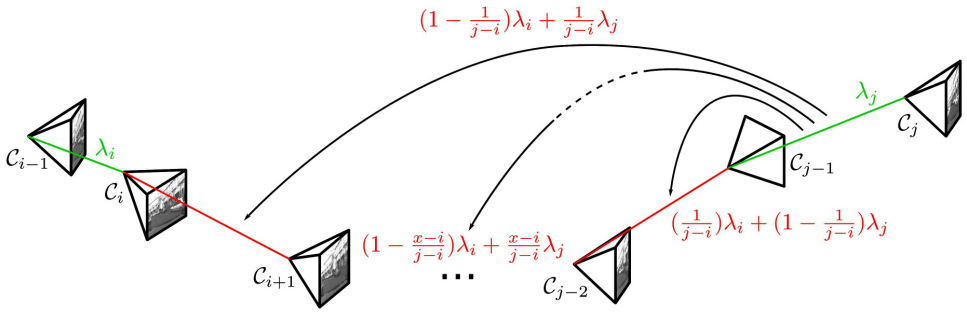


FIGURE 7.21 – **Approche proposée pour l'intégration du facteur d'échelle.** Le facteur d'échelle propagé entre les caméras $\mathcal{C}_i$ et $\mathcal{C}_j$ est une interpolation linéaire entre λ_i et λ_j.

Une fois les caméras ainsi repositionnées, les points 3D observés sont retriangulés à partir de ces nouvelles poses de caméras. Cette étape rapide permet de garder la cohérence entre les caméras clés et le nuage de points 3D. De plus, le nuage de points étant remis à la bonne échelle, le processus de SLAM va implicitement transmettre le facteur d'échelle ainsi corrigé à la suite de la reconstruction.

L'approche proposée est simple mais permet d'intégrer le facteur d'échelle dans la méthode de SLAM sans nécessiter un temps de calcul important. Cela nous permettra en particulier de tester rapidement l'apport pour le processus SLAM de l'estimation du facteur d'échelle. Dans la section ci-après, nous allons de plus mettre en évidence que, malgré la simplicité de l'approche retenue, les résultats obtenus sont très satisfaisants.

7.7 Résultats expérimentaux

Cette section a pour but de présenter les résultats obtenus avec la méthode de correction du facteur d'échelle proposée dans ce chapitre (*i.e.* l'estimation et l'intégration du facteur d'échelle). Après avoir décrit le protocole expérimental suivi pour ces expériences, nous détaillerons alors les résultats obtenus sur deux séquences différentes.

7.7.1 Protocole expérimental

Le facteur d'échelle qui sera intégré dans la méthode de SLAM est celui calculé avec la méthode proposée dans le chapitre précédent (appelée *méthode hybride* dans la section 7.5).

Avec la méthode d'intégration du facteur d'échelle retenue dans ce chapitre, il existe deux estimations distinctes de la position de chacune des caméras clés. La première position possible est celle estimée par la méthode SLAM, au moment même de la création de cette caméra clé. Le facteur d'échelle associé à cette position provient alors de la dernière estimation possible du facteur d'échelle (pour un couple de caméras clés précédent) qui a ensuite été propagée par le processus de SLAM. L'autre position possible est celle obtenue après la correction *a posteriori* du facteur d'échelle, c'est à dire suite à la rétropropagation du facteur d'échelle d'un couple de caméras ultérieur. Le facteur d'échelle associé à cette position est alors issu de l'interpolation linéaire entre le dernier facteur d'échelle estimé avant celle-ci et le premier facteur d'échelle estimé après celle-ci. Dès lors, deux types de résultats peuvent être considérés en fonction de l'application visée :

- ▷ **Trajectométrie.** Si le but est de reconstruire la trajectoire du véhicule *a posteriori*, les données à étudier sont alors celles liées à la reconstruction SLAM finale, c'est à dire incluant la correction de l'historique. Dans la suite, nous parlerons de *reconstruction avec historique corrigé*.

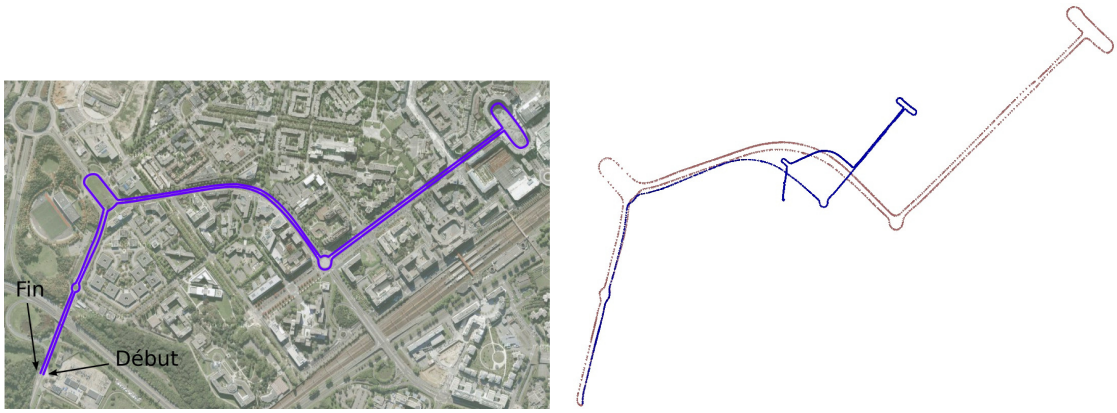
- ▷ **Localisation temps-réel.** Si le but est de localiser un véhicule en temps-réel (à l'instar des systèmes de localisation basés GPS), les données à étudier sont les positions de la caméra mobile à chaque instant. En particulier, cela revient à dire que la position finale des caméras clés est celle qu'elles ont à leur création, sans aucune correction *a posteriori*. Nous désignerons cette reconstruction comme *localisation à chaque instant*.

7.7.2 Résultats obtenus

Dans cette section, nous détaillerons les résultats obtenus tout d'abord sur la séquence ODIAAC puis sur la séquence Versailles 1.

7.7.2.1 Séquence ODIAAC

Dans cette première sous-section, nous allons décrire les résultats obtenus sur la séquence ODIAAC qui est détaillée dans la section 7.5.1. Un résumé de cette séquence peut être trouvé dans la figure 7.22. Comme nous l'avons vu précédemment, l'avantage de cette séquence est essentiellement que nous possédons une vérité terrain (fournie par un trajectomètre) de la position de la caméra à chaque instant.



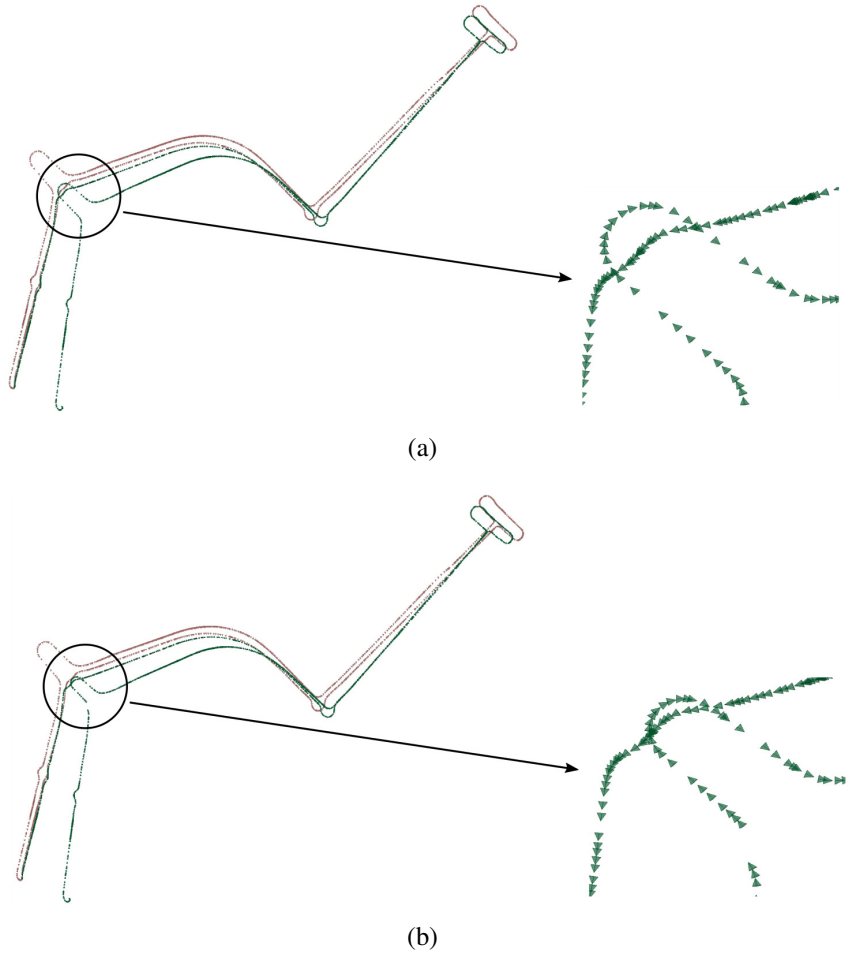
(a) (b)

FIGURE 7.22 – **Résumé de la séquence ODIAAC.** La séquence ODIAAC (a) est une trajectoire de 4,5 kilomètres. (b) est la trajectoire reconstruite par la méthode SLAM seule (en bleu) comparée à la vérité terrain (en rouge).

Les deux reconstructions possibles obtenues après intégration du facteur d'échelle (*i.e.* la reconstruction avec historique corrigé et la localisation à chaque instant) sont regroupées dans la figure 7.23. Ces deux reconstructions sont très proches, la différence majeure étant que la localisation à chaque instant présente localement des discontinuités dans la trajectoire (figure 7.23(b)). En effet, la position des caméras clés pour lesquelles le facteur d'échelle peut être calculé est corrigée à l'aide de cette donnée supplémentaire, ce qui implique cette discontinuité.

En comparant les figures 7.22(b) et 7.23, il apparaît visuellement que la méthode de calcul du facteur d'échelle et de son intégration permet d'éviter la dérive de ce facteur observée habituellement dans la méthode de SLAM classique. Notons de plus que la méthode proposée permet en particulier d'obtenir la métrique de la trajectoire reconstruite, information qui n'est pas disponible dans les méthodes de SLAM monoculaire.

Les résultats numériques présentés dans la figure 7.24 et le tableau 7.4 corroborent les observations faites précédemment. Notons que contrairement à la section 7.5, les statistiques sont ici calculées sur l'ensemble des couples de caméras clés. Le SLAM seul ne possédant aucune métrique, nous avons initialisé les deux premières caméras clés avec la distance fournie par le trajectomètre.



(a) (b)

FIGURE 7.23 – **Reconstructions obtenues après l'intégration du facteur d'échelle sur la séquence ODIAAC.** La première ligne est la reconstruction avec historique corrigé. La deuxième ligne est la localisation à chaque instant.

	Méthode SLAM seule	Reconstruction avec historique corrigé	Localisation à chaque instant
Erreur moyenne sur la distance inter-caméra (m)	1,86	**0,26**	0,40
Ecart-type (m)	1,76	**0,25**	0,78
Erreur médiane sur la distance inter-caméra (m)	1,46	**0,16**	0,21
Erreur moyenne sur la distance inter-caméra (%)	54,49	**7,44**	14,91
Ecart-type (%)	27,63	**6,16**	35,59
Erreur médiane sur la distance inter-caméra (%)	66,37	**6,37**	8,48

TABLE 7.4 – **Statistiques sur l'intégration du facteur d'échelle sur la séquence ODIAAC.** Les statistiques sont calculées sur l'ensemble des caméras clés.

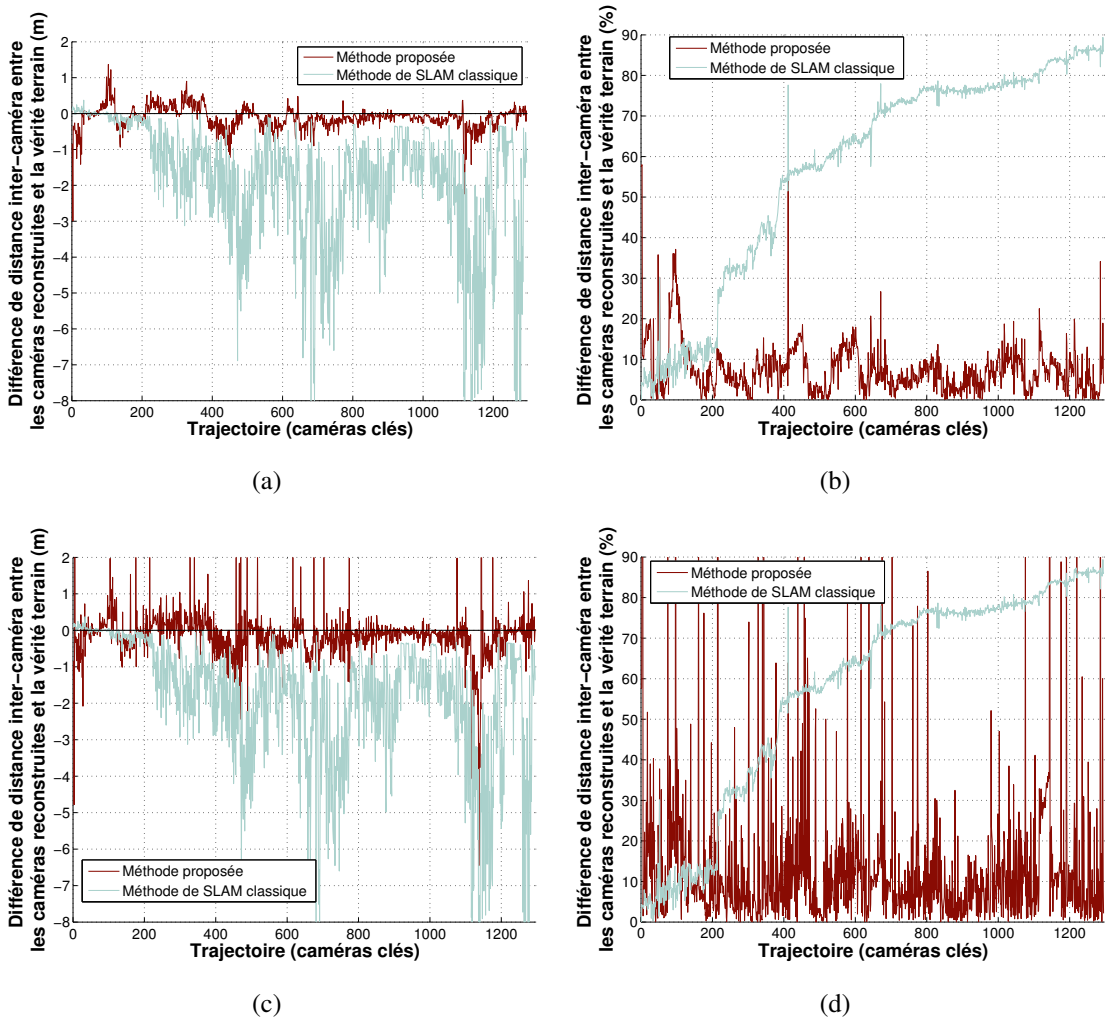


FIGURE 7.24 – **Résultats numériques sur l'intégration du facteur d'échelle sur la séquence ODIAAC.** La première ligne (a,b) est relative à la reconstruction avec historique corrigé. La deuxième ligne (c,d) est relative à la localisation à chaque instant.

Le bruit observable sur les données de localisation à chaque instant est lié à la discontinuité de la trajectoire dont nous avons discuté précédemment. Il est donc préférable de prendre en compte la médiane de l'erreur plutôt que sa moyenne, cette dernière étant plus sensible à ce type de bruit. Cette observation étant faite, nous retrouvons numériquement le bénéfice de la correction de la dérive du facteur d'échelle. En effet, si l'erreur moyenne de distance inter-caméra est de 54,49% dans la méthode de SLAM seule, elle est réduite à environ 10% après intégration du facteur d'échelle.

7.7.2.2 Séquence Versailles 1

La méthode d'intégration du facteur d'échelle a également été testée sur la séquence Versailles 1 (précédemment détaillée à la section 5.2). Rappelons que pour cette séquence, nous ne disposons d'aucune vérité terrain. Il n'est donc pas possible de comparer ici la reconstruction *a posteriori* de la localisation à chaque instant. De plus, les deux types de reconstruction étant visuellement identiques, nous ne présenterons ici que la localisation à chaque instant.

Comme nous l'avons vu dans le chapitre précédent (section 7.5.3), le calcul du facteur d'échelle est délicat sur la séquence tournée à Versailles du fait de la non-planarité de la route. La normale de la route n'est alors pas constante tout au long du trajet, ce qui entre en contradiction avec nos hypothèses de travail. Aucune méthode d'estimation automatique (par vision, IMU, *etc.*) n'étant pour l'instant intégrée à notre méthode d'estimation du facteur d'échelle, nous ne traiterons que la sous-partie de la séquence de Versailles 1 pour laquelle l'hypothèse de planarité de la route est majoritairement respectée (figure 7.25(a)).



FIGURE 7.25 – **Résultats sur la séquence de Versailles.** (a) est la vérité terrain, (b) la reconstruction SLAM seule et (c) la reconstruction obtenue après intégration du facteur d'échelle.

En comparant visuellement la reconstruction obtenue par la méthode SLAM (figure 7.25(b)) avec le résultat de notre méthode (figure 7.25(c)), nous pouvons apercevoir que la dérive du facteur d'échelle est fortement réduite. En effet, en superposant les différentes reconstructions sur le modèle 3D de la ville, il est possible de mettre en avant que la trajectoire obtenue grâce à la méthode proposée est plus proche de la vérité terrain que la trajectoire obtenue par la méthode de SLAM seule.

7.8 Discussion

Nous avons pu voir dans ce chapitre que la méthode proposée pour estimer puis intégrer le facteur d'échelle dans le SLAM permet de réduire sensiblement la dérive inhérente à cette méthode. Par exemple, nous avons vu que sur une séquence de 4,5 kilomètres, l'erreur moyenne de distance inter-caméra passe de 54,49% à environ 10%. Il est donc généralement possible de garantir une localisation relativement précise sur plusieurs dizaines de mètres.

Néanmoins, à grande échelle, la correction du facteur d'échelle n'est pas suffisante. En effet, avec une erreur relative de distance inter-caméra de l'ordre de 10%, la dernière caméra clé (sur 4,5 kilomètres) pourrait être localisée dans le pire des cas avec une erreur de l'ordre de 450 mètres. Ceci est dû au fait que la méthode proposée n'utilise pas, en l'état, d'information absolue sur la localisation de la caméra. Ainsi, la dérive liée à l'accumulation d'erreur est toujours présente et peut être très elevée dès lors que la distance parcourue est importante.

C'est en ce sens que nous allons proposer dans le chapitre suivant une approche permettant d'apporter cette information absolue manquante. Ceci permettra de corriger ponctuellement la dérive d'accumulation du SLAM et ainsi de permettre une localisation précise du véhicule sur de longues distances (*i.e.* plusieurs kilomètres).

CHAPITRE 8

Méthode de correction de l'accumulation d'erreur

Dans ce chapitre, nous allons montrer qu'il est possible d'extraire d'un Système d'Information Géographique une information absolue sur la position courante de la caméra. Nous montrerons alors que cette information peut être exploitée afin de corriger ponctuellement la dérive de la méthode de SLAM liée à l'accumulation d'erreur. En particulier, nous montrerons qu'un modèle 3D de la ville ou une carte de la route peuvent être utilisés pour corriger ponctuellement la position courante de la caméra.

8.1 Objectif de l'étude

L'objectif de ce chapitre est de corriger la dérive résiduelle observée sur la méthode de SLAM encore présente après la correction du facteur d'échelle. En effet, les expériences présentées à la section 7.7 ont mis en avant que, malgré la correction du facteur d'échelle, la dérive résiduelle liée à l'accumulation d'erreur peut être encore très importante sur des trajectoires de grande échelle. Le but de ce chapitre est donc d'estimer et de corriger cette dérive. Pour cela, nous exprimons cette dérive comme étant la transformation 3D qui existe entre la position courante estimée de la caméra et sa position réelle. Notons dès à présent que la reconstruction SLAM est préalablement corrigée en facteur d'échelle grâce à la méthode proposée dans le chapitre 7. Ceci permet de simplifier la transformation 3D recherchée, ce qui rend son estimation plus robuste. Ainsi, la transformation recherchée dans ce chapitre est uniquement une transformation euclidienne, c'est à dire une rotation et une translation dans l'espace, et non une similitude.

Pour estimer cette transformation, nous proposons d'exploiter l'information apportée par un SIG, comme celui détaillé dans l'introduction de cette partie. En pratique, nous proposons d'estimer la dérive accumulée en cherchant ponctuellement la transformation euclidienne qui permet de recaler la reconstruction SLAM courante sur le SIG. Rappelons que les méthodes de recalage ne sont efficaces que si la position initiale des deux ensembles à aligner n'est pas trop éloignée de la solution recherchée. Dans notre approche, c'est la correction du facteur d'échelle (chapitre 7) que nous effectuons au fil de la méthode SLAM qui permet de fournir cette initialisation.

Au cours de ce chapitre, nous préciserons tout d'abord les configurations dans lesquelles les contraintes géométriques entre la reconstruction SLAM et le SIG sont suffisantes pour permettre un tel recalage. En fonction des données disponibles sur l'environnement parcouru, l'estimation de la dérive sera réalisée en associant soit les points reconstruits avec le modèle 3D de la ville (section 8.3), soit les caméras clés avec la carte de la route (section 8.4). Une fois la dérive estimée, nous détaillerons alors comment cette information est intégrée dans la méthode de SLAM (section 8.5). Enfin, la méthode proposée sera évaluée sur plusieurs séquences de grande échelle (section 8.6).

