N° d'ordre: 3910

THÈSE

Présentée devant

l'Université de Rennes 1

pour obtenir

le grade de : Docteur de l'Université de Rennes 1
Mention Informatique

par

Fabien SERVANT

Équipe d'accueil : IRISA - Projet Lagadic
École Doctorale : Matisse
Composante universitaire : IFSIC

Titre de la thèse :

Localisation et cartographie simultanées en vision monoculaire et en temps réel basé sur les structures planes

soutenue le 18 juin 2009 devant la commission d'examen

M. :	Bruno	ARNALDI	Président
MM. :	Vincent	CHARVILLAT	Rapporteurs
	Malik	MALLEM	
MM. :	Patrick	RIVES	Examinateurs
	Pascal	HOULIER	
	Eric	MARCHAND	

A mon neveu à venir,
en espérant que tu me dédies ta propre thèse.

Remerciements

Je souhaite tout d'abord remercier l'ensemble du jury pour m'avoir fait l'honneur de participer à ma soutenance de thèse. Je les remercie pour leurs lectures attentives de mon rapport, leurs questions et avis nombreux et constructifs.

Je souhaite remercier particulièrement mes responsables de thèse. Eric Marchand, mon directeur de thèse, pour avoir su me guider et m'aider à choisir avec clairvoyance les voies à suivre. Pascal Houlier et Isabelle Marchal, mes responsables en entreprise, pour m'avoir aidé à la bonne réalisation de cette thèse dans un cadre industriel.

Je remercie François Chaumette, Danielle Pelé et à travers eux l'ensemble de la hiérarchie de l'IRISA et de Orange pour avoir mis à ma disposition l'ensemble des ressources matérielles, technologues et scientifiques qui m'ont permis de mettre en oeuvre mes travaux de recherche. J'espère que la bonne ambiance qu'ils ont su distiller dans leur équipe, et qui a rendu ces 3 années de thèse