8.2 Alignement et contraintes géométriques exploitables

Le but de cette section est de présenter les contraintes apportées par un Système d'Information Géographique. Comme nous allons le voir, l'information qu'il est possible d'en extraire est différente en fonction de la géométrie du lieu parcouru.

8.2.1 Paramètres contraints dans les lignes droites

Comme cela a été détaillé dans la section précédente, l'information recherchée dans ce chapitre est la transformation euclidienne (*i.e.* la translation et la rotation) qui existe entre la position estimée courante de la caméra et sa position réelle. Or, dans les lignes droites, il n'est pas possible d'estimer tous les paramètres de cette transformation. En effet, la reconstruction SLAM obtenue dans une ligne droite peut être translatée le long de la direction de sa trajectoire sans perdre la cohérence avec le SIG (figure 8.1) : on parle alors du problème d'*aperture*. Ce phénomène est intuitif lors de l'utilisation d'une carte de la route. Etant donné la simplicité des SIG exploités dans nos travaux, le phénomène d'aperture apparaît également lors de l'utilisation des modèles 3D. En effet, comme nous le savons, ces modèles 3D ne présentent très généralement qu'un unique plan pour représenter la façade des bâtiments (section 2.6.2.3). En conséquence, au cours des lignes droites, les contraintes géométriques fournies par ce modèle sont trop faibles pour estimer la dérive en translation.

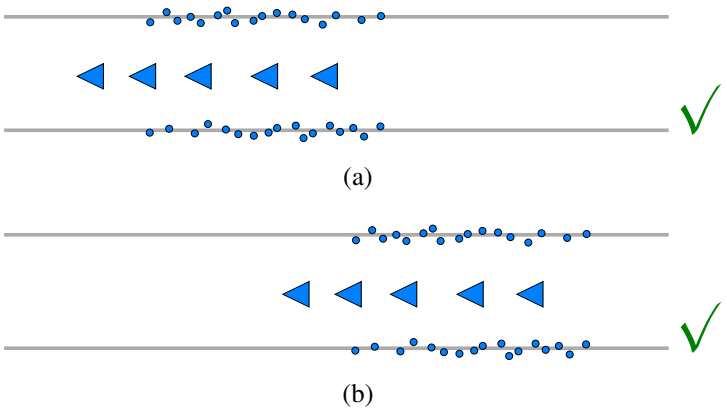


FIGURE 8.1 – **Illustration du problème d'aperture dans les lignes droites.** Les lignes droites ne fournissent pas assez d'informations géométriques pour contraindre la reconstruction SLAM. (a) et (b) sont deux positions différentes possibles pour la reconstruction.

Néanmoins, il est possible de retrouver l'erreur en orientation accumulée sur la reconstruction SLAM (figure 8.2). Dans notre cadre d'étude, nous ne pourrons cependant extraire cette

information que lors de l'utilisation d'une carte de la route. En effet, des expériences ont montré que la précision des modèles 3D de l'environnement et le bruit existant sur le nuage de points reconstruit ne permettent pas de recaler précisément ces données au cours des lignes droites.

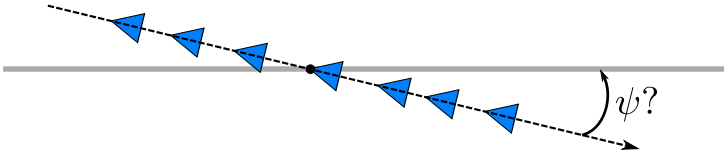
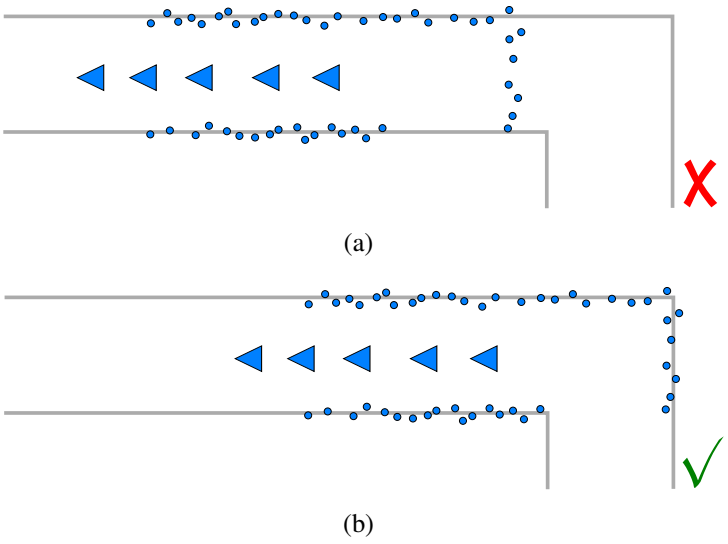


FIGURE 8.2 – **Estimation possible de la rotation.** Lors de l'utilisation d'une carte de la route, il est possible d'estimer l'erreur d'orientation accumulée.

8.2.2 Paramètres contraints dans les virages

Dans les virages, le problème d'aperture disparaît (figure 8.3). En effet, les contraintes apportées par le virage rendent possible la détermination de la rotation et de la translation recherchées, à la fois lors de l'utilisation d'une carte de la route et d'un modèle 3D (les façades des bâtiments étant très généralement parallèles à la route).



(a)

(b)

FIGURE 8.3 – **Contraintes géométriques dans les virages.** Les contraintes géométriques dans les virages permettent de fixer tous les paramètres de l'alignement recherché. En effet, contrairement à la figure (a), la figure (b) permet d'expliquer la géométrie reconstruite par le SLAM.

Notons qu'il sera donc nécessaire de différencier les lignes droites des virages au cours de la reconstruction. Afin de détecter les virages, nous utiliserons la méthode de polygonalisation proposée par Lowe (1987) à chaque nouvelle image clé. L'apparition d'un nouveau segment est alors équivalent à l'apparition d'un virage.

A partir des observations réalisées dans cette section, nous allons présenter comment est estimée la dérive en accumulation d'erreur dans le cas de l'utilisation d'un modèle 3D de la ville (section 8.3) puis dans le cas de l'utilisation d'une carte de la route (section 8.4).

8.3 Estimation de la dérive à l'aide d'un modèle 3D de ville

Dans notre problématique, l'estimation de la dérive est équivalente à la recherche de la transformation qui permet le recalage entre la reconstruction SLAM et le SIG 3D. Dans cette section, nous allons détailler la façon dont est effectué ce recalage lors de l'exploitation d'un modèle 3D de la ville.

8.3.1 Aperçu de la méthode

L'idée de cette section est de rechercher le meilleur recalage entre le nuage de points reconstruit et le modèle 3D de la ville. La méthode de recalage utilisée ici est une adaptation de l'approche proposée dans le chapitre 3. Néanmoins, des différences importantes impliquent des changements notables dans cette section :

- ▷ nous ne travaillons ici que sur un sous-ensemble de la reconstruction (qui correspond au virage courant). Les contraintes géométriques existantes entre le nuage de points reconstruit et les bâtiments 3D sont donc beaucoup plus réduites. Pour assurer la robustesse de l'alignement, nous limitons donc la transformation recherchée à une transformation 2D constituée de la position 2D de la reconstruction (dans le plan horizontal) et de son orientation autour de la verticale (angle de lacet).

- ▷ la méthode de correction du facteur d'échelle implique des erreurs relatives de l'ordre de 10% (section 7.7). Ainsi, après une longue ligne droite, l'erreur de positionnement des caméras clés peut être importante. Cette position étant utilisée comme initialisation du recalage, l'association point-plan au plus proche peut être erronée (figure 8.4(a)). Nous proposons donc désormais d'associer chaque point reconstruit non pas au plan le plus proche mais à celui qui est le plus probable. Pour cela, les amers reconstruits dans la méthode SLAM ne sont désormais plus de simples points 3D mais des patchs orientés. La normale de ces patchs peut alors être utilisée pour améliorer l'association des données (figure 8.4(b)).

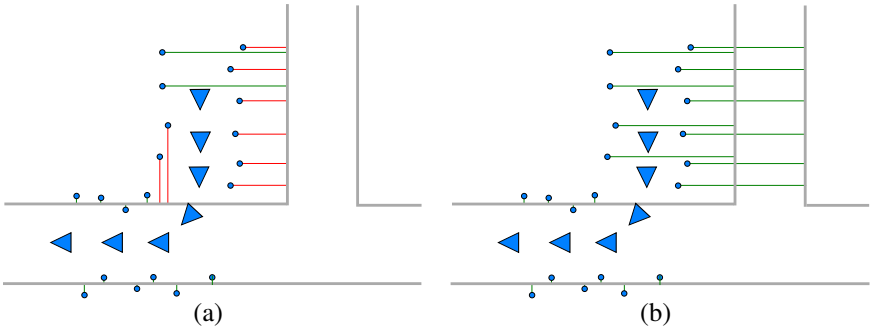


FIGURE 8.4 – **Amélioration des associations point-plan.** Les normales des points 3D peuvent être utilisées pour améliorer l'association des données. Les liens verts sont les associations correctes et les rouges les associations incorrectes. (a) représente les associations au plus proche. (b) représente les associations prenant en compte les normales.

Ainsi, après avoir présenté la façon dont sont estimés les patchs orientés, nous décrirons alors comment la méthode de recalage tire profit de cette nouvelle information.

8.3.2 Calcul de patchs orientés

La méthode présentée ici pour calculer la normale associée au voisinage de chaque point 3D reconstruit est inspirée des approches utilisées par exemple par Molton et al. (2004); Berger and Lacroix (2008); Charmette et al. (2009). Le contexte dans lequel nous nous plaçons est décrit dans la figure 8.5, où $\mathcal{C}_1$ et $\mathcal{C}_2$ sont deux images clés qui observent un point $\mathcal{Q}^i$.

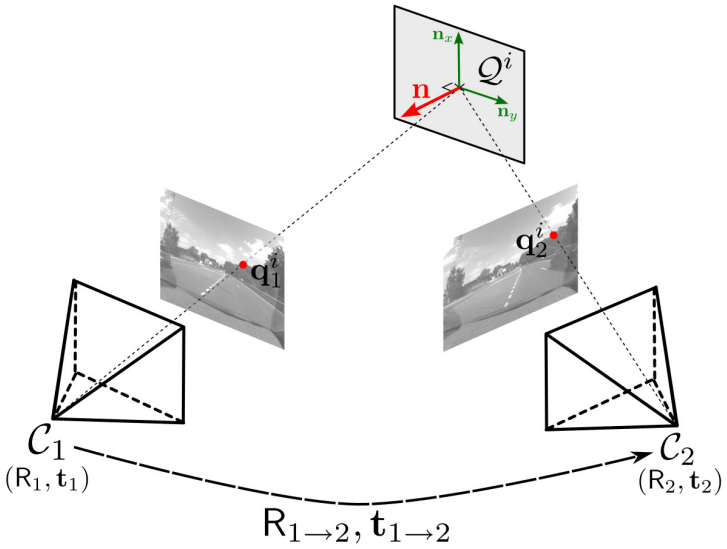


FIGURE 8.5 – **Contexte de l'estimation des normales.**

8.3.2.1 Formalisation du problème

L'idée générale sur laquelle s'appuie l'estimation de la normale d'un point 3D est de considérer que le voisinage de ce point 3D peut être approximé localement comme étant plan. Dès lors, nous savons que le déplacement relatif entre les deux caméras, l'équation de ce plan et l'homographie suivie par les points situés sur ce plan sont liées par la relation :

$$\mathsf{H}_{1\to2} \sim \mathsf{K}(\mathsf{R}_{1\to2} - \frac{\mathbf{t}_{1\to2}\mathbf{n}^\mathsf{T}}{d})\mathsf{K}^{-1} \tag{8.1}$$

où le déplacement relatif $(\mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$ entre les caméras clés est connu (car fourni par le SLAM). $\mathbf{n}$ est la normale du plan et d la distance entre le plan et la caméra $\mathcal{C}_1$. Cette distance peut être exprimée comme le produit scalaire entre les coordonnées du point 3D : $d = \mathcal{Q}^\mathsf{T}\mathbf{n}$. La normale est alors la seule inconnue de l'équation :

$$\mathsf{H}_{1\to2}(\mathbf{n}) \sim \mathsf{K}(\mathsf{R}_{1\to2} - \frac{\mathbf{t}_{1\to2}\mathbf{n}^\mathsf{T}}{\mathcal{Q}^\mathsf{T}\mathbf{n}})\mathsf{K}^{-1} \tag{8.2}$$

8.3.2.2 Estimation initiale

Une première estimation de la normale (notée $\mathbf{n}_0$) est nécessaire lors de la création d'un point 3D, c'est à dire lorsqu'il est vu pour la première fois. La normale initiale $\mathbf{n}_0$ est fixée comme étant le vecteur reliant le point 3D et le centre de la première caméra qui l'observe.

8.3.2.3 Optimisation de la normale

A chaque nouvelle image clé, la normale de chacun des points 3D qu'elle observe peut être raffinée. Ainsi, pour raffiner la normale $\mathbf{n}_j$ d'un point $\mathcal{Q}^i$, on minimisera l'erreur de transfert entre la première caméra clé observant ce point et la caméra clé courante.

$$\mathbf{n}_{j+1} = \underset{\mathbf{n}}{argmin} \sum_{\mathbf{x}\in\mathcal{V}(\mathbf{q}_0^i)} ||\mathcal{I}_{j+1}(\pi(\mathsf{H}_{0\to j+1}(\mathbf{n})\tilde{\mathbf{x}})) - \mathcal{I}_0(\mathbf{x})||^2 \tag{8.3}$$

où π est la fonction permettant de passer des notations homogènes aux coordonnées euclidiennes (voir la section 2.1). $\mathcal{I}_j$ est l'image capturée par la $j$ème caméra observant le point 3D considéré. $\mathbf{q}_0^i$ est l'observation de ce point par la première caméra qui le voit. $\mathcal{V}(\mathbf{q}_0^i)$ est un voisinage autour de cette observation (en pratique une fenêtre carrée 25×25 centrée sur ce point).

Une optimisation non-linéaire permet alors de minimiser cette erreur de transfert, la dernière estimation de la normale (*i.e.* $\mathbf{n}_j$) étant utilisée comme initialisation de cette minimisation. La normale étant un vecteur de dimension 3 normalisé, seuls deux paramètres sont à optimiser. En pratique, la normale raffinée $\mathbf{n}_{j+1}$ est définie comme étant la dernière estimation de la normale $\mathbf{n}_j$ à laquelle on ajoute un incrément $\mathbf{n}_{inc}$, avec $\mathbf{n}_{inc} = \alpha\mathbf{n}_x + \beta\mathbf{n}_y$ (figure 8.5). La normale $\mathbf{n}_{j+1}$ peut alors s'écrire :

$$\begin{aligned} \mathbf{n}_{j+1} &= \mathbf{n}_j + \mathbf{n}_{inc} \\ &= \mathbf{n}_j + (\alpha\mathbf{n}_x + \beta\mathbf{n}_y) \end{aligned} \tag{8.4}$$

(α, β) deviennent alors les seuls paramètres à estimer :

$$(\alpha, \beta) = \underset{(\alpha,\beta)}{argmin} \sum_{\mathbf{x}\in\mathcal{V}(\mathbf{q}_0^i)} ||\mathcal{I}_{j+1}(\pi(\mathsf{H}_{0\to j+1}(\mathbf{n}_i + \mathbf{n}_{inc})\tilde{\mathbf{x}})) - \mathcal{I}_0(\mathbf{x})||^2 \tag{8.5}$$

où

$$\mathsf{H}_{0\to j+1}(\mathbf{n}_j + \mathbf{n}_{inc}) \sim \mathsf{K}(\mathsf{R}_{0\to j+1} - \frac{\mathbf{t}_{0\to j+1}(\mathbf{n}_j + \mathbf{n}_{inc})^\mathsf{T}}{\mathcal{Q}^\mathsf{T}(\mathbf{n}_j + \mathbf{n}_{inc})})\mathsf{K}^{-1} \tag{8.6}$$

A chaque nouvelle caméra clé, l'équation 8.5 doit être résolue pour l'ensemble des points 3D qu'elle observe. Afin d'assurer un temps de traitement minimal, la méthode de résolution par *composition inverse* proposée par Baker and Matthews (2004) est utilisée. L'avantage de cette approche est que la jacobienne du problème à résoudre est constante au cours de la minimisation, ce qui implique que son calcul n'a besoin d'être réalisé qu'une seule fois.

8.3.2.4 Ambiguïté sur le sens de la normale

L'homographie $\mathcal{H}(\mathbf{n})$ étant définie à un facteur près, il y a une ambiguïté sur le sens de la normale associée aux points 3D. Dans les travaux de Molton et al. (2004) par exemple, les patchs sont très souvent observés de façon fronto-parallèle. La normale initiale $\mathbf{n}_0$ est alors

suffisamment proche de la solution réelle pour que le sens de la normale soit correctement défini. Cependant, dans le contexte qui nous intéresse, les patchs plans observés sont situés sur les bâtiments le long de la route et sont donc quasiment tangents à l'axe optique des caméras. Ceci revient à dire que la normale réelle du patch est souvent perpendiculaire à l'axe optique de la caméra. Dans ce cas de figure, la normale initiale $\mathbf{n}_0$ peut donc être très éloignée de la solution réelle. En pratique, il arrive alors souvent que cette mauvaise estimation de $\mathbf{n}_0$ entraîne une erreur sur le sens de la normale.

Or dans notre cas, le sens de la normale est important puisque c'est cette information qui nous permettra de différencier les plans auxquels seront associés les points 3D reconstruits. Ainsi, après chaque raffinement d'une normale, nous utilisons une méthode visant à déterminer le sens de celle-ci. Nous savons que les points que nous cherchons particulièrement à reconstruire sont sur des façades perpendiculaires à l'axe optique de la caméra mobile. Ceci implique que plus la caméra avance, plus le rayon optique issu de la caméra et passant par le point 3D est proche de la normale de ce point. A chaque nouvelle caméra clé observant le point $\mathcal{Q}^i$, le sens de la normale est choisi comme étant celui qui rend négatif le produit scalaire entre la normale et le rayon optique issu de la caméra et passant par ce point (figure 8.6) :

$$\overrightarrow{\mathcal{C}_j\mathcal{Q}^i}^{\mathsf{T}}\,\mathbf{n} < 0 \tag{8.7}$$

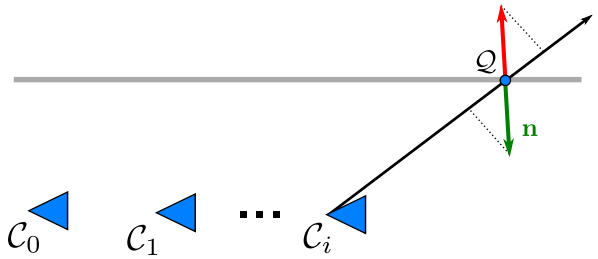


FIGURE 8.6 – **Ambiguïté sur le sens de la normale.** La méthode de validation proposée permet de fixer le sens des normales reconstruites. La normale verte est la normale retenue alors que la normale rouge est la normale rejetée par notre critère.

8.3.2.5 Reconstruction obtenue

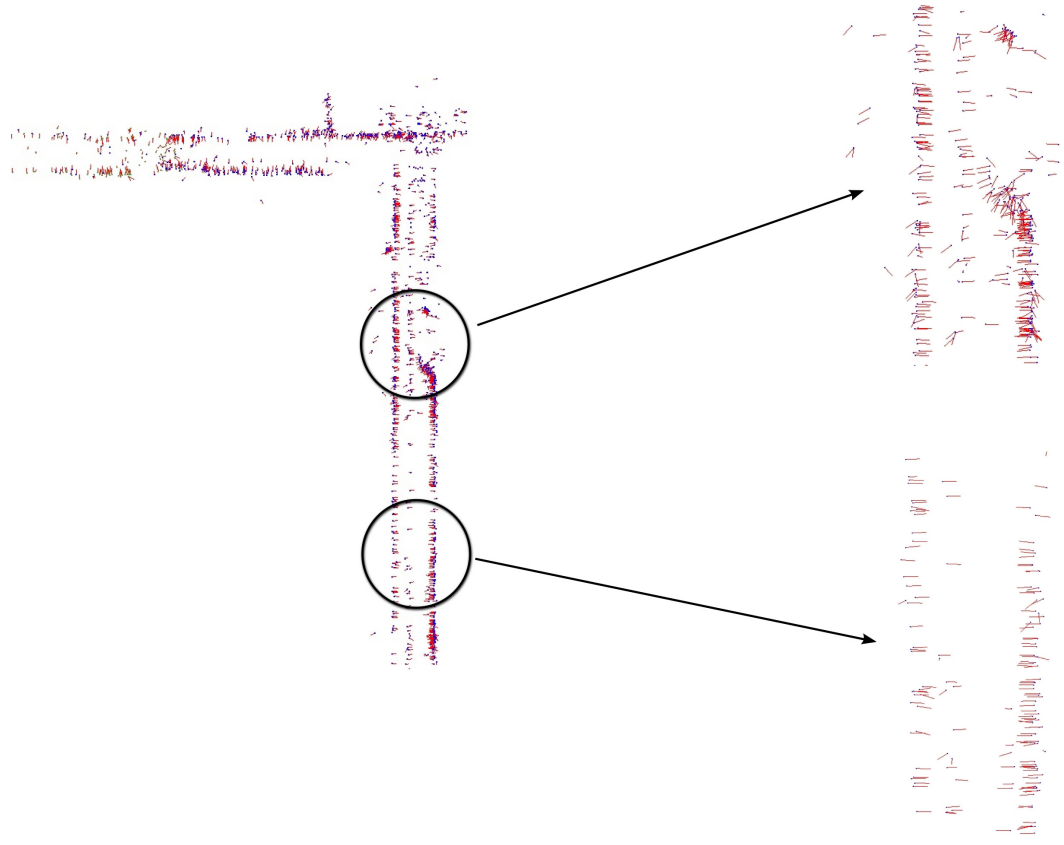
La figure 8.7(a) illustre un exemple de nuage de points reconstruits associés à leur normale ainsi estimée. Comme le montre cette figure, les normales reconstruites sont bruitées. Néanmoins, nous pouvons constater que ces normales sont globalement cohérentes avec la scène réelle reconstruite. Nos verrons en particulier dans la suite de ce chapitre que la précision obtenue sur ces normales est suffisante pour différencier efficacement les différents plans du modèle 3D.

8.3.3 Recalage par ICP

Les normales des points 3D reconstruits étant désormais calculées, la méthode ICP peut être utilisée pour recaler le nuage de points reconstruits et le modèle 3D. Tout d'abord, il sera montré que l'association des données peut être améliorée grâce à la connaissance de ces normales. Dès



(a)



(b)

FIGURE 8.7 – **Exemple de normales reconstruites.** Les normales reconstruites (b) sont généralement bruitées mais restent cohérentes avec la scène réelle (a).

lors, nous montrerons que le recalage optimal peut être trouvé en minimisant la métrique relative à ces associations.

8.3.3.1 Association des données

Contrairement au chapitre 3, le recalage entre les points reconstruits et le modèle 3D ne se fait pas sur l'ensemble de la reconstruction SLAM mais uniquement sur les données caractérisant le virage qui est en train d'être parcouru. Pour cela, nous prenons en compte dans l'ICP la sous-reconstruction SLAM constituée des 20 dernières caméras clés situées avant le virage et de toutes les caméras clés reconstruites après le virage, la notion d'avant et après virage étant définie par la polygonalisation de la trajectoire (section 8.2).

Comme nous l'avons vu précédemment (section 8.3.1), l'association point-plan au plus proche n'est plus optimale lorsque l'alignement est réalisé en ligne. Pour gagner en robustesse, l'idée principale est de permettre l'association entre un point et un plan uniquement s'ils possèdent tous deux des normales dont l'orientation est proche. Ainsi, le plan $\mathbf{\Pi}_{h_i}$ associé au point $\mathcal{Q}^i$ est désormais :

$$\mathbf{\Pi}_{h_i} = \underset{\mathbf{\Pi}_j \in \mathcal{M}^*}{argmin}\, d(\mathcal{Q}^i, \mathbf{\Pi}_j) \tag{8.8}$$

où $\mathcal{M}^*$ est le sous-ensemble de plans de $\mathcal{M}$ dont les normales sont cohérentes avec celle de $\mathcal{Q}^i$ et d est la distance orthogonale entre un point 3D et les plans de $\mathcal{M}^*$. En pratique, la normale d'un plan est dite cohérente avec celle de $\mathcal{Q}^i$ si l'angle entre ces deux vecteurs est inférieur à $\pi/4$.

Notons que l'étape d'association réalisée ici n'est validée que si au moins N points 3D (dans nos expériences 150) sont effectivement associés à un plan (*i.e.* si leur normale est cohérente avec au moins un plan du modèle) avant et après le virage. En effet, dans le cas contraire, trop peu de données sont utilisées pour assurer la robustesse de l'alignement. La méthode de recalage n'est alors pas lancée immédiatement mais sera retentée à la caméra clé suivante.

8.3.3.2 Minimisation de l'erreur associée

Une fois l'association des données réalisée, il est nécessaire de définir la métrique à minimiser. Celle-ci est très proche de la métrique définie dans le chapitre 3. Deux différences notables sont cependant à prendre en compte :

- ▷ **Réduction du nombre de paramètres.** A des fins d'amélioration de la robustesse, le nombre de paramètres à estimer ici est limité par rapport à l'étude du chapitre 3. Pour cela, la sous-reconstruction SLAM prise en compte dans le recalage est considérée comme étant rigide. Ainsi, seuls 3 paramètres sont pris en compte : (X, Y), le déplacement 2D dans le plan horizontal et ψ, l'orientation autour du vecteur orthogonal au sol.

- ▷ **Nouvelle normalisation des données.** Il a été défini dans la section 3.3 qu'il était nécessaire de normaliser les résidus des points de façon à donner à chaque fragment le même poids dans la minimisation. Dans le présent cadre d'étude, nous ne normaliserons plus les résidus par rapport au fragment auquel ils sont liés (puisqu'un seul fragment est considéré ici) mais par rapport aux plans du modèle.

Dès lors, le problème à résoudre est de trouver les paramètres (X, Y, ψ) :

$$(X, Y, \psi) = \underset{(X,Y,\gamma)}{argmin} \sum_i \rho^{\dagger}_{h_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})) \tag{8.9}$$

$\rho^{\dagger}_{h_i}$ désigne le M-estimateur associé au plan Π_{h_i}, à savoir :

$$\rho^{\dagger}_{h_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})) = \frac{\rho_{h_i}(d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i}))}{\max\limits_{\mathcal{Q}^j \in \mathcal{N}_{h_i}} \rho_{h_i}(d(\mathcal{Q}^j, \mathbf{\Pi}_{h_i})) \times card(\mathcal{N}_{h_i})} \tag{8.10}$$

où $\mathcal{N}_{h_i}$ est l'ensemble des points 3D associés au plan $\mathbf{\Pi}_{h_i}$, $card(\mathcal{N}_{h_i})$ est son cardinal et ρ_{h_i} le M-estimateur de Tukey dont le seuil est calculé sur les résidus de $\mathcal{N}_{h_i}$ (grâce au MAD). L'équation 8.9 est alors en pratique résolue grâce à l'algorithme de Levenberg-Marquardt (Levenberg (1944)).

8.3.3.3 Itération de l'ICP

Dans les méthodes ICP classiques (voir section 3.1), les étapes d'association de données et de minimisation de la métrique sont itérées de façon à permettre aux points 3D de changer le plan auquel ils sont associés. Néanmoins, il apparaît que dans notre cadre d'étude une seule itération est suffisante pour atteindre le recalage recherché. Ceci s'explique à la fois par le fait que l'initialisation de l'ICP est en règle générale proche de la solution (grâce à la correction du facteur d'échelle) et que l'association des données est réalisée de façon robuste grâce à l'information apportée par les normales. La majorité des points 3D reconstruits sont donc généralement liés dès la première association au plan qui leur correspond dans la réalité. Enfin, les points aberrants restant sont automatiquement rejetés de part l'utilisation du M-estimateur.

Dans la section suivante, nous allons montrer comment les bâtiments 3D peuvent être remplacés par une carte de la route, en particulier hors des villes.

8.4 Estimation de la dérive à l'aide d'une carte de la route

Dans cette section, nous nous plaçons sous l'hypothèse qu'aucun modèle 3D n'est disponible pour la zone parcourue (par exemple en dehors des villes). Nous montrons alors que le recalage effectué entre le nuage de points et le modèle 3D peut être remplacé par un recalage entre les caméras clés reconstruites et la route.

8.4.1 Aperçu de la méthode

La carte simple de la route utilisée est considérée comme étant un ensemble de segments 3D. Afin de réaliser le recalage entre la trajectoire reconstruite et la carte de la route, l'idée est alors de remplacer la distance entre les points 3D reconstruits et le modèle 3D utilisée à la section 8.3.3) par une nouvelle distance entre les caméras clés reconstruites et la carte de la route.

Néanmoins, il est à noter que les données utilisées (à savoir les caméras clés reconstruites et la carte de la route) sont nettement moins bruitées que lors de l'utilisation des bâtiments 3D. En particulier, nous savons que toutes les caméras reconstruites sont situées sur la route. Dès lors, comme nous l'avons précisé à la section 8.2, l'information issue de la carte pourra être utilisée dans deux contextes différents :

▷ **Pour estimer la dérive d'orientation dans les lignes droites**. La qualité des données utilisées permet d'estimer la dérive de cap de la reconstruction, c'est à dire son orientation autour de la verticale, au cours des lignes droites. Ceci permet alors d'arriver au niveau des virages avec l'erreur la plus faible possible.

▷ **Pour estimer la dérive de position dans les virages.** A l'instar de la méthode s'appuyant sur le modèle 3D, la carte de la route permet d'estimer l'erreur d'accumulation en position au moment des virages.

Une fois la distance utilisée définie, la suite de cette section détaillera les méthodes d'alignement évoquées ci-avant entre la reconstruction SLAM et la carte de la route.

8.4.2 Distance utilisée

La métrique recherchée a pour but de mesurer si la trajectoire calculée (*i.e.* l'ensemble des caméras clés reconstruites) est cohérente avec la carte de la route. La carte étant décrite comme un ensemble de segments 3D, l'idée la plus classique serait de retenir la distance *au plus proche*, c'est à dire la distance orthogonale d entre la caméra et le segment de route qui lui est le plus proche (figure 8.8(a)) :

$$d(\mathcal{C}_i, \mathcal{R}) = \min_j d(\mathcal{C}_i, \boldsymbol{D}_j) \tag{8.11}$$

où d est la distance orthogonale, $\mathcal{R}$ la carte de la route et $(\boldsymbol{D}_j)_j$ l'ensemble des segments qui la composent. Néanmoins, même si cette distance est efficace lorsque la reconstruction SLAM est proche de sa position réelle, elle n'est plus pertinente en cas de dérive importante (figure 8.8(a)). Remarquons que ces conclusions sont équivalentes à celles précédemment réalisées dans la section 8.3.1 sur l'utilisation de la distance orthogonale pour l'association entre les points reconstruits et le modèle 3D de ville.

Pour pallier ces problèmes de mauvaises associations entre caméras et segments de route, la distance au segment le plus proche est remplacée par une distance au segment *le plus probable*. Nous définissons donc une nouvelle distance caméra-segment notée $d_T(\mathcal{C}_i, \boldsymbol{D}_j)$. Celle-ci correspond à la distance entre la caméra $\mathcal{C}_i$ et le segment de carte $\boldsymbol{D}_j$ le long de la perpendiculaire à la direction de la trajectoire reconstruite au niveau de la caméra $\mathcal{C}_i$ (figure 8.8(b)). La direction de la trajectoire pour la caméra $\mathcal{C}_i$ est définie comme étant le vecteur joignant les caméras $\mathcal{C}_{i-1}$ et $\mathcal{C}_{i+1}$. Cette nouvelle distance étant définie, la métrique retenue entre une caméra et la carte de la route s'exprime sous la forme :

$$d(\mathcal{C}_i, \mathcal{R}) = \min_j d_T(\mathcal{C}_i, \boldsymbol{D}_j) \tag{8.12}$$

avec

$$d_T(\mathcal{C}_i, \boldsymbol{D}_j) = +\infty \tag{8.13}$$

si la perpendiculaire à la trajectoire en la caméra $\mathcal{C}_i$ et le segment de carte $\boldsymbol{D}_j$ ne s'intersecte pas.

Comme l'illustre la figure 8.8(b), la nouvelle distance proposée permet de mieux prendre en compte la géométrie de la reconstruction. Ainsi, cette distance est plus robuste à une mauvaise position initiale de la reconstruction SLAM par rapport à la carte. De plus, afin de lever les ambiguïtés entre les différentes voies de circulation parallèles, la distance entre une caméra et un segment est fixée comme étant infinie si leurs sens de circulation respectifs ne sont pas les mêmes. En pratique, on vérifie pour cela que le produit scalaire entre la direction de la trajectoire au niveau la caméra considérée et celui de la route est positif.

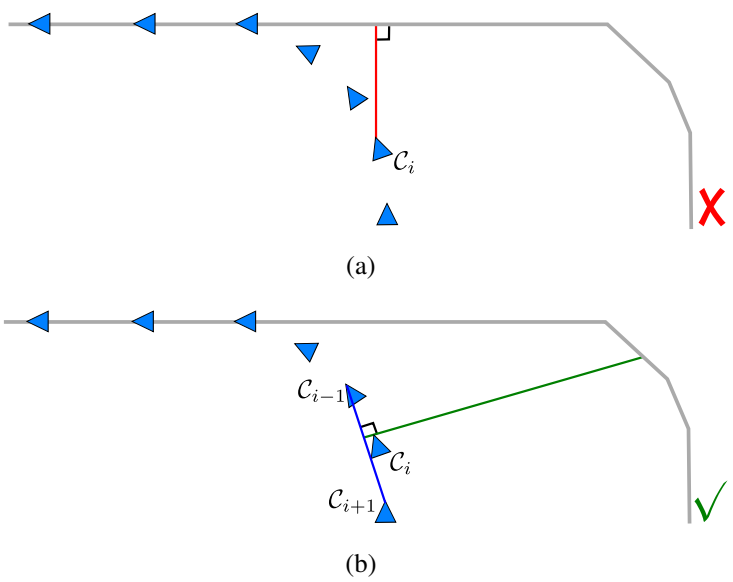


FIGURE 8.8 – **Distance utilisée entre la reconstruction SLAM et la route.** La distance orthogonale à la route (a) peut amener à de mauvaises associations. La distance prenant en compte la direction de la trajectoire (b) permet d'y pallier.

8.4.3 Estimation de la dérive en orientation dans les lignes droites

Comme nous l'avons vu à la section 8.2, il est possible d'estimer l'erreur accumulée en orientation au cours des lignes droites grâce à l'information apportée par une carte de la route. Pour cela, on recherche l'angle de la rotation autour de la verticale (*i.e.* l'angle de lacet) qui permet de recaler au mieux la trajectoire reconstruite sur la route, c'est à dire qui minimise la distance d définie dans l'équation 8.12. Pour des raisons de temps de traitement, l'estimation de la dérive en orientation n'est pas effectuée à chaque nouvelle image clé. De plus, seul un sous-ensemble de la reconstruction est pris en compte. La fonction $\mathcal{F}$, minimisée grâce à l'algorithme de Levenberg-Marquardt (Levenberg (1944)), est alors :

$$\mathcal{F}(\psi) = \sum_{i=N-M+1}^{N} \rho_T(d_T(\mathcal{S}(\psi, \mathcal{C}_i), \mathcal{R})) \tag{8.14}$$

où $\mathcal{S}(\psi, \mathcal{C}_i)$ est la rotation de la caméra $\mathcal{C}_i$ d'angle ψ autour de l'axe gravité passant par le centre de la caméra médiane (*i.e.* $\mathcal{C}_{(2N-M+1)/2}$). ρ_T est le M-estimateur de Tukey dont le seuil est fixé automatiquement grâce au MAD. N est l'indice de la caméra clé courante et M le nombre de caméras clés prises en compte dans l'optimisation. En pratique, nous avons réalisé nos expériences en optimisant le cap de la reconstruction toutes les 20 caméras clés sur les 20 dernières caméras clés reconstruites ($M = 20$).

Il est important de noter que, le cap ayant pu dériver depuis sa dernière correction, optimiser uniquement le lacet de la reconstruction en cours n'est pas suffisant. En effet, en plus de cette rotation, une translation est également à appliquer au sous-ensemble de caméras considéré dans l'optimisation (figures 8.9(a) et 8.9(b)). Cependant, étant donné le problème d'aperture mis en avant à la section 8.2, cette translation ne peut pas être optimisée avec l'angle de lacet durant la minimisation de $\mathcal{F}$. Nous avons donc choisi d'appliquer avant l'optimisation de l'angle de lacet la translation qui permet de placer le centre de rotation (*i.e.* la caméra médiane $\mathcal{C}_{(2N-M+1)/2}$) sur la route. Afin d'être le plus cohérent possible avec la trajectoire reconstruite, cette translation

est effectuée le long de la perpendiculaire à la trajectoire reconstruite au niveau de la caméra médiane (figure 8.9(a)). L'ensemble du traitement réalisé est résumé dans la figure 8.9.

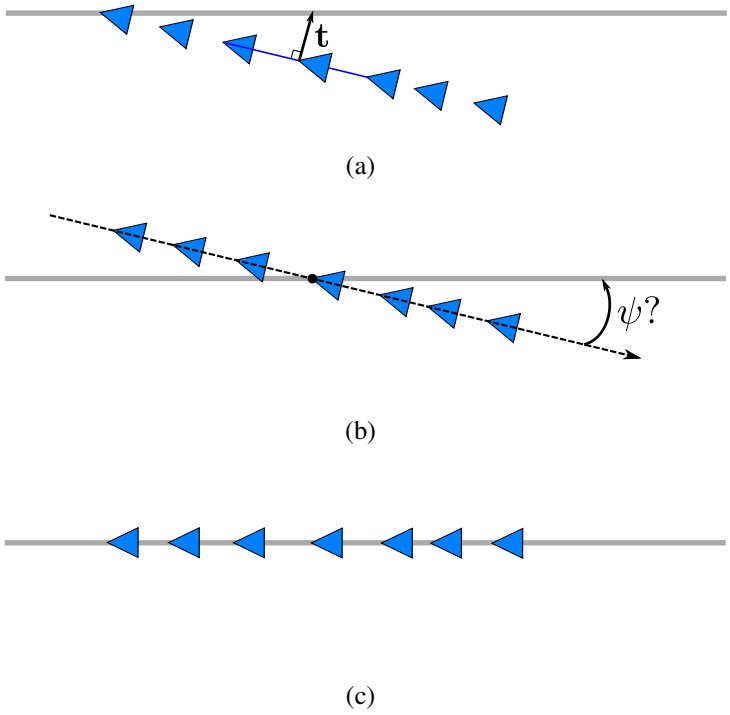


FIGURE 8.9 – **Les étapes de la correction du cap de la reconstruction.** Pour corriger le cap d'une reconstruction SLAM (a), deux étapes sont nécessaires : corriger la translation (b) puis corriger l'orientation (c).

Notons qu'en pratique, nous n'utilisons plus ici l'approche ICP. En particulier, les étapes d'association des données et de minimisation ne sont plus distinguées. En effet, très peu de résidus ont besoin d'être calculé ici. Nos expériences ont montré qu'il était alors plus efficace (en terme de temps de traitement) de remettre en cause l'association des données au sein même de l'optimisation.

8.4.4 Estimation de la dérive dans les virages

A l'instar de l'approche qui a été utilisée lors de l'exploitation des modèles 3D des bâtiments (section 8.3.3), les virages présentent suffisamment de contraintes géométriques pour permettre d'estimer l'erreur accumulée sur la position de la caméra. En effet, lorsqu'un virage est détecté dans la reconstruction courante (voir section 8.2), cette erreur peut être estimée en cherchant la transformation euclidienne qui recale au mieux les dernières caméras clés reconstruites sur la route. Comme dans la section 8.3.3, la transformation recherchée est limitée à 2 dimensions. Les paramètres à estimer ici sont à la fois l'angle de lacet ψ et la position 2D (X, Y) dans le plan horizontal. La fonction de coût $\mathcal{F}$ à minimiser est donc la suivante :

$$\mathcal{F}(X, Y, \psi) = \sum_{i=N-M+1}^{N} \rho_{l_i}(d_T(\mathcal{S}(\{X, Y, \psi\}, \mathcal{C}_i), \mathcal{R})) \tag{8.15}$$

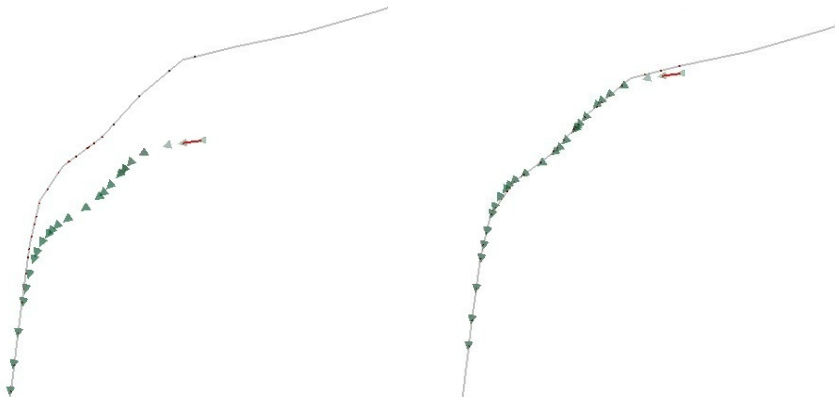
où $\mathcal{S}(\{X, Y, \psi\}, \mathcal{C}_i)$ est la rotation de la caméra $\mathcal{C}_i$ d'angle ψ autour de l'axe gravité passant par le centre de la caméra médiane (*i.e.* $\mathcal{C}_{(2N-M+1)/2}$) puis sa translation selon le vecteur (X, Y). N est l'indice de la caméra clé courante et M est le nombre de caméras considérées dans l'optimisation. Dans nos expériences, les caméras considérées sont les 20 dernières caméras situées avant le virage et l'ensemble des caméras situées après le virage. Notons que l'optimisation n'est lancé que si au moins 10 caméras ont été créées après le virage. Ceci a pour but d'avoir suffisamment de contraintes pour recaler la reconstruction SLAM sans ambiguïté.

ρ_{l_i} est le M-estimateur de Tukey dont le seuil est fixé grâce au MAD. Notons cependant que, de façon similaire aux observations faites sur l'alignement des points 3D avec le modèle (section 8.3), le seuil du M-estimateur n'est pas le même pour toutes les caméras. En effet, du fait de la dérive, les caméras après le virage sont généralement initialement beaucoup plus éloignées de la route que les caméras avant le virage (figure 8.10(a)). Ainsi, un seuil de Tukey sera utilisé pour les caméras avant le virage et un autre seuil sera utilisé pour les caméras après le virage. Dès lors, il sera nécessaire de normaliser les résidus afin que tous aient un poids similaire dans l'optimisation. Le M-estimateur finalement retenu est donc $\rho^{\dagger}_{l_i}$:

$$\rho^{\dagger}_{l_i} = \frac{\rho_{l_i}(d(\mathcal{S}(\{X, Y, \psi\}, \mathcal{C}_i), \mathcal{R}))}{\max\limits_{\mathcal{C}_j \in \mathcal{N}_{l_i}} \rho_{l_i}(d(\mathcal{S}(\{X, Y, \psi\}, \mathcal{C}_j) \times card(\mathcal{N}_{l_i})} \tag{8.16}$$

où $\mathcal{N}_{l_i}$ est l'ensemble des caméras appartenant au fragment l_i (*i.e.* ici avant ou après le virage) et $card(\mathcal{N}_{l_i})$ le cardinal de cet ensemble. La fonction $\mathcal{F}$ finalement optimisée est donc :

$$\mathcal{F}(X, Y, \psi) = \sum_{j=N-M+1}^{N} \rho^{\dagger}_{l_i}(d(\mathcal{S}(\{X, Y, \psi\}, \mathcal{C}_i), \mathcal{R})) \tag{8.17}$$



(a) (b)

FIGURE 8.10 – **Exemple de recalage dans les virages.** Le recalage dans les virages permet de corriger l'accumulation d'erreur en position.

Notons que, comme cela a été précisé pour l'estimation du cap dans les lignes droites (section 8.4.3), l'approche ICP n'est pas employée. En effet, du fait du faible nombre de résidus à calculer, il semble également ici plus efficace de remettre en cause l'association entre les caméras et les segments de la route directement au sein de l'optimisation.

Il a été montré dans les sections 8.3 et 8.4 que la transformation euclidienne relatant la dérive en accumulation d'erreur peut être ponctuellement estimée (totalement ou partiellement) à partir de la connaissance d'un SIG. Dans la section suivante, nous présenterons la façon dont est exploitée cette transformation afin de corriger en ligne la trajectoire estimée du véhicule.

8.5 Intégration de la nouvelle information

A l'issue des sections 8.3 et 8.4, nous disposons d'une estimation de l'erreur accumulée sous la forme d'une transformation euclidienne. Cette transformation euclidienne traduit le déplacement entre la position de la caméra estimée par le SLAM et la position estimée à l'aide du SIG. Cette transformation étant estimée à l'extérieur du processus SLAM, il est alors nécessaire de la réinjecter dans la reconstruction dans le but de corriger son erreur d'accumulation.

Comme cela était le cas pour l'intégration du facteur d'échelle (section 7.6), nous disposons donc pour la caméra courante de deux observations différentes pour la même pose de caméra. A l'instar de ce qui a été présentée dans cette section 7.6, les méthodes classiques de fusion de données sont difficilement utilisables de façon efficace dans notre contexte. Ceci est dû en particulier au fait qu'aucune covariance n'est associée à nos observations. Plus de détails à ce sujet peuvent être trouvés dans la section 7.6.

A partir de ce constat, nous avons décidé de faire entièrement confiance à l'estimation réalisée à l'aide du SIG. La pose retenue pour la caméra courante est donc la pose estimée par le SLAM à laquelle est appliquée la transformation euclidienne calculée précédemment. Néanmoins, pour le bon déroulement du SLAM, il est également nécessaire que l'historique des caméras clés reste cohérent avec la nouvelle position courante. Il est par conséquent nécessaire de définir la correction à apporter à chacun des éléments reconstruits (*i.e.* les caméras ainsi que les points 3D) de l'historique. Dans le but d'obtenir au final une trajectoire globalement cohérente, il serait donc bénéfique de définir un modèle de déformations reflétant au mieux la dérive d'accumulation d'erreur. Par exemple, on pourrait pour cela se baser sur le modèle élastique défini dans la section 3.2. Néanmoins, dans cette partie, l'information recherchée est avant tout la position courante de la caméra à chaque instant. La correction de l'historique n'a donc pas besoin d'être très réaliste. Cette correction doit uniquement permettre au processus de SLAM de pouvoir continuer l'estimation de la trajectoire. En ce sens, la méthode de correction que nous utilisons est très simple : l'ensemble de la reconstruction est considéré comme un ensemble rigide. La transformation euclidienne estimée est donc appliquée à chacun des éléments de la reconstruction (*i.e.* à toutes les caméras et points 3D reconstruits).

La méthode d'intégration présentée ci-avant est très simple et reste critiquable. En particulier, on peut penser qu'une correction plus fine de l'historique pourrait permettre d'améliorer la précision de la suite du processus SLAM. Néanmoins, la méthode retenue nous permettra de valider rapidement que l'utilisation d'un SIG peut permettre de réduire ponctuellement l'erreur accumulée au cours du processus.

8.6 Résultats expérimentaux

Nous allons présenter ici les résultats obtenus sur la méthode de localisation absolue complète (*i.e.* la correction du facteur d'échelle et de l'accumulation d'erreur). Après avoir détaillé le protocole expérimental suivi, nous présenterons alors les résultats obtenus pour l'utilisation des modèles 3D puis pour l'utilisation d'une carte de la route.

8.6.1 Protocole expérimental

Le facteur d'échelle utilisé dans les expériences présentées ici est corrigé grâce à la méthode proposée dans le chapitre 7. Les résultats obtenus sur la correction de l'accumulation d'erreur seront présentées sur deux séquences différentes. Ces séquences exploiteront respectivement le modèle 3D des bâtiments et une carte de la route. Notons qu'en pratique, nous ne disposons pas d'une telle carte. Afin d'en créer une, nous avons sous-échantillonné les données du trajectomètre de la séquence ODIAAC (séquence décrite à la section 7.5.1). Cette carte ainsi créée nous permettra de tester rapidement l'approche proposée. Néanmoins, au jour d'aujourd'hui, de telles cartes sont librement distribuées par exemple par la communauté OpenStreetMap [1].

La méthode de localisation absolue s'appuyant sur la connaissance des bâtiments 3D a été testée sur la sous-séquence de Versailles (décrite à la section 7.7.2.2) pour laquelle le calcul du facteur d'échelle est possible. En effet, nous possédons actuellement les données associées aux bâtiments 3D uniquement sur ce quartier de Versailles. Aucune vérité terrain n'étant associée à cette séquence, nous jugerons de la qualité de la reconstruction uniquement visuellement, en fonction de la cohérence de la trajectoire reconstruite par rapport au modèle 3D de l'environnement.

La méthode de localisation absolue s'appuyant sur une carte de la route a été testée sur la séquence ODIAAC (présentée en détail dans la section 7.5.1). Comme cela a été mis en avant précédemment, l'avantage principal de cette séquence est que le flux vidéo est associé à un trajectomètre. Ces données supplémentaires permettent d'obtenir une vérité terrain sur la position des caméras clés.

8.6.2 Résultats sur l'exploitation des bâtiments

La figure 8.11 résume les trajectoires obtenues pour les différentes étapes de la méthode proposée. La trajectoire représentée pour la méthode complète (figure 8.11(d)) est celle qui correspond à la localisation à chaque instant, c'est à dire sans correction *a posteriori* de l'historique. En première observation, nous pouvons remarquer que cette trajectoire est bien cohérente avec le parcours réel effectué (figure 8.11(a)). En particulier, la dérive d'accumulation qui est encore observable après la correction du facteur d'échelle (figure 8.11(c)) est cette fois corrigée à chacun des virages.

La trajectoire finale obtenue présente des discontinuités dans les virages qui sont le témoignage du recalage par ICP réalisé à ces endroits. Comme on peut le voir, ces recalages permettent d'annuler l'erreur de positionnement accumulée dans les différentes lignes droites. Il est important de noter que le seuil fixé sur le nombre de points associés nécessaires pour activer le recalage (section 8.3.3) agit directement sur la localisation obtenue. En effet, c'est ce seuil qui permet de régler le rapport entre la réactivité de la méthode (plus le seuil est faible, plus le recalage sera lancé tôt) et la robustesse de la méthode (plus le seuil est élevé, plus le nombre d'appariements nécessaires est important).

Ces résultats, ainsi que la superposition de la trajectoire obtenue avec l'image satellite de la zone concernée (figure 8.12), tendent à montrer que la méthode proposée permet de localiser de façon relativement précise le véhicule. En particulier, en annulant ponctuellement la dérive d'accumulation d'erreur, la méthode de recalage rend possible la localisation du véhicule sur des trajectoires de grande échelle.

1. www.openstreetmap.fr

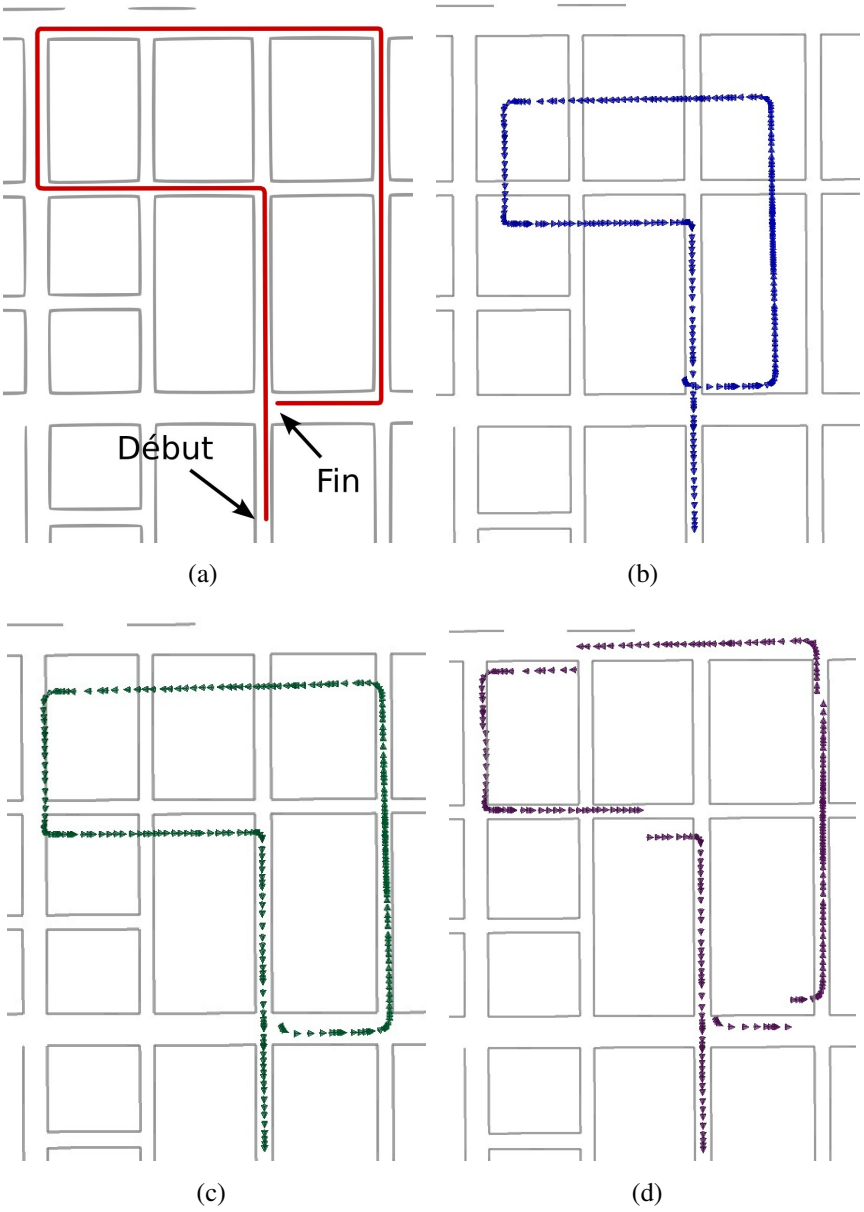


FIGURE 8.11 – **Les différentes étapes de la méthode proposée sur la séquence Versailles 1.** (a) est la trajectoire réelle. Les trajectoires sont celles obtenues avec (b) la méthode SLAM seule, (c) la méthode de correction du facteur d'échelle et (d) la méthode proposée complète.



FIGURE 8.12 – **Résultat de la localisation absolue obtenue sur la séquence Versailles 1.**

8.6.3 Résultats sur l'exploitation d'une carte de la route

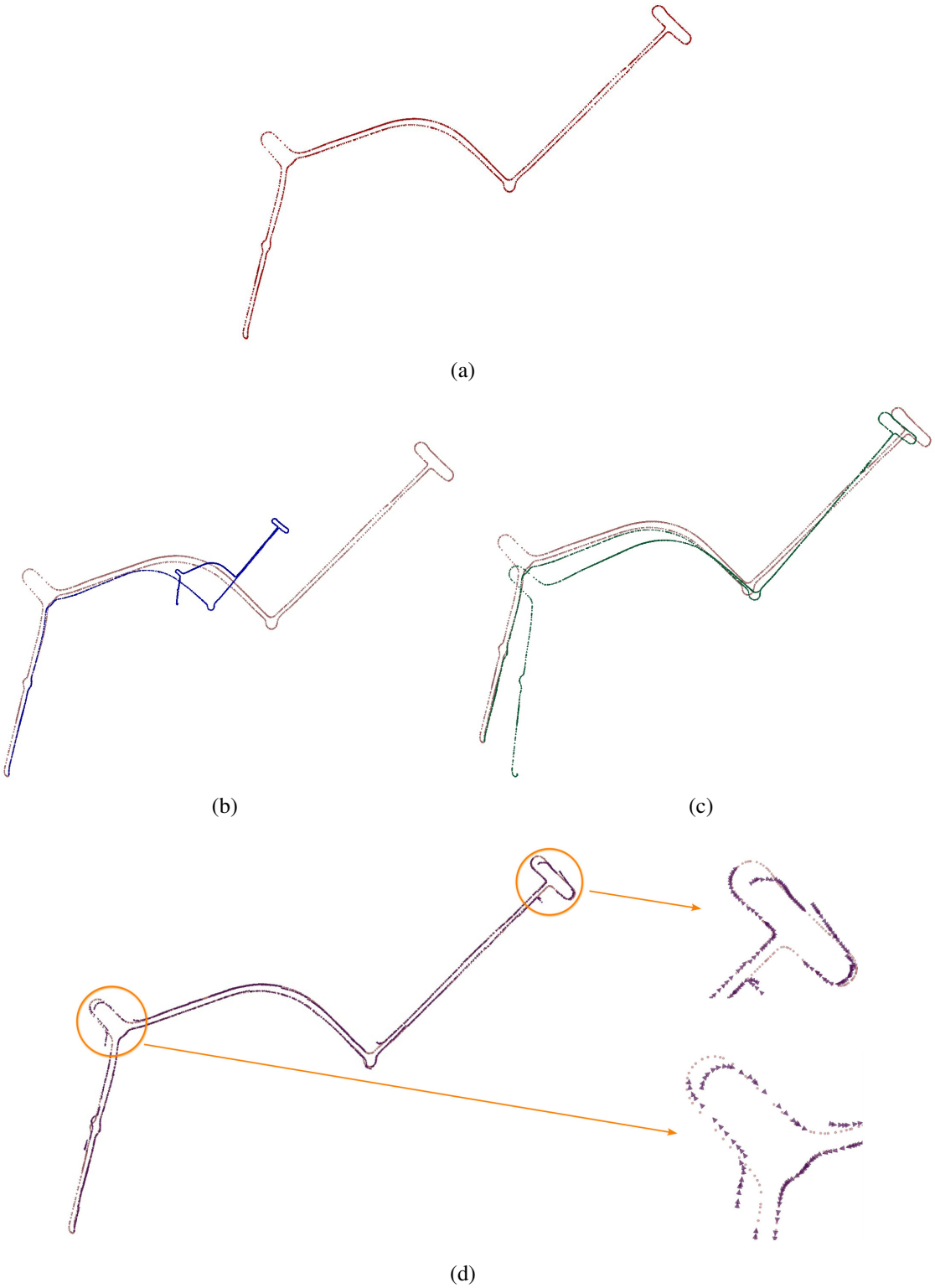
Les différentes étapes de la méthode proposée sont résumées dans la figure 8.13. Les figures 8.13(b) et 8.13(c) reprennent les résultats de la correction du facteur d'échelle détaillés dans la section 7.7.

La comparaison des figures 8.13(c) et 8.13(d) met clairement en avant l'intérêt de l'ajout de l'information absolue. Comme nous l'avions constaté dans la section 7.7.2.1, la méthode de correction du facteur d'échelle ne permet pas de localiser précisément une caméra mobile à grande échelle. En effet, l'erreur accumulée sur la position et le cap de la caméra mobile rend rapidement la localisation fortement erronée. Au contraire, la figure 8.13(d) met en avant que la méthode de recalage proposée permet de toujours rester cohérent avec la carte de la route. En particulier, les discontinuités dans la trajectoire obtenue correspondent aux recalages effectués dans les virages. Notons que, comme cela a été mis en avant pour la méthode de recalage sur les bâtiments (section 8.6.2), le seuil fixé sur le nombre minimal de caméras nécessaires après le virage pour lancer le recalage permet de régler le ratio entre la robustesse et la réactivité de la méthode.

La figure 8.14, ainsi que le tableau 8.1, confirment les résultats présentés. On peut en particulier y voir l'évolution de l'erreur absolue de positionnement de la caméra mobile au cours de la trajectoire. Il y apparaît clairement que les recalages dans les virages permettent de réduire notablement l'erreur accumulée dans les lignes droites (figure 8.14(b)). L'erreur de positionnement moyenne obtenue par la méthode complète proposée est d'environ 15 mètres, ce qui est de l'ordre de grandeur de la localisation obtenue par un capteur GPS. La superposition de la trajectoire reconstruite avec l'image satellite associée (figure 8.15) met en avant la cohérence et la précision de la trajectoire obtenue.

8.6.4 Temps de traitement

La méthode étant destinée à rendre possible la localisation d'un véhicule en temps réel, il est intéressant d'étudier si l'approche proposée peut atteindre une cadence suffisamment im-



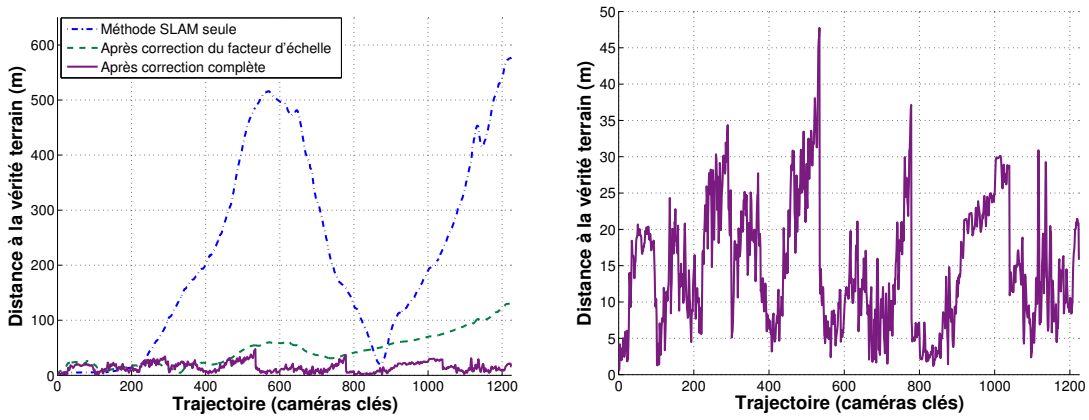
(a)

(b)

(c)

(d)

FIGURE 8.13 – **Les différentes étapes de la méthode proposée sur la séquence ODIAAC.** (a) est la vérité terrain. Les trajectoires sont celles obtenues avec (b) la méthode SLAM seule, (c) la méthode de correction du facteur d'échelle et (d) la méthode proposée complète. Chacune de ces reconstructions est superposée à la vérité terrain.



(a) (b)

FIGURE 8.14 – **Evolution de l'erreur de positionnement absolue.** (a) compare l'évolution de l'erreur de positionnement absolue pour la méthode SLAM seule, la méthode de correction du facteur d'échelle et la méthode de localisation proposée complète. (b) est le détail du résultat obtenu avec la méthode complète proposée.

	Localisation par SLAM seul	Localisation avec échelle corrigée	Localisation avec la méthode complète
Erreur moyenne sur le positionnement absolu (m)	241,6	51,2	**14,7**
Ecart-type (m)	196,9	36,1	**8,4**
Erreur médiane sur le positionnement absolu (m)	193,4	44,3	**13,5**

TABLE 8.1 – **Statistiques sur la localisation absolue.** Les statistiques sont calculées sur l'ensemble des caméras clés.

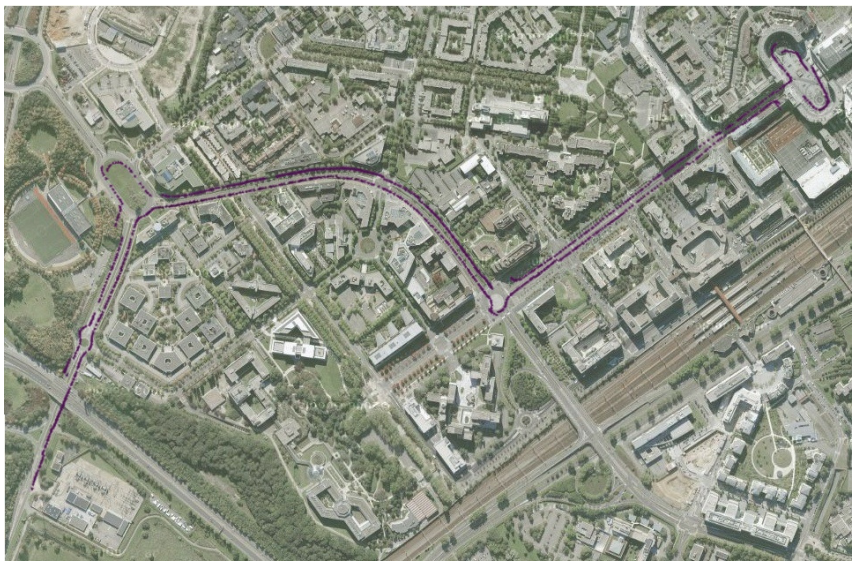


FIGURE 8.15 – **Résultat de la localisation absolue obtenue sur la séquence ODIAAC.**

portante pour permettre une localisation assez rapide. Nous avons expliqué dans la section 7.5 que les temps de traitement nécessaires à la correction du facteur d'échelle sont potentiellement suffisamment faibles pour que cette correction soit intégrée au processus en ligne.

Pour obtenir la méthode complète, il est également nécessaire d'incorporer au SLAM le processus de correction de l'accumulation d'erreur présenté dans ce chapitre. La totalité de ce traitement a été implémentée en C++, sans effort particulier sur l'optimisation du code écrit. En l'état, le processus de SLAM auquel est ajouté la méthode de correction présentée ici a un temps de traitement de l'ordre d'une demi-seconde par image clé, sachant qu'une image clé est créée en moyenne toutes les demi-secondes à 50 km/h. La majorité du temps de calcul est occupée par deux étapes :

- ▷ **Estimation du recalage.** A l'instar de ce qui a été fait dans la première partie, les dérivées de l'ICP sont calculées numériquement. Un calcul analytique de ces dérivées permettra donc d'accélérer considérablement le recalage.

- ▷ **Estimation des patchs orientés.** A chaque image clé, une cinquantaine de patchs sont observés. C'est donc une cinquantaine de normales qu'il est nécessaire de raffiner. En l'état, la méthode d'estimation des normales des patchs orientés possède un code optimisé. En particulier, la méthode d'alignement par composition inverse proposée par Baker and Matthews (2004) permet de réduire considérablement les temps de calcul. Néanmoins, le calcul des normales est entièrement effectué par le CPU. Or, une implémentation sur GPU permettrait d'accélérer fortement ce type de calcul.

Ainsi, même si elle ne l'est pas actuellement, il est raisonnable d'envisager que la méthode de localisation proposée dans cette partie puisse s'effectuer en temps-réel.

8.7 Discussion

Dans ce chapitre, nous avons avant tout souhaité montrer que la connaissance d'un SIG simple de l'environnement peut permettre de corriger la dérive inhérente aux méthodes SLAM. La méthode résultante permet ainsi de localiser un véhicule sur des séquences de plusieurs kilomètres.

En particulier, il a été illustré que le SIG permet d'estimer l'erreur d'accumulation en effectuant un recalage entre la reconstruction SLAM et ce SIG lorsque cela est possible. Les expériences ont montré que la méthode proposée complète, c'est à dire la correction du facteur d'échelle et la correction liée au SIG, permet de localiser la caméra mobile avec une précision de l'ordre de 15 mètres. Le système proposé peut donc par exemple être utilisé comme une alternative au système GPS classique dans les milieux denses où le masquage des signaux GPS est fréquent.

Néanmoins, en l'état, la méthode proposée possède encore des problèmes qu'il faudra résoudre pour assurer sa robustesse et améliorer sa précision. En particulier, il est important de remarquer que dans l'approche proposée, l'erreur de positionnement n'est pas uniformément répartie sur la trajectoire. En effet, l'erreur s'accumule dans les lignes droites avant d'être fortement réduite dans les virages (pour atteindre moins de 3 mètres d'erreur dans ces endroits). Les statistiques sur la localisation absolue du véhicule vont donc être fortement dépendantes de la trajectoire parcourue. Ainsi, plus les lignes droites sont longues, plus l'accumulation d'erreur sera importante. La localisation de la caméra pourra alors être grandement erronée. Au contraire, plus la trajectoire comporte de virages et de courtes lignes droites, plus les statistiques seront bonnes. On peut en particulier raisonnablement penser que la localisation sur la séquence de Versailles a une erreur moyenne nettement inférieure à la séquence ODIAAC.

Une des perspectives directes de l'approche proposée serait donc de permettre une correction de l'accumulation d'erreur dans les lignes droites. On peut penser en particulier à utiliser les informations géométriques issues des carrefours traversés. L'utilisation de modèles de bâtiments plus précis pourrait certainement également apporter des contraintes géométriques exploitables dans les lignes droites.

Conclusion

Travaux réalisés

Notre objectif dans cette thèse a été de proposer une solution pour la géolocalisation par vision d'un véhicule sur des parcours de plusieurs kilomètres. En particulier, nous souhaitions montrer que l'information apportée par un Système d'Information Géographique peut être utilisée dans le but de corriger les dérives inhérentes aux méthodes de localisation et cartographie simultanées classiques. Nos travaux se sont articulés autour de deux approches distinctes présentant des avantages et inconvénients différents.

La première solution proposée consiste à construire hors ligne une base d'amers visuels géoréférencée à partir de laquelle il est alors possible de localiser une caméra mobile en temps-réel. Notre contribution a été de proposer un processus permettant de construire automatiquement une telle base de données à partir d'une reconstruction SLAM et d'un modèle 3D de l'environnement. En particulier, deux étapes ont été présentées : un ICP non-rigide, ayant pour but de corriger la cohérence globale de la reconstruction, puis l'ajustement de faisceaux ST-CBA visant à raffiner le résultat obtenu. La méthode a été expérimentée sur différentes séquences de synthèse et réelles. De plus, le concept général de localisation en ligne d'une caméra a été validé par un exemple d'application d'aide à la navigation.

Dans un deuxième temps, nous avons souhaité explorer la possibilité de corriger la reconstruction SLAM en ligne (c'est à dire au fur et à mesure du parcours) uniquement à partir de l'information géométrique extraite du SIG. Le fait de localiser la caméra à l'aide d'une information purement géométrique offre plusieurs avantages. Tout d'abord, ceci permet de se passer de la nécessité de construire et de maintenir à jour une base de données d'amers visuels décrivant l'environnement. De plus, puisque non basée sur de la mise en correspondance de données photométriques, la méthode est insensible aux conditions d'observation de la scène. Enfin, l'information utilisée n'a pas besoin de contenir d'information de texture, ce qui la rend plus compacte et par conséquent plus facile à embarquer. Afin de démontrer la faisabilité et l'intérêt de cette approche, nous avons proposé une méthode relativement simple mais fonctionnelle. Cette méthode est constituée de deux processus distincts. Tout d'abord, nous avons proposé une méthode d'estimation du facteur d'échelle qui utilise la connaissance de la distance entre la caméra et le sol. En particulier, nous avons montré qu'il est possible d'exploiter le mouvement estimé par le SLAM dans le but de rendre plus robuste le processus de calcul du facteur d'échelle. Dès lors, cette formalisation est utilisée afin de corriger en temps-réel la dérive en facteur d'échelle

de la méthode SLAM. Nous avons alors montré qu'il est possible de corriger ponctuellement la position courante de la caméra en exploitant le Système d'Information Géographique dont nous disposons. Pour cela, la reconstruction SLAM est recalée, lorsque cela est possible, sur un modèle 3D de l'environnement ou une carte de la route. La validation expérimentale a permis de conclure qu'une telle approche permet de géolocaliser un véhicule sur des trajectoires de plusieurs kilomètres.

Perspectives

Avant toute chose, la vocation de nos travaux a été de montrer l'intérêt d'exploiter un Système d'Information Géographique pour contraindre les méthodes de SLAM. Ceci implique que nos travaux sont avant tout des démonstrations de la faisabilité de cette idée. Néanmoins, les résultats obtenus sont encourageants et confirment l'intérêt d'une telle approche. Les études et expériences réalisées nous ont permis de mettre en évidence certaines perspectives directes de nos travaux :

- ▷ **Exploitation des points de l'environnement.** En l'état actuel, les points 3D reconstruits qui ne sont pas situés sur le modèle 3D (arbres, véhicules, *etc.*) ne sont pas pris en compte lors de l'ajustement de faisceaux ST-CBA. Il semble pourtant intéressant d'exploiter cette information. En effet, cela permettrait d'utiliser une information en plus grande quantité et mieux répartie dans l'image. Cela permettrait également de résoudre les problèmes liés à l'absence de bâtiments (section 5.3.2). Pour cela, il serait possible de s'inspirer de l'approche proposée par Vacchetti et al. (2004). La fonction de coût contiendrait alors à la fois les résidus liés à la nouvelle fonction de coût proposée pour les points du modèle et les résidus liés à la reprojection classique pour les autres points. Cependant, il sera alors nécessaire de résoudre les problèmes liés à la pondération de ces résidus afin de s'assurer que les deux types d'erreur aient le même poids dans la minimisation. Des tests récents réalisés au sein du laboratoire confirment que cette approche tend à améliorer la précision et la robustesse de l'ajustement de faisceaux.

- ▷ **Meilleure différenciation des points 3D reconstruits.** A la perspective présentée ci-dessus s'ajoute naturellement le problème de différenciation des points 3D, c'est à dire la façon dont on peut définir si un point 3D reconstruit est sur un bâtiment ou non dans la réalité. Pour cela, on peut par exemple penser à détecter dans les images les éléments généralement observés dans un milieu urbain (arbres, véhicules garés sur le bas-côté, routes, bâtiments, *etc.*). En particulier, Brostow et al. (2008) ont montré que la structure reconstruite par l'algorithme SLAM peut être utilisée afin de segmenter efficacement l'image courante. Dès lors, un point 3D pourrait être caractérisé par le type de zone dans laquelle se situent ses observations.

- ▷ **Utilisation de modèles 3D plus précis.** Comme cela a été présenté dans la section 2.6.2.3, les modèles 3D que nous utilisons sont très peu précis et très peu détaillés. Néanmoins, la qualité des modèles disponibles s'est très nettement améliorée ces dernières années. Les façades des bâtiments sont souvent plus détaillées. De plus, les différents bâtiments d'une même rue sont facilement différentiables et leur géométrie est raffinée. Dès lors, il sera important de s'interroger sur les avantages et les inconvénients liés à l'utilisation de tels modèles. On peut raisonnablement penser que l'information

fournie par ces nouveaux modèles apportera des contraintes supplémentaires, en particulier dans les lignes droites. Ainsi, les problèmes d'aperture que nous rencontrons (section 8.2) pourraient être évités. L'accumulation d'erreur encore importante que nous observons dans les lignes droites (section 8.7) serait alors corrigée. Néanmoins, on peut penser que l'utilisation de modèles 3D complexes amènera des temps de calcul et de transfert certainement très importants.

▷ **Etude des contraintes informatiques liées aux Systèmes d'Information Géographique.** Enfin, des problèmes d'implémentation seront également à considérer dans l'optique de fournir un produit final. En particulier, en l'état actuel, le Système d'Information Géographique est directement inclus dans la brique logicielle développée. Ainsi, nous avons pour l'instant masqué les problèmes liés à la communication avec le SIG. En particulier, il sera important de mesurer les temps de latences observés lors de l'interrogation d'un SIG. Il sera alors très certainement nécessaire de modifier l'approche retenue en fonction de cette latence.

ANNEXE A

Ajustement de faisceaux par combinaison linéaire des fonctions de coût

Dans cette première annexe est détaillée une fonction de coût permettant d'intégrer à l'ajustement de faisceaux classique la contrainte associée à la connaissance d'un modèle 3D de la scène. Ainsi, cette fonction de coût sera une combinaison linéaire de deux métriques traduisant respectivement la contrainte de reprojection des points 3D et la contrainte d'appartenance des points 3D au modèle.

Les travaux décrits dans ce chapitre ont donné lieu à une publication (Lothe et al. (2010b)).

A.1 Approches existantes

Lorsque deux métriques dépendantes $\mathcal{F}$ et $\mathcal{G}$ sont à minimiser simultanément, une approche souvent retenue est de minimiser leur somme. Néanmoins, en pratique, les fonctions $\mathcal{F}$ et $\mathcal{G}$ n'ont pas nécessairement le même ordre de grandeur (voire parfois la même unité). Ainsi, il est possible qu'une de ces deux métriques soit prépondérante dans la minimisation, ce qui empêche alors la convergence vers la solution recherchée. Dès lors, un facteur α est introduit de façon à pondérer ces deux fonctions (Horn and Schunck (1981); Chui and Rangarajan (2003); Modersitzki (2004); Pilet et al. (2005); Michot et al. (2010)). La fonction $\mathcal{F}_{CL}$ à minimiser est alors :

$$\mathcal{F}_{CL} = \mathcal{F} + \alpha \times \mathcal{G} \tag{A.1}$$

La principale difficulté lorsqu'on utilise ce type d'approches est de déterminer le facteur α qui fournit la solution optimale. Notons qu'ici, la notion d'optimalité se rapporte non pas à la valeur de α qui fournira le minimum absolu de $\mathcal{F}_{CL}$ (qui serait nécessairement $\alpha = 0$) mais à celle qui permettra d'obtenir la solution réellement recherchée. Déterminer α est d'autant plus problématique que la meilleure valeur de ce facteur est généralement fortement dépendante de la séquence traitée (Bartoli et al. (2008)). En pratique, ce paramètre est généralement fixé manuellement. Notons cependant que certaines méthodes tendent à l'estimer automatiquement. On peut citer notamment la validation croisée (Farenzena et al. (2008)) et plus récemment la

méthode basée sur les L-Curve (Michot et al. (2010)). Néanmoins, les temps de calcul nécessaires à l'estimation automatique de α deviennent rapidement extrêmement importants lorsque le nombre de paramètres à optimiser est élevé.

A.2 Utilisation dans notre cadre d'étude

Adapter cette approche à notre cadre d'étude revient à déterminer les fonctions $\mathcal{F}_{IP}$ et $\mathcal{G}$ telles que :

$$\mathcal{F}_{CL} = \mathcal{F}_{IP} + \alpha \times \mathcal{G} \tag{A.2}$$

où $\mathcal{F}_{IP}$ est la fonction de coût relative à la contrainte entre les points 3D et leurs observations dans les images. La fonction de coût $\mathcal{G}$ est liée quant à elle à la contrainte d'appartenance des points reconstruits au modèle 3D.

A.2.1 Contrainte d'appartenance au modèle 3D

La fonction $\mathcal{G}$ a pour but de traduire la contrainte d'appartenance des points 3D au modèle 3D des bâtiments. La distance d'un point 3D $\mathcal{Q}^i$ au modèle est définie comme étant la distance orthogonale d entre ce point 3D et le plan $\mathbf{\Pi}_{h_i}$ du modèle qui lui est le plus proche. Pour pondérer au mieux les deux métriques dans l'optimisation, il est préférable que chaque point 3D ait autant de résidus issus de la contrainte au mur que de l'erreur de reprojection. La distance entre le point 3D $\mathcal{Q}^i$ et $\mathbf{\Pi}_{h_i}$ sera donc prise en compte pour chacune des observations de $\mathcal{Q}^i$. La fonction $\mathcal{G}$ retenue est donc :

$$\mathcal{G}(\mathcal{Q}^1, \ldots, \mathcal{Q}^M) = \sum_{1 \leq j \leq N} \sum_{i \in \mathcal{A}_j} d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})^2 \tag{A.3}$$

où N est le nombre de caméras reconstruites et $\mathcal{A}_j$ l'ensemble des indices des points observés par la caméra j.

A.2.2 Critère de cohérence dans les images

La fonction $\mathcal{F}_{IP}$ a pour but de mesurer l'erreur de reconstruction effectuée sur le point 3D $\mathcal{Q}^i$ par rapport à ses observations dans les images. Nous avons vu que l'erreur classiquement utilisée pour cela est l'erreur de reprojection (section 2.3.2.5). Néanmoins, l'erreur de reprojection est une erreur 2D. Cette erreur est donc incompatible avec l'erreur liée à $\mathcal{G}$. En effet, pour garantir la cohérence de la fonction $\mathcal{F}_{CL}$, il est préférable que les fonctions $\mathcal{F}_{IP}$ et $\mathcal{G}$ aient la même unité de mesure.

En ce sens, Ramalingam et al. (2006) ont proposé de mesurer l'erreur de reconstruction du point $\mathcal{Q}^i$ comme étant la distance entre ce point 3D et chacun des rayons optiques issus de ses observations dans les images. Notre approche est de remplacer l'utilisation du rayon optique par les plans d'interprétation $\mathbf{\Pi}_x^{i,j}$ et $\mathbf{\Pi}_y^{i,j}$. Ces plans sont deux plans particuliers qui correspondent à la rétro-projection des droites parallèles aux axes de l'image et passant par l'observation considérée (figure A.1).

Dès lors, la fonction de coût $\mathcal{F}_{IP}$ relative à la contrainte apportée par les images peut s'écrire :

$$\mathcal{F}_{IP}(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E, \mathcal{Q}^1, \ldots, \mathcal{Q}^M) = \sum_{1 \leq j \leq N} \sum_{i \in \mathcal{A}_j} (d(\mathcal{Q}^i, \mathbf{\Pi}_x^{i,j})^2 + d(\mathcal{Q}^i, \mathbf{\Pi}_y^{i,j})^2) \tag{A.4}$$

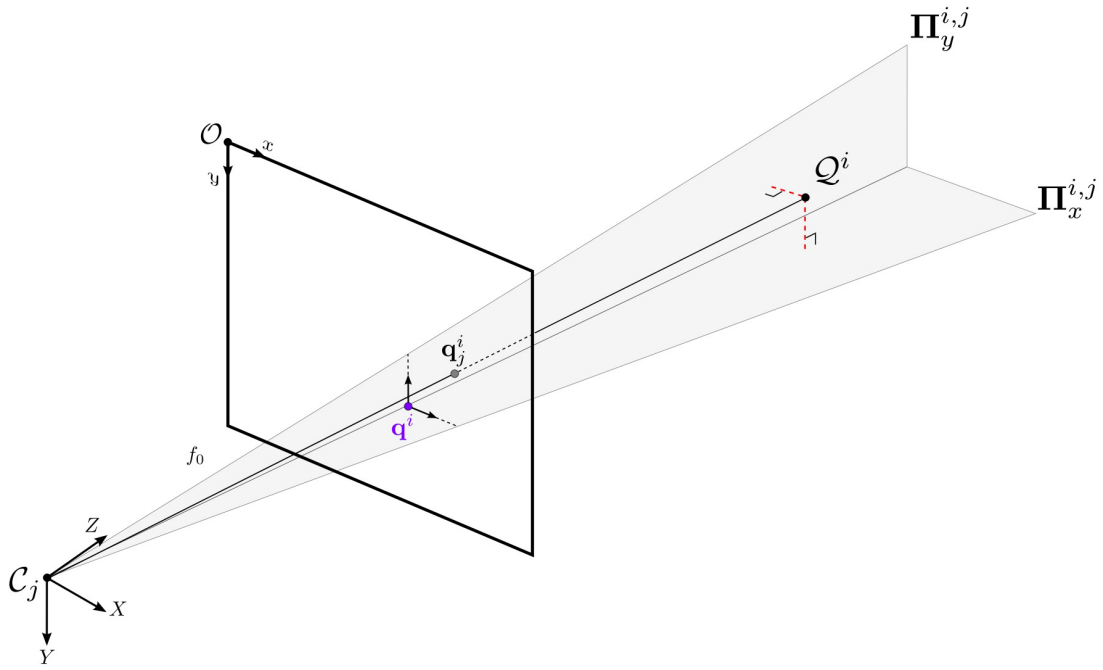


FIGURE A.1 – **Plans d'interprétation et résidus 3D.** Les deux résidus sont les distances entre le point $\mathcal{Q}^i$ et les plans $\mathbf{\Pi}_x^{i,j}$ et $\mathbf{\Pi}_y^{i,j}$.

où C_j^E sont les paramètres intrinsèques de la caméra $\mathcal{C}_j$.

A.2.3 Métrique utilisée et optimisation

Les deux fonctions $\mathcal{F}_{IP}$ et $\mathcal{G}$ étant précisées, il nous est maintenant possible de définir la fonction de coût $\mathcal{F}_{CL}$ retenue. Nous détaillerons alors l'algorithme permettant de la minimiser de façon robuste.

A.2.3.1 Fonction de coût retenue

Grâce à cette nouvelle expression de l'erreur de reprojection, chaque observation du point $\mathcal{Q}^i$ est liée à trois plans différents de l'espace : le plan d'interprétation $\mathbf{\Pi}_x^{i,j}$, le plan d'interprétation $\mathbf{\Pi}_y^{i,j}$ et le plan $\mathbf{\Pi}_{h_i}$ du modèle le plus proche de $\mathcal{Q}^i$. La fonction de coût $\mathcal{F}_{IP}$ peut donc s'écrire comme étant :

$$\mathcal{F}_{CL}(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E, \mathcal{Q}^1, \ldots, \mathcal{Q}^M) = \sum_{1 \leq j \leq N} \sum_{i \in \mathcal{A}_j} \left(d(\mathcal{Q}^i, \mathbf{\Pi}_x^{i,j})^2 + d(\mathcal{Q}^i, \mathbf{\Pi}_y^{i,j})^2 + \alpha \, d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i})^2 \right) \tag{A.5}$$

où α est le scalaire permettant de pondérer les duex critères. Dans notre cas, sa valeur sera fixée manuellement (voir section A.3).

A.2.3.2 Robustesse de l'optimisation

Pour assurer la convergence vers le minimum recherché, il est nécessaire d'être robuste à la fois aux points aberrants et aux mauvaises associations point-plan.

Robustesse aux points aberrants. Pour être robuste aux points aberrants (appariements 2D erronés, points 3D n'appartenant pas au modèle dans la réalité, *etc.*), le M-Estimateur de Tukey ρ_T est utilisé. Le seuil de ce M-estimateur est alors réglé automatiquement à l'aide du MAD. La fonction concrètement optimisée est alors :

$$\mathcal{F}_{CL}(\mathsf{C}_1^E, \ldots, \mathsf{C}_N^E, \mathcal{Q}^1, \ldots, \mathcal{Q}^M) = \sum_{1 \leq j \leq N} \sum_{i \in \mathcal{A}_j} \rho_T \left(d(\mathcal{Q}^i, \mathbf{\Pi}_x^{i,j}) + d(\mathcal{Q}^i, \mathbf{\Pi}_y^{i,j}) + \alpha \, d(\mathcal{Q}^i, \mathbf{\Pi}_{h_i}) \right) \tag{A.6}$$

Remise en cause des associations point-plan. A l'instar de l'ajustement de faisceaux proposé dans la section 4.2, il est nécessaire de remettre en cause au cours de l'optimisation l'association entre le point 3D $\mathcal{Q}^i$ et le plan qui lui est le plus proche. La méthode d'optimisation retenue consiste alors à itérer deux étapes :

- ▷ **Associations des données.** Chacun des points 3D reconstruits est associé au plan $\mathbf{\Pi}_{h_i}$ du modèle qui lui est le plus proche au sens de la distance orthogonale. Notons que cette association prend en compte le fait que les plans $(\mathbf{\Pi}_i)_i$ sont des plans finis. Ainsi, pour être associé au plan $\mathbf{\Pi}$, un point 3D $\mathcal{Q}$ doit avoir sa projection orthogonale à l'intérieur des limites de $\mathbf{\Pi}$. Les points qui ne sont associés à aucun plan sont alors simplement retirés de l'optimisation.

- ▷ **Minimisation de la métrique.** Une fois l'association des données réalisée, la métrique $\mathcal{F}_{IP}$ est minimisée à l'aide de l'algorithme de Levenberg-Marquardt (Levenberg (1944)).

A.3 Résultats expérimentaux

Dans cette section, nous tâcherons de présenter des résultats quantitatifs et qualitatifs sur les reconstructions obtenues grâce à la méthode présentée dans cette annexe. Notons que ces résultats pourront être comparés aux résultats obtenus avec les autres méthodes proposées (chapitre 5).

A.3.1 Protocole expérimental

Les tests réalisés ci-après consistent à optimiser la scène 3D obtenue après l'ICP non-rigide (chapitre 3) dont les résultats seront rappelés par la suite. Comme nous l'avons vu précédemment (section A.2.3.2), cette optimisation consiste alors à itérer les étapes d'association de données et de minimisation de la métrique retenue. En pratique, 10 itérations association-minimisation sont effectuées. Chacune des minimisations comprend 10 itérations du Levenberg-Marquardt.

A.3.2 Résultats obtenus

Dans cette section seront décrits les résultats obtenus sur la séquence Synthèse 1 puis sur la séquence Versailles 1.

A.3.2.1 Séquence Synthèse 1

Des premiers tests ont été réalisés sur la séquence Synthèse 1 décrite à la section 5.1. Comme précisé auparavant, l'avantage de cette séquence est qu'elle fournit une vérité terrain de la position de la caméra à chaque instant. Des mesures quantitatives sur les résultats obtenus sont donc possibles.

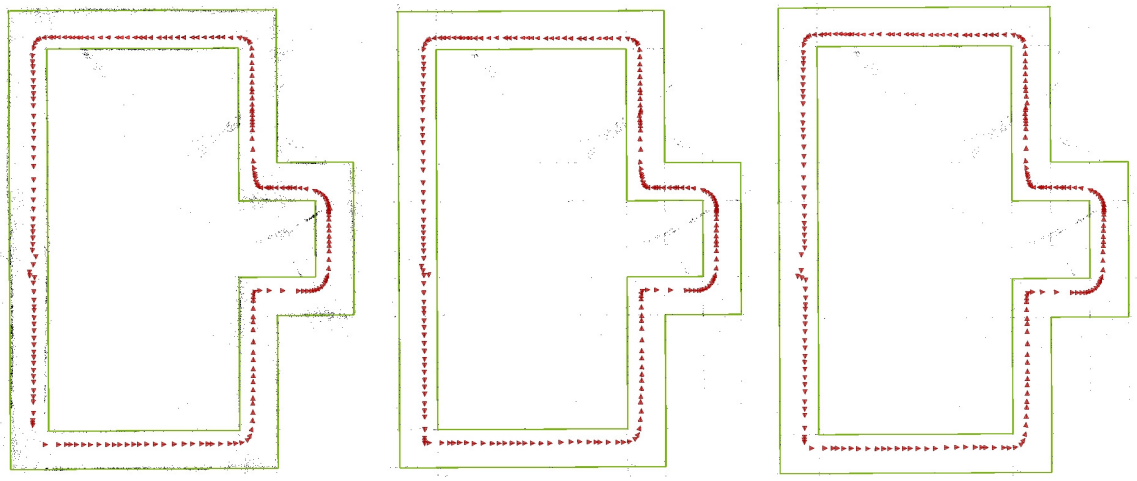
Cette vérité terrain nous permettra également de déterminer la valeur optimale du paramètre α. En pratique, 20 valeurs différentes ont été testées entre 0 et 2. Les statistiques obtenues sur les reconstructions pour les valeurs les plus significatives de α sont répertoriées dans le tableau A.1. Cette expérience souligne la sensibilité de cette méthode vis à vis du choix du paramètre α. En effet, les résultats obtenus montrent qu'une faible variation de α peut avoir un impact notable sur la reconstruction SLAM obtenue. On notera même que certaines valeurs de α peuvent entraîner une dégradation de cette reconstruction. Par exemple, pour $\alpha = 2$, la distance moyenne entre les caméras reconstruites et la vérité terrain passe de 0,51 mètres à 0,53 mètres.

	Après ICP non-rigide (initialisation)	$\alpha = 0,01$	$\alpha = 0,1$	$\alpha = 0,5$	$\alpha = 1$	$\alpha = 1,5$	$\alpha = 2$
Distance moyenne entre les caméras et la vérité terrain (m)	0,51	0,52	0,48	**0,32**	0,35	0,42	0,53
Ecart-type (m)	0,59	0,89	0,67	**0,23**	**0,23**	0,35	0,77
Distance point-plan moyenne (m)	0,11	0,13	0,06	**0,05**	**0,05**	**0,05**	**0,05**
Ecart-type (m)	0,08	**0,04**	0,06	0,07	0,07	0,07	0,07
Seuil de Tukey	0,38	×	×	×	×	×	×

TABLE A.1 – **Résultats numériques obtenus sur la séquence de synthèse.** Chaque valeur est une moyenne sur l'ensemble de la reconstruction.

Nous avons vu précédemment que le paramètre α a pour but de régler le poids des fonctions $\mathcal{F}_{IP}$ et $\mathcal{G}$ dans la minimisation. Ainsi, dans notre cas, plus la valeur de α sera élevée, plus les points auront tendance à être plaqués sur les murs. Ce phénomène est directement observable sur la figure A.2. En effet, en comparant les reconstructions obtenues avec différentes valeurs de α, on peut apercevoir que la géométrie de ces reconstructions sont très dépendantes de la valeur de α. En particulier, pour $\alpha = 2$ (figure A.2(c)), les points 3D sont presque tous forcés à être sur le modèle 3D. Ceci explique les mauvais résultats observés pour cette valeur dans le tableau A.1.

Néanmoins, nos expériences sur cette séquence de synthèse nous ont permis de déterminer que la valeur optimale de α (*i.e.* celle qui donne la reconstruction la plus proche de la vérité terrain) est $\alpha = 0.5$. Cette valeur permet de réduire notablement l'erreur de positionnement des caméras présente à la sortie de l'étape d'ICP non rigide (figure A.3). En particulier, le tableau A.1 indique que l'erreur de positionnement moyenne des caméras passe de 50 à 35 centimètres.



(a) $\alpha = 0.01$ (b) $\alpha = 0.5$ (c) $\alpha = 2$

FIGURE A.2 – **Reconstructions obtenus pour différentes valeurs de α.** Plus la valeur de α augmente, plus les points 3D ont tendance à être plaqués sur le modèle.

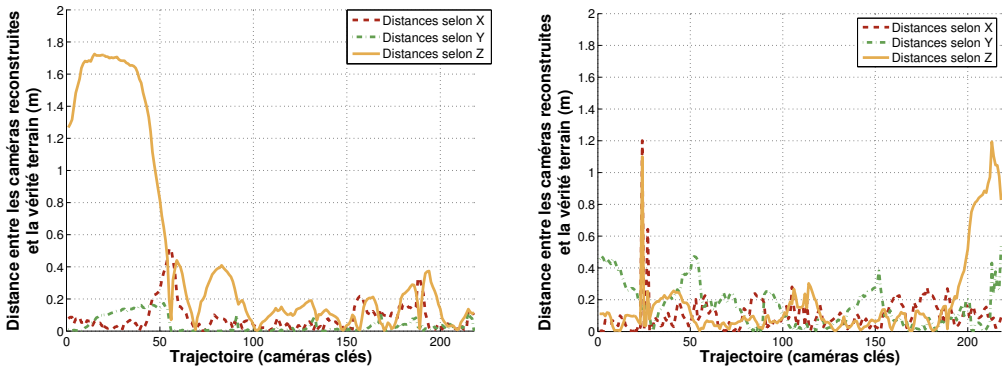
A.3.2.2 Séquence Versailles 1

Des tests ont également été réalisés sur la séquence réelle Versailles 1 afin d'étudier l'adaptabilité de la méthode à un environnement réel. Une description détaillée de cette séquence peut être trouvée à la section 5.2. Comme il a été précisé dans cette même section, aucune vérité terrain n'est disponible sur cette séquence. Les résultats obtenus seront donc uniquement analysés quantitativement. De plus, de part cette absence de vérité terrain, il est impossible de définir la valeur de α optimale. Nous utiliserons donc la valeur optimale trouvée sur la séquence de synthèse, c'est à dire $\alpha = 0,5$.

La figure A.4 regroupe les différentes étapes de la méthode complète proposée (*i.e.* ICP non-rigide suivi de l'ajustement de faisceaux proposé ici). En particulier, la figure A.4(c) met en avant la reconstruction obtenue suite à l'étape d'ICP non-rigide suivie de l'ajustement de faisceaux proposé dans cette annexe. Le premier commentaire qu'il est possible de faire est que la reconstruction finale a retrouvé sa cohérence globale vis à vis du modèle 3D de l'environnement.

Néanmoins, le nuage de points reconstruit semble être plaqué de façon importante sur le modèle 3D des bâtiments. Cela met en avant les deux problèmes majeurs de cette approche. Tout d'abord, cela montre qu'il est important de régler finement la valeur du paramètre α. En effet, il semblerait que dans cette expérience, la valeur retenue (*i.e.* $\alpha = 0,5$) soit trop élevée, ce qui implique un poids important de la contrainte aux plans dans la minimisation. Il en découle dès lors que, comme nous l'avions évoqué à la section A.1, la valeur optimale de α est dépendante de la séquence traitée. En effet, si $\alpha = 0,5$ est optimale sur la séquence Synthèse 1, il semblerait que ce réglage ne convienne pas à la séquence Versailles 1.

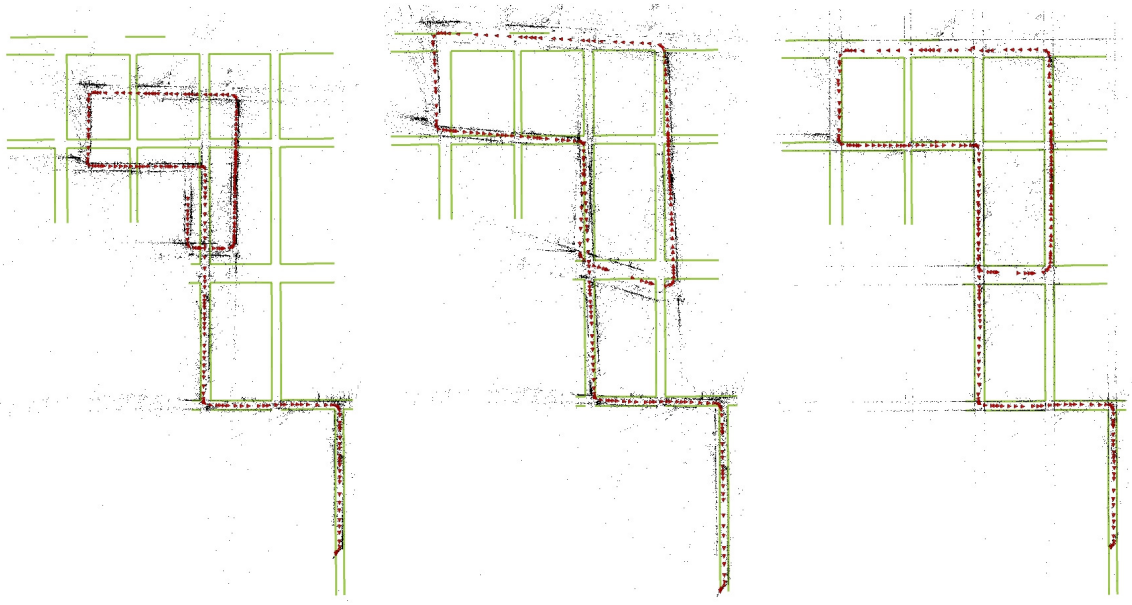
C'est pour pallier ces différents problèmes que nous avons proposé dans nos travaux une nouvelle fonction de coût qui prend en compte à la fois les contraintes aux plans et les contraintes de reprojection dans les images sans nécessiter le réglage d'un paramètre de pondération. Cette fonction de coût est présentée dans le chapitre 4.



(a) Après l'ICP non-rigide

(b) Après l'ICP non-rigide + l'ajustement de faisceaux par combinaison linéaire ($\alpha = 0,5$)

FIGURE A.3 – **Erreur de positionnement des caméras.** Le repère (X, Y, Z) est relatif à chacune des caméras : Z correspond à l'axe optique, X la direction latérale et Y l'altitude.



(a) (b) (c)

FIGURE A.4 – **Résultats sur la séquence Versailles 1.** (a) est la reconstruction initiale obtenue avec la méthode de Mouragnon et al. (2006), (b) est l'initialisation avant l'ICP non-rigide et (c) est la reconstruction obtenue après l'ICP non-rigide puis l'ajustement de faisceaux présenté dans cet annexe.

ANNEXE B

Séquences vidéos utilisées

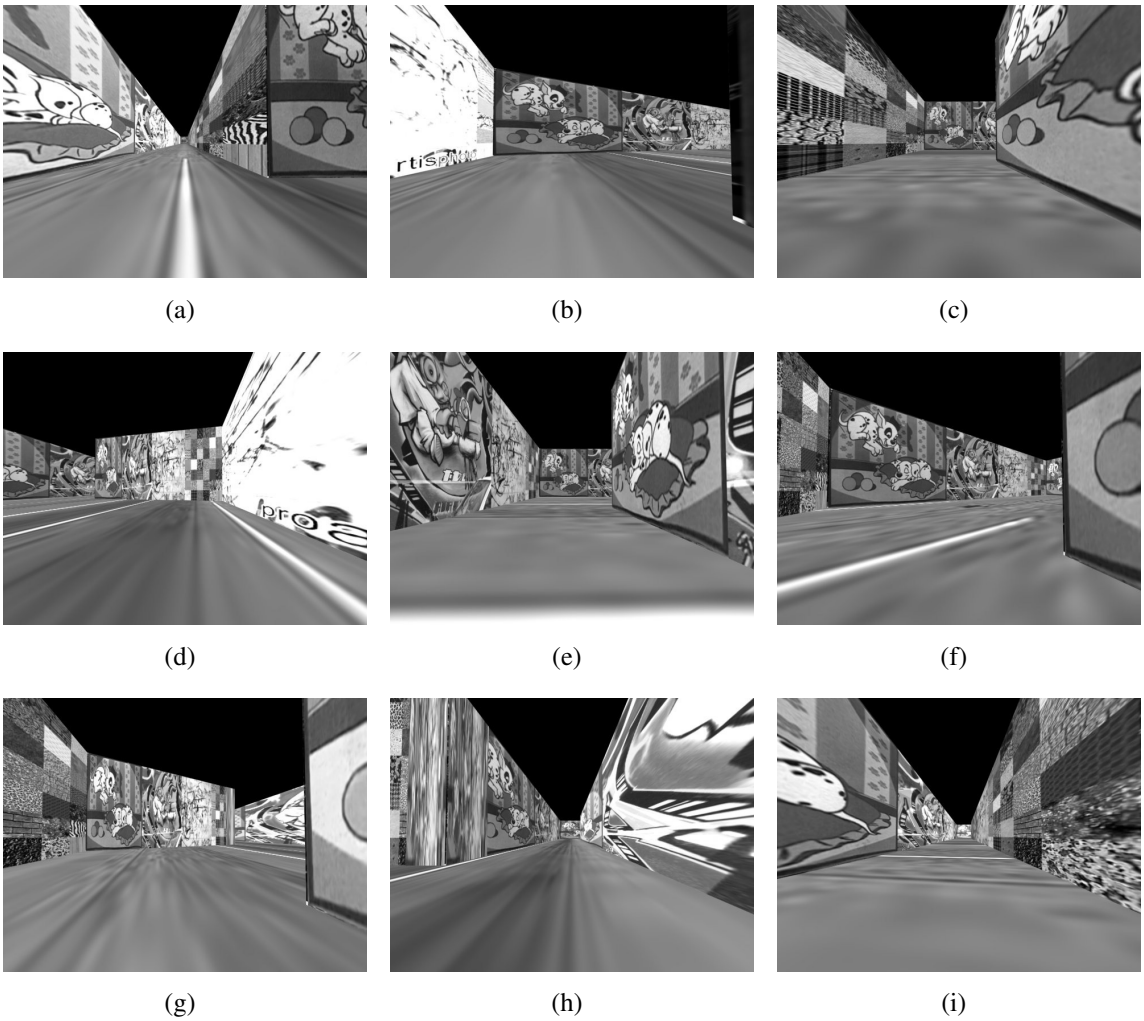
B.1 Séquence Synthèse 1

B.1.1 Statistiques

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
Synthèse 1	420	218	6848

TABLE B.1 – **Statistiques sur la séquence Synthèse 1.**

B.1.2 Extraits de la vidéo



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE B.1 – **Captures de la séquence Synthèse 1.**

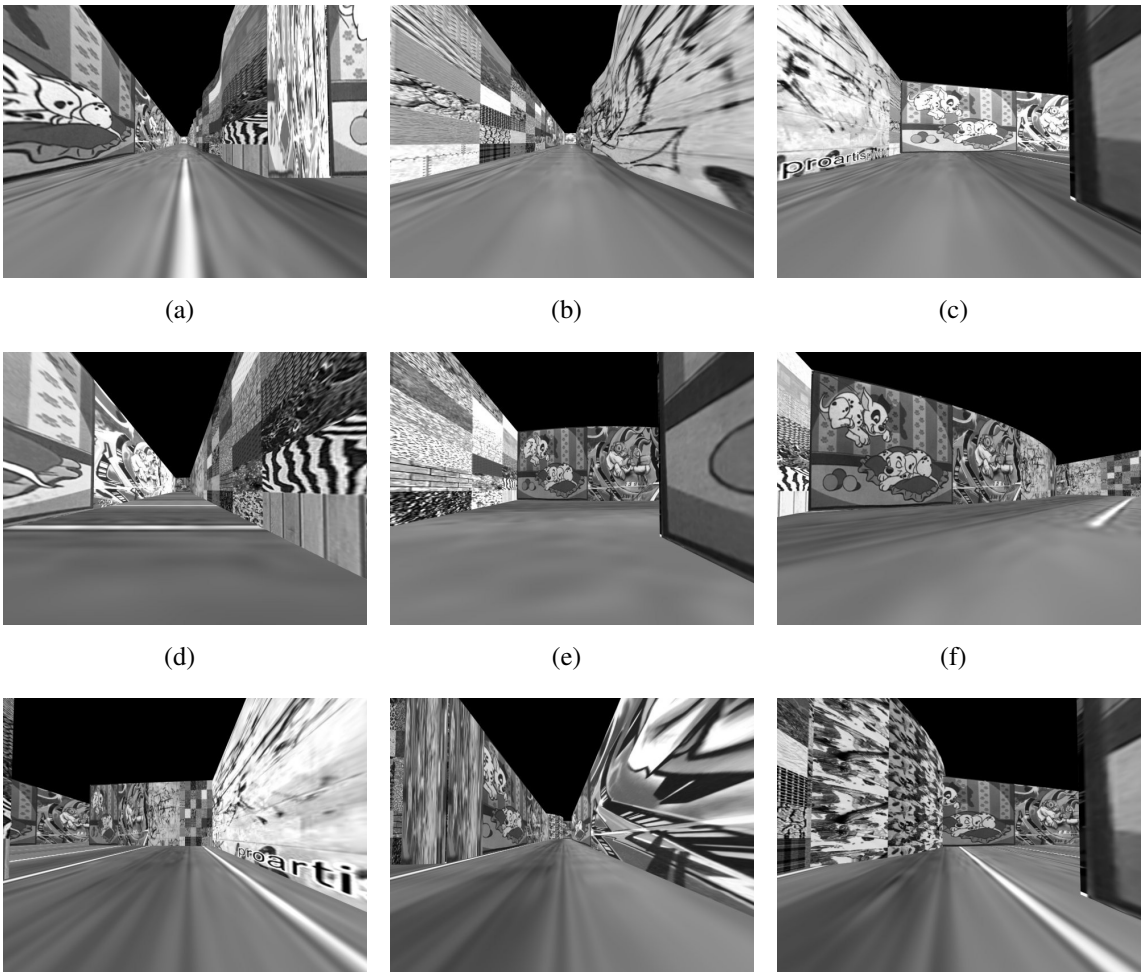
B.2 Séquence Synthèse 2

B.2.1 Statistiques

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
Synthèse 2	420	184	6258

TABLE B.2 – **Statistiques sur la séquence Synthèse 2.**

B.2.2 Extraits de la vidéo



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE B.2 – **Captures de la séquence Synthèse 2.**

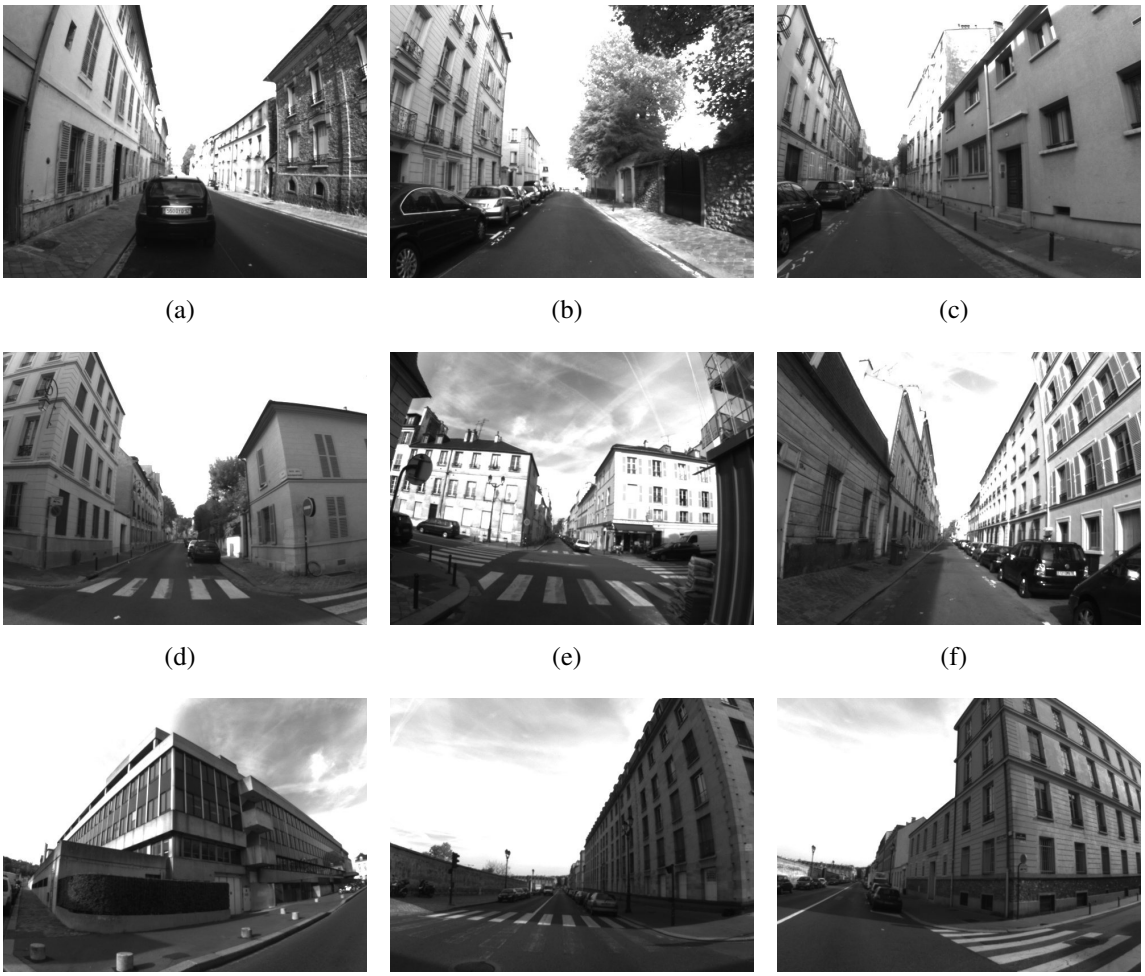
B.3 Séquence Versailles 1

B.3.1 Statistiques

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
Versailles 1	1500	240	16761

TABLE B.3 – **Statistiques sur la séquence Versailles 1.**

B.3.2 Extraits de la vidéo



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE B.3 – **Captures de la séquence Versailles 1.**

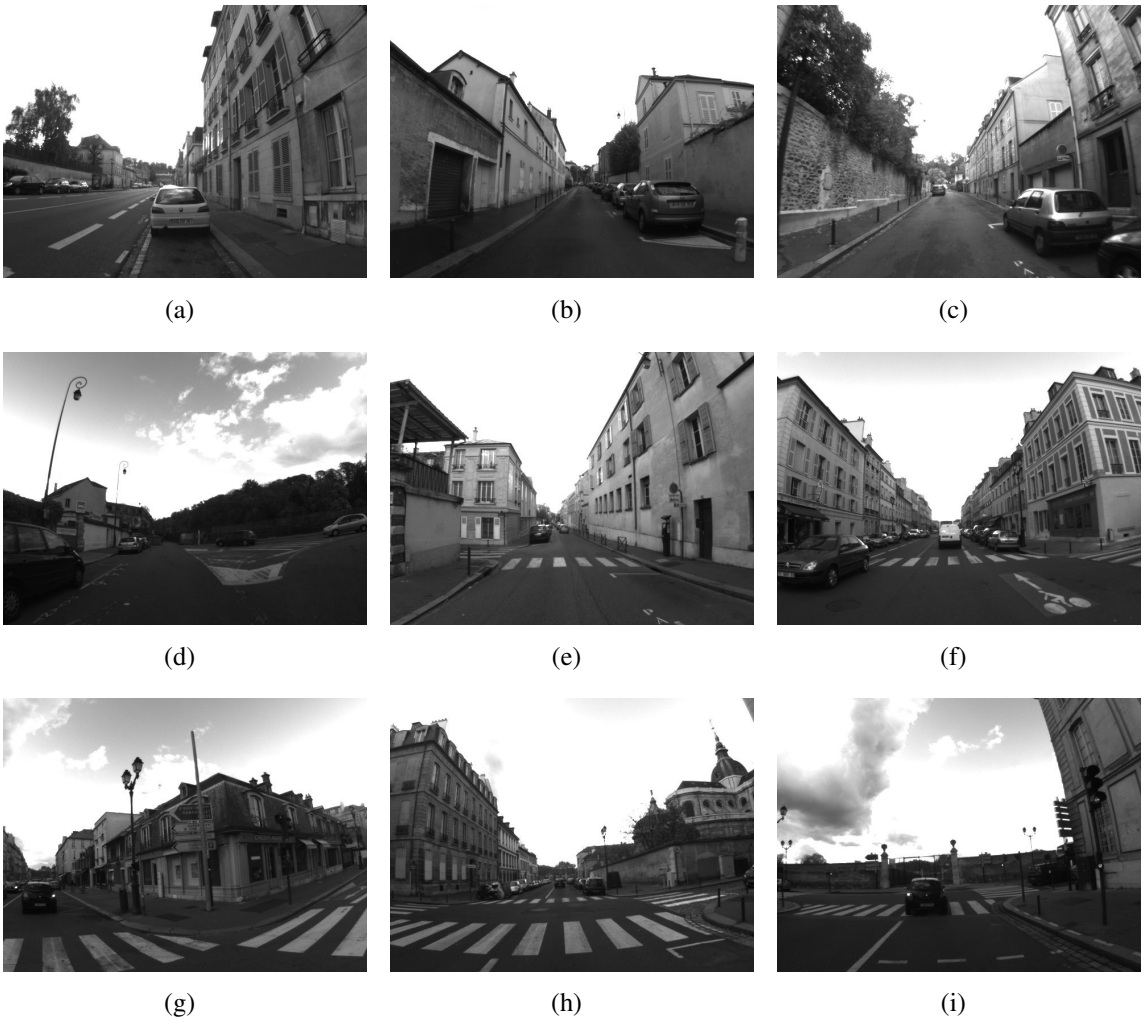
B.4 Séquence Versailles 2

B.4.1 Statistiques

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
Versailles 2	2000	313	14780

TABLE B.4 – **Statistiques sur la séquence Versailles 2.**

B.4.2 Extraits de la vidéo



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE B.4 – **Captures de la séquence Versailles 2.**

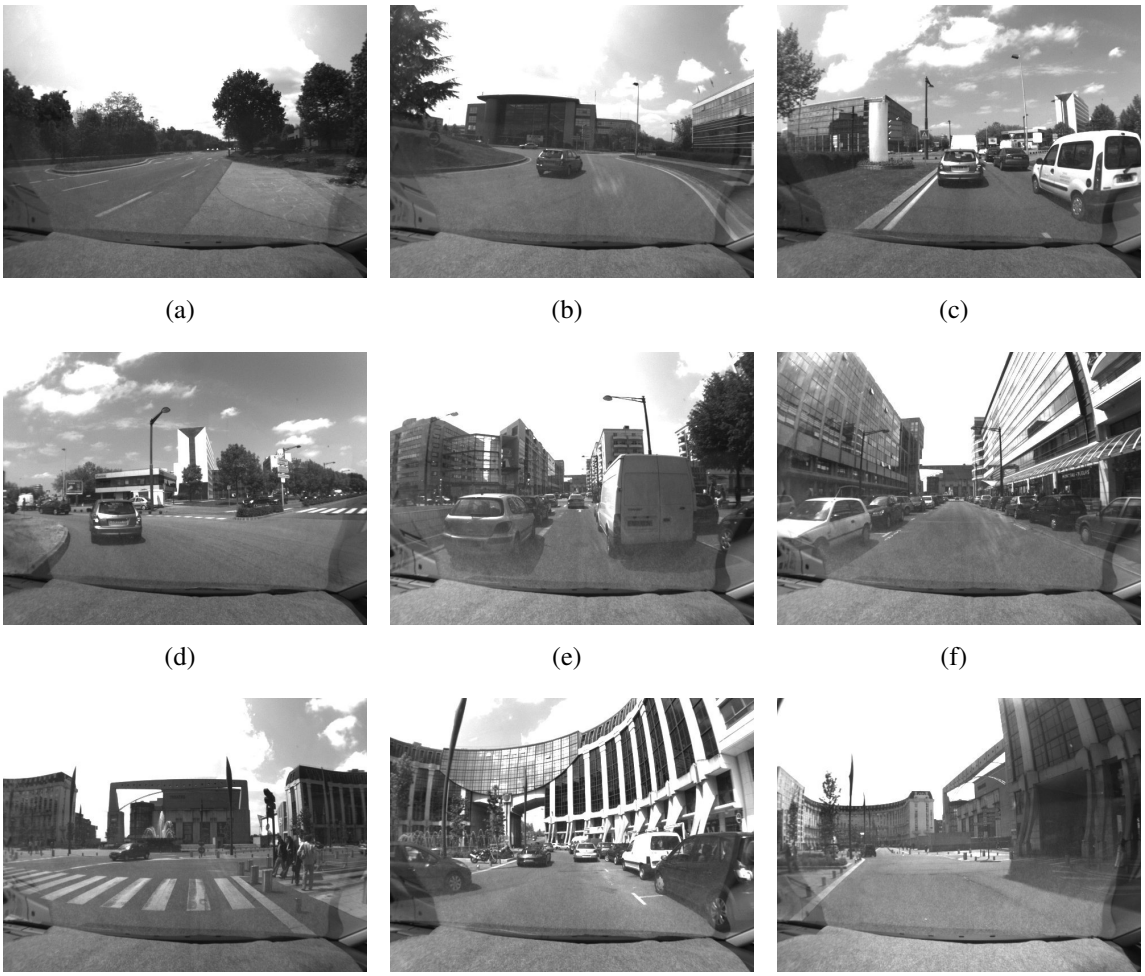
B.5 Séquence ODIAAC

B.5.1 Statistiques

	Longueur (m)	Nombre de caméras clés	Nombre de points 3D reconstruits
ODIAAC	4500	1296	39304

TABLE B.5 – **Statistiques sur la séquence ODIAAC.**

B.5.2 Extraits de la vidéo



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE B.5 – **Captures de la séquence ODIAAC.**

Bibliographie

H. Adams, S. Singh, and D. Strelow. An empirical comparison of methods for image-based motion estimation. In *International Conference on Intelligent Robots and Systems*, 2002.

S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, R. Szeliski, and R. Szeliski. Building Rome in a day. In *International Conference on Computer Vision*, 2009.

M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *International Conference on Pattern Recognition*, pages 1063–1068, 2006.

S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

A. Bartoli and P. Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52:45–64, 2003.

A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 346–359, 2006.

C. Berger and S. Lacroix. Using planar facets for stereovision slam. In *International Conference on Intelligent Robots and Systems*, pages 1606–1611, 2008.

N. Bioret, G. Moreau, and M. Servières. Géolocalisation en milieu urbain par appariement entre une collection d'images et un SIG 2D= Outdoor localization in urban environment using matching between an image collection and a 2D GIS. *Ingénierie des systèmes 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*, pages 44–57, 2008.

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*, 14:3–12, 2010.

U. Castellani, V. Gay-Bellile, and A. Bartoli. Joint reconstruction and registration of a deformable planar surface observed by a 3d sensor. In *3-D Digital Imaging and Modeling*, pages 201–208, 2007.

R. O. Castle and D. W. Murray. Object recognition and localization while tracking and mapping. In *International Symposium on Mixed and Augmented Reality*, 2009.

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.

B. Charmette, E. Royer, and F. Chausse. Matching planar features for robot localization. In *International Symposium on Advances in Visual Computing*, pages 201–210, 2009. ISBN 978-3-642-10330-8.

H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

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*, pages 3498–3504, 2009.

L. Clemente, Andrew Davison, Ian Reid, J. Neira, and J. Tardos. Mapping Large Loops with a Single Hand-Held Camera. In *Robotics: Science and Systems*, 2007.

A.I. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *International Conference on Robotics and Automation*, pages 40 –45, 2007.

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.

B. N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, (6):793–800, 1934.

Y. Dumortier, M. Kais, and R. Benenson. Real-time vehicle motion estimation using texture learning and monocular vision. In *International Conference on Computer Vision and Graphics*, 2006.

E. Eade and T. Drummond. Scalable monocular slam. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–476, 2006.

A.F. Elaksher, J.S. Bethel, and E.M. Mikhail. Reconstructing 3d building wireframes from multiple images. In *Proceedings of the ISPRS Commission III Symposium on Photogrammetric Computer Vision*, page A: 91, 2002.

C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In *In Photogrammetric Computer Vision*, 2006.

A. Eudes and M. Lhuillier. Error propagations for local bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418, 2009.

M. Farenzena, A. Bartoli, and Y. Mezouar. Efficient camera smoothing in sequential structure-from-motion using approximate cross-validation. In *European Conference on Computer Vision*, pages 196–209, 2008.

O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *British Machine Vision Conference*, pages 411–420, 2001.

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. Grzeszczuk, J. Košecka, R. Vedantham, and H. Hile. Creating Compact Architectural Models by Geo-registering Image Collections. In *3-D Digital Imaging and Modeling*, 2009.

R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. 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 *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

R. I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence*, 19:580–593, 1997. ISSN 0162-8828.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

Jeroen D. Hol, Thomas B. Schon, Henk Luinge, Per J. Slycke, and Fredrik Gustafsson. Robust real-time tracking by fusing measurements from inertial and vision sensors. *Journal of Real-Time Image Processing*, 2(2):149–160, 2007. ISSN 1861-8200.

B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3): 185–203, 1981.

P. Huber. *Robust Statistics*. Wiley, New-York, 1981.

Sei Ikeda, Tomokazu Sato, Koichiro Yamaguchi, and Naokazu Yokoya. Construction of feature landmark database using omnidirectional videos and gps positions. In *3-D Digital Imaging and Modeling*, pages 249–256, 2007.

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*, pages 2599–2606, 2009.

M. Kaess and F. Dellaert. Probabilistic structure matching for visual slam with a multi-camera rig. *Computer Vision and Image Understanding*, 114(2):286–296, 2010.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–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 *Second 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*, volume 1, 2003.

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 *In Proc. International Symposium on Robotics Research*, 2007.

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. ISSN 0920-5691.

J. Leonard and P. Newman. Consistent, convergent, and constant-time slam. In *International Joint Conference on Artificial intelligence*, 2003.

V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

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. *Quart. Appl. Math.*, 2:164–168, 1944.

Anat Levin and Richard Szeliski. Visual odometry and map correlation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 611–618, 2004.

B. Liang and N. Pears. Visual navigation using planar homographies. In *International Conference on Robotics and Automation*, 2002.

P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Toward large scale model construction for vision-based global localisation. In *International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, February 2009a.

P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Slam monoculaire et cartographie : vers une mise en cohérence fine. In *Actes de la conférence Compression et Représentation des Signaux Audiovisuels (CORESA)*, Toulouse, France, March 2009b.

P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Vers la géolocalisation par vision d'une caméra mobile : exploitation d'un modèle 3d de ville et application au recalage visuel temps réel. In *Actes du Congrès des jeunes chercheurs en vision par ordinateur ORASIS*, Tregastel, France, June 2009c.

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*, Miami, Florida, June 2009d.

P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Bases d'amers visuels à grande échelle : correction et géolocalisation de reconstructions slam monoculaires à l'aide de modèles 3d grossiers de villes. In *Actes de la conférence Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, Caen, France, January 2010a.

P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Real-time vehicle global localisation with a single camera in dense urban areas: Exploitation of coarse 3d city models. *Revised Selected Papers Series: Communications in Computer and Information Science*, July 2010b.

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*, San Francisco, California, June 2010c.

D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

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*, 22:610–622, 2000.

E. Malis and E. Marchand. Experiments with robust estimation techniques in real-time robot vision. In *International Conference on Intelligent Robots and Systems*, pages 223–228, 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 *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, pages 128–142, 2002.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

R. Mittu and F. Segaria. Common operational picture and common tactical picture management via a consistent networked information stream. In *Proceedings of the Command and Control Research and Technology Symposium*, 2000.

J. Modersitzki. *Numerical methods for image registration*. Oxford University Press, USA, 2004.

N. Molton, A. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *British Machine Vision Conference*, 2004.

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*, pages 363–370, 2006.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. *International Symposium on Robotics Research*, 1989.

D. Nistér. 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*, pages 652–659, 2004.

D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23:2006, 2006.

David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2005.

O. Pink. Visual map matching and localization using a global feature map. In *Computer Vision and Pattern Recognition Workshops*, 2008.

J. Platonov, H. Heibel, P. Meier, B. Grollmann, and B. Metaio. A mobile markerless AR system for maintenance and repair. In *International Symposium on Mixed and Augmented Reality*, pages 105–108, 2006.

T. Pylvänäinen, K. Roimela, R. Vedantham, J. Itäranta, R. Wang, and R. Grzeszczuk. Automatic Alignment and Multi-View Segmentation of Street View Data using 3D Shape Priors. In *3D Data Processing, Visualization and Transmission*, 2010.

Srikumar Ramalingam, Suresh Lodha, and Peter Sturm. A generic structure-from-motion framework. *Computer Vision and Image Understanding*, 103(3):218–228, 2006.

Gerhard Reitmayr and Tom Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *International Symposium on Mixed and Augmented Reality*, pages 109–118, 2006.

P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

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*, pages 114–121, 2005.

E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3): 237–260, 2007.

S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling*, pages 145–152, 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, Special Issue on Visual SLAM. In press.*, 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.

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. *Robotics, IEEE Transactions on*, 24(5):969 –979, october 2008.

N. Simond and P. Rives. Trajectography of an uncalibrated stereo rig in urban environments. In *International Conference on Intelligent Robots and Systems*, pages 3381–3386, 2004.

R. Smith and P. Cheesman. On the representation of spatial uncertainty. *International Journal of Robotics Research*, 1987.

G. Sourimant, L. Morin, and K. Bouatouch. Gps, gis and video fusion for urban modeling. In *Computer Graphics International*, may 2007.

H. Strasdat, J. M. M. Montiel, and A. J. Davison. Scale-drift aware large sclale monocular SLAM. In *Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

C. Strecha, T. Pylvänäinen, and P. Fua. Dynamic and scalable large scale image reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

D. W. Strelow and S. Singh. Motion estimation from image and inertial measurements. *The International Journal of Robotique Research*, 2004.

J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *International Conference on Computer Vision*, 2009.

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*, pages 2531–2538, 2008.

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*, 24:271–300, 1997.

B. Triggs. Autocalibration from planar scenes. In *European Conference on Computer Vision*, 1998.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *International Conference on Computer Vision*, pages 298–372, 2000.

L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004.

H. Wang, K. Yuan, W. Zou, and Q. Zhou. Visual odometry based on locally planar ground assumption. In *Information Acquisition, 2005 IEEE International Conference on*, 27 2005.

Table des figures

1 Exemples de systèmes d'aide à la navigation exploitant la vision 2

1.1 Utilisation de modèles 3D texturés 12
1.2 Recalage d'une reconstruction SLAM sur un SIG 3D 13
1.3 Exploitation de la géométrie du modèle 3D 15

2.1 Projection perspective 19
2.2 Géométrie épipolaire 23
2.3 Triangulation de points 3D 26
2.4 Erreur de reprojection 27
2.5 Homographies 2D 29
2.6 Exemples de M-estimateurs 35
2.7 Schéma du fonctionnement du SLAM de Mouragnon et al. (2006) 36
2.8 Les étapes de la méthodes de Mouragnon et al. (2006) 38
2.9 Exemples de reconstruction obtenues avec la méthode de Mouragnon et al. (2006) 39
2.10 Exemples de SIG 40
2.11 Modèle 3D texturé 41
2.12 Modèle 3D utilisé dans nos travaux 42
2.13 Résumé de la méthode de relocalisation d'une caméra 49

3.1 Approximation polygonale de la trajectoire du véhicule 54
3.2 Transformation par fragments d'une reconstruction SLAM 55
3.3 Histogramme des distances pour l'ICP non-rigide 58
3.4 Résultat de l'alignement par ICP non-rigide 59

4.1 Comportement de l'ajustement de faisceaux classique 62
4.2 Fonction de coût proposée 65
4.3 Robustesse de l'ajustement de faisceaux 66
4.4 Résultat de la reconstruction après l'ajustement de faisceaux proposé 67

5.1 Extraits d'une vidéo de synthèse 69
5.2 Déroulement de la méthode sur la séquence Synthèse 1 71
5.3 Evolution du facteur d'échelle sur la séquence Synthèse 1 72
5.4 Erreur de positionnement des caméras sur la séquence Synthèse 1 73

5.5 Robustesse à un modèle 3D erroné 75
5.6 Précision obtenue avec un modèle 3D erroné 77
5.7 Extraits des séquences vidéos à Versailles 78
5.8 Séquences de Versailles 79
5.9 Recalage des reconstructions obtenues sur les images satellites 80
5.10 Exemple d'amélioration de la géométrie locale 81
5.11 Projection du modèle 3D de ville dans les caméras clés. 81
5.12 Fusion de plusieurs reconstructions 82
5.13 Comparaison des matrices jacobiennes et hessiennes 84
5.14 Exemple de manque de contrainte dans l'ajustement de faisceaux 85

6.1 Données utilisées pour la relocalisation 88
6.2 Relocalisation d'une caméra dans un environnement connu 89
6.3 Couplage du SLAM avec des données géoréférencées 91
6.4 Relocalisation pour l'aide à la navigation 92
6.5 Résultats complémentaires en réalité augmentée 93
6.6 Données du Système d'Information Géographique 3D 101

7.1 Modules nécessaires à l'estimation du facteur d'échelle 107
7.2 Contexte de la recherche du facteur d'échelle 108
7.3 Recherche de l'homographie principale d'un couple d'images 110
7.4 Données utilisées pour la sélection globale des points du sol 112
7.5 Résultats obtenus avec la sélection globale 113
7.6 Méthode de sélection locale des points du sol 113
7.7 Résultats obtenus avec la sélection locale 114
7.8 Aperçu de la validation stricte 116
7.9 Description de la méthode globale 117
7.10 Description de la méthode locale 117
7.11 Résumé de la méthode d'estimation du facteur d'échelle hybride 118
7.12 Trajectoire de la séquence ODIAAC 119
7.13 Extraits de la séquence ODIAAC 119
7.14 Vérité terrain de la séquence ODIAAC 120
7.15 Reconstruction SLAM originale obtenue 121
7.16 Scénario d'un trafic dense 122
7.17 Illustration de la mauvaise estimation de la normale du sol 124
7.18 Résultat obtenu avec une meilleure estimation de la normale 125
7.19 Contexte de l'intégration du facteur d'échelle 125
7.20 Observations de la position de la caméra clé courante 126
7.21 Approche proposée pour l'intégration du facteur d'échelle 128
7.22 Résumé de la séquence ODIAAC 130
7.23 Reconstructions obtenues après l'intégration du facteur d'échelle sur la séquence ODIAAC 131
7.24 Résultats numériques sur l'intégration du facteur d'échelle sur la séquence ODIAAC 132
7.25 Résultats sur la séquence de Versailles 133

8.1 Illustration du problème d'aperture dans les lignes droites 136
8.2 Estimation possible de la rotation 137

8.3 Contraintes géométriques dans les virages 137
8.4 Amélioration des associations point-plan 138
8.5 Contexte de l'estimation des normales 139
8.6 Ambiguïté sur le sens de la normale 141
8.7 Exemple de normales reconstruites 142
8.8 Distance utilisée entre la reconstruction SLAM et la route 146
8.9 Les étapes de la correction du cap de la reconstruction 147
8.10 Exemple de recalage dans les virages 148
8.11 Les différentes étapes de la méthode proposée sur la séquence Versailles 1 . . . 151
8.12 Résultat de la localisation absolue obtenue sur la séquence Versailles 1 152
8.13 Les différentes étapes de la méthode proposée sur la séquence ODIAAC 153
8.14 Evolution de l'erreur de positionnement absolue 154
8.15 Résultat de la localisation absolue obtenue sur la séquence ODIAAC 155

A.1 Plans d'interprétation et résidus 3D 163
A.2 Reconstructions obtenus pour différentes valeurs de α 165
A.3 Erreur de positionnement des caméras 166
A.4 Résultats sur la séquence Versailles 1 167

B.1 Captures de la séquence Synthèse 1 170
B.2 Captures de la séquence Synthèse 2 171
B.3 Captures de la séquence Versailles 1 172
B.4 Captures de la séquence Versailles 2 173
B.5 Captures de la séquence ODIAAC 174

Liste des tableaux

2.1 Description simplifiée de l'algorithme à la base de RANSAC et LMedS 33

4.1 Résultats numériques obtenus avec l'ajustement de faisceaux classique 63
4.2 Qualité de la reconstruction en fonction du facteur α 64

5.1 Statistiques sur les reconstructions des séquences de synthèse 70
5.2 Résultats numériques obtenus sur la séquence Synthèse 1 72
5.3 Statistiques sur la précision obtenue avec un modèle erroné 77
5.4 Statistiques sur les séquences réelles 78

7.1 Statistiques sur la reconstruction de la séquence ODIAAC 120
7.2 Statistiques sur l'estimation du facteur d'échelle 121
7.3 Statistiques sur l'estimation du facteur d'échelle pour le scénario proposé . . . 123
7.4 Statistiques sur l'intégration du facteur d'échelle sur la séquence ODIAAC . . . 131

8.1 Statistiques sur la localisation absolue 154

A.1 Résultats numériques obtenus sur la séquence de synthèse 165

B.1 Statistiques sur la séquence Synthèse 1 170
B.2 Statistiques sur la séquence Synthèse 2 171
B.3 Statistiques sur la séquence Versailles 1 172
B.4 Statistiques sur la séquence Versailles 2 173
B.5 Statistiques sur la séquence ODIAAC 174

Table des matières

Introduction **1**

1 Etat de l'art **7**
1.1 Localisation monoculaire en environnement inconnu 7
1.1.1 Idée générale 7
1.1.2 Familles de méthodes existantes 8
1.1.2.1 Odométrie visuelle 8
1.1.2.2 Localisation et cartographie simultanées par vision 8
1.1.2.3 Structure from Motion 9
1.1.3 Limites des méthodes existantes 10
1.1.3.1 Localisation non géoréférencée 10
1.1.3.2 Dérives sur les longues trajectoires 10
1.2 Environnements partiellement connus : exploitation des Systèmes d'Information Géographique 11
1.2.1 Exploitation de l'information photo-géométrique 11
1.2.2 Exploitation exclusive de l'information géométrique 12
1.2.2.1 Localisation hors ligne 13
1.2.2.2 Localisation en ligne 14

2 Notions de base et données utilisées **17**
2.1 Géométrie projective 17
2.1.1 Le plan projectif 18
2.1.2 L'espace projectif 3D 18
2.2 Caméras perspectives et géométrie associée 18
2.2.1 Projection perspective 18
2.2.1.1 Paramètres extrinsèques 20
2.2.1.2 Projection centrale 20
2.2.1.3 Paramètres intrinsèques et distorsion 21
2.2.2 Notion de rétroprojection 21
2.3 Géométrie multi-vue 22
2.3.1 Géométrie épipolaire 22

2.3.1.1 Matrice fondamentale . . . 22
2.3.1.2 Matrice essentielle . . . 23
2.3.1.3 Relation entre matrice essentielle et déplacement relatif . . . 23
2.3.2 Calcul de la géométrie de l'environnement . . . 24
2.3.2.1 Poses de caméras et déplacement relatif . . . 24
2.3.2.2 Calcul du déplacement relatif par associations 2D/2D . . . 25
2.3.2.3 Calcul de la structure de l'environnement . . . 25
2.3.2.4 Calcul de pose par associations 2D/3D . . . 26
2.3.2.5 Erreur de reprojection et ajustement de faisceaux . . . 26
2.4 Cas d'une scène plane . . . 28
2.4.1 Homographies 2D . . . 28
2.4.2 Relation entre homographie 2D, équation de plan et déplacement relatif 28
2.5 Optimisation numérique . . . 30
2.5.1 Moindres carrés . . . 30
2.5.2 Méthodes de résolution linéaires . . . 31
2.5.3 Méthodes de résolution non-linéaires . . . 31
2.5.4 Optimisation robuste . . . 32
2.5.4.1 RANSAC et LMedS . . . 32
2.5.4.2 M-estimateurs . . . 34
2.6 Algorithmes et données utilisés . . . 35
2.6.1 Algorithme de localisation et cartographie simultanées . . . 35
2.6.1.1 Traitements 2D . . . 36
2.6.1.2 Traitements 3D . . . 37
2.6.2 Les modèles 3D urbains . . . 39
2.6.2.1 Système d'Information Géographique . . . 40
2.6.2.2 Les modèles 3D à grande échelle. . . . 40
2.6.2.3 Caractéristiques des modèles 3D utilisés . . . 42

I Création d'une base d'amers visuels et relocalisation d'une caméra mobile 43

Présentation de la méthode 47

3 ICP non-rigide 51
3.1 Méthodes d'alignement 3D . . . 51
3.2 Espace de transformations utilisé . . . 52
3.2.1 Modélisation de la dérive du SLAM . . . 52
3.2.2 Fragmentation de la reconstruction . . . 53
3.2.2.1 Approximation polygonale de la trajectoire . . . 53
3.2.2.2 Formation des fragments de la reconstruction . . . 54
3.2.3 Paramétrisation des transformations retenues . . . 55
3.3 Recherche de l'alignement optimal . . . 56
3.3.1 Métrique utilisée et association des données . . . 56
3.3.2 Minimisation robuste de la métrique . . . 57
3.3.3 Initialisation de l'algorithme . . . 58
3.4 Discussion . . . 59

4 Ajustements de faisceaux contraints par un SIG 61
4.1 Méthodes classiques et limites . . . 61
4.1.1 Ajustement de faisceaux classique . . . 61
4.1.2 Combinaison linéaire de fonctions de coût . . . 62
4.2 Approche proposée : ST-CBA . . . 63
4.2.1 Fonction de coût proposée . . . 64
4.2.2 Optimisation robuste . . . 66
4.2.2.1 Robustesse aux points aberrants . . . 66
4.2.2.2 Remise en cause des associations point-plan . . . 66

5 Résultats expérimentaux 69
5.1 Evaluation quantitative sur des données de synthèse . . . 69
5.1.1 Evaluation de la précision . . . 70
5.1.1.1 Reconstruction SLAM originale . . . 70
5.1.1.2 Résultats de l'ICP non-rigide . . . 70
5.1.1.3 Apport de l'ajustement de faisceaux ST-CBA . . . 73
5.1.2 Evaluation de la robustesse . . . 74
5.1.2.1 Robustesse de la reconstruction à un modèle 3D erroné . . . 74
5.1.2.2 Précision obtenue . . . 76
5.2 Evaluation qualitative sur des données réelles . . . 77
5.2.1 Données utilisées . . . 77
5.2.2 Résultats obtenus . . . 78
5.2.3 Evaluation qualitative de la précision obtenue . . . 78
5.2.3.1 Apport de l'ajustement de faisceaux ST-CBA . . . 80
5.2.3.2 Précision obtenue . . . 81
5.3 Discussion . . . 82
5.3.1 Performance de l'approche proposée . . . 82
5.3.1.1 Qualité des résultats obtenus . . . 82
5.3.1.2 Temps de traitement nécessaires . . . 83
5.3.2 Limites de la méthode actuelle . . . 84

6 Relocalisation et réalité augmentée 87
6.1 Présentation de l'application visée . . . 87
6.2 Localisation absolue dans la base d'amers . . . 87
6.3 Vers une application d'aide à la navigation . . . 89
6.3.1 Limites de l'existant . . . 89
6.3.2 Approche proposée . . . 90
6.3.3 Résultats . . . 91
6.4 Discussion . . . 92

II Vers la correction en ligne d'une reconstruction SLAM 95

Présentation de la méthode 99

7 Méthode de correction du facteur d'échelle 103
7.1 Etat de l'art . . . 103
7.1.1 Utilisation d'un capteur tiers . . . 103
7.1.2 Utilisation d'une information tierce . . . 104
7.1.2.1 Points 3D géoréférencés . . . 104
7.1.2.2 Stéréoscopie . . . 104
7.1.2.3 Homographie du plan de la route . . . 104
7.1.2.4 Mouvement non holonome . . . 105
7.2 Contraintes disponibles et positionnement de nos travaux . . . 105
7.2.1 Données exploitées . . . 105
7.2.2 Approche proposée . . . 105
7.3 Outils nécessaires à l'estimation du facteur d'échelle . . . 106
7.3.1 Aperçu de l'algorithme développé pour l'estimation du facteur d'échelle 106
7.3.2 Module d'estimation du facteur d'échelle . . . 107
7.3.2.1 Le facteur d'échelle comme seul paramètre de l'homographie du sol . . . 107
7.3.2.2 Résolution numérique du problème . . . 109
7.3.3 Modules de sélection des points d'intérêt situés sur le sol . . . 110
7.3.3.1 Sélection globale . . . 111
7.3.3.2 Sélection locale . . . 113
7.3.4 Modules de validation du facteur estimé . . . 114
7.3.4.1 Critère souple . . . 114
7.3.4.2 Critère strict . . . 115
7.4 Méthodes d'estimation du facteur d'échelle proposées . . . 115
7.4.1 Méthode d'estimation globale . . . 116
7.4.2 Méthode d'estimation locale . . . 116
7.4.3 Méthode retenue : une approche hybride . . . 117
7.5 Validation expérimentale de l'estimation du facteur d'échelle . . . 118
7.5.1 Protocole expérimental . . . 118
7.5.1.1 Données utilisées . . . 118
7.5.1.2 Reconstruction SLAM originale . . . 120
7.5.2 Résultats obtenus . . . 120
7.5.2.1 Calcul du facteur d'échelle . . . 120
7.5.2.2 Scénario de trafic dense . . . 122
7.5.2.3 Temps de traitement . . . 123
7.5.3 Discussion . . . 123
7.6 Intégration du facteur d'échelle dans la méthode SLAM . . . 124
7.6.1 Contexte du problème étudié . . . 125
7.6.2 Limites des méthodes existantes . . . 126
7.6.2.1 Filtre de Kalman . . . 126
7.6.2.2 Contrainte de l'ajustement de faisceaux . . . 127
7.6.3 Approche retenue . . . 127
7.7 Résultats expérimentaux . . . 129
7.7.1 Protocole expérimental . . . 129
7.7.2 Résultats obtenus . . . 129
7.7.2.1 Séquence ODIAAC . . . 130
7.7.2.2 Séquence Versailles 1 . . . 133

7.8 Discussion . . . 133

8 Méthode de correction de l'accumulation d'erreur 135
8.1 Objectif de l'étude . . . 135
8.2 Alignement et contraintes géométriques exploitables . . . 136
8.2.1 Paramètres contraints dans les lignes droites . . . 136
8.2.2 Paramètres contraints dans les virages . . . 137
8.3 Estimation de la dérive à l'aide d'un modèle 3D de ville . . . 138
8.3.1 Aperçu de la méthode . . . 138
8.3.2 Calcul de patchs orientés . . . 139
8.3.2.1 Formalisation du problème . . . 139
8.3.2.2 Estimation initiale . . . 140
8.3.2.3 Optimisation de la normale . . . 140
8.3.2.4 Ambiguïté sur le sens de la normale . . . 140
8.3.2.5 Reconstruction obtenue . . . 141
8.3.3 Recalage par ICP . . . 141
8.3.3.1 Association des données . . . 143
8.3.3.2 Minimisation de l'erreur associée . . . 143
8.3.3.3 Itération de l'ICP . . . 144
8.4 Estimation de la dérive à l'aide d'une carte de la route . . . 144
8.4.1 Aperçu de la méthode . . . 144
8.4.2 Distance utilisée . . . 145
8.4.3 Estimation de la dérive en orientation dans les lignes droites . . . 146
8.4.4 Estimation de la dérive dans les virages . . . 147
8.5 Intégration de la nouvelle information . . . 149
8.6 Résultats expérimentaux . . . 149
8.6.1 Protocole expérimental . . . 150
8.6.2 Résultats sur l'exploitation des bâtiments . . . 150
8.6.3 Résultats sur l'exploitation d'une carte de la route . . . 152
8.6.4 Temps de traitement . . . 152
8.7 Discussion . . . 156

Conclusion 157

Annexes 159

A Ajustement de faisceaux par combinaison linéaire des fonctions de coût 161
A.1 Approches existantes . . . 161
A.2 Utilisation dans notre cadre d'étude . . . 162
A.2.1 Contrainte d'appartenance au modèle 3D . . . 162
A.2.2 Critère de cohérence dans les images . . . 162
A.2.3 Métrique utilisée et optimisation . . . 163
A.2.3.1 Fonction de coût retenue . . . 163
A.2.3.2 Robustesse de l'optimisation . . . 163
A.3 Résultats expérimentaux . . . 164
A.3.1 Protocole expérimental . . . 164
A.3.2 Résultats obtenus . . . 164
A.3.2.1 Séquence Synthèse 1 . . . 164

A.3.2.2 Séquence Versailles 1 166

B Séquences vidéos utilisées **169**
B.1 Séquence Synthèse 1 170
B.1.1 Statistiques 170
B.1.2 Extraits de la vidéo 170
B.2 Séquence Synthèse 2 171
B.2.1 Statistiques 171
B.2.2 Extraits de la vidéo 171
B.3 Séquence Versailles 1 172
B.3.1 Statistiques 172
B.3.2 Extraits de la vidéo 172
B.4 Séquence Versailles 2 173
B.4.1 Statistiques 173
B.4.2 Extraits de la vidéo 173
B.5 Séquence ODIAAC 174
B.5.1 Statistiques 174
B.5.2 Extraits de la vidéo 174

Bibliographie **175**

Table des figures **185**

Liste des tableaux **187**

Table des matières **194**

Résumé

Les travaux réalisés au cours de cette thèse s'inscrivent dans les problématiques de localisation d'un véhicule par vision. Nous nous plaçons en particulier dans le cas de parcours sur de longues distances, c'est à dire plusieurs kilomètres. Les méthodes actuelles de localisation et cartographie simultanées souffrent de problèmes de dérives qui les rendent difficilement exploitables après plusieurs centaines de mètres. Nous proposons dans ce mémoire de pallier ces limites en exploitant une connaissance *a priori* sur la géométrie de l'environnement parcouru. Cette information est extraite d'un Système d'Information Géographique. En particulier, les travaux réalisés se basent sur les modèles 3D des bâtiments des villes et sur une carte de la route.

Dans la première partie de ce mémoire, nous proposons une approche permettant de corriger hors ligne une reconstruction SLAM en exploitant la connaissance d'un modèle 3D simple de l'environnement. Cette correction s'applique en deux étapes. En premier lieu, un recalage non-rigide entre le nuage de points reconstruit et le modèle 3D est effectué de sorte à retrouver la cohérence globale de la reconstruction. Dans le but de raffiner le nuage de points obtenu, un ajustement de faisceaux contraint par le SIG est alors effectué sur l'ensemble de la reconstruction. La particularité de cet ajustement de faisceaux est qu'il prend implicitement en compte les contraintes géométriques apportées par le modèle 3D. La reconstruction ainsi corrigée est alors utilisée en tant que base de données pour la relocalisation en ligne d'une caméra mobile. La précision de relocalisation obtenue est en particulier suffisante pour les applications de réalité augmentée.

Dans la deuxième partie de ce mémoire, nous détaillons une solution permettant de corriger en ligne la reconstruction SLAM. Pour cela, les contraintes géométriques apportées par le SIG sont exploitées au fur et à mesure de la trajectoire du véhicule. Nous montrons tout d'abord que la connaissance de la position relative de la caméra par rapport à la route permet de corriger de façon robuste la dérive de facteur d'échelle. De plus, lorsque les contraintes géométriques sont suffisantes, la reconstruction SLAM réalisée jusqu'à l'instant courant est recalée sur le SIG. Cela permet de corriger ponctuellement la dérive observée sur la position courante de la caméra. Le processus complet permet dès lors de localiser le véhicule avec une précision semblable à celle d'un système GPS sur des trajectoires de plusieurs kilomètres.

Les deux méthodes proposées ont été testées à la fois sur des séquences de synthèse et réelles. Des résultats qualitatifs et quantitatifs sont présentés tout au long de ce mémoire.

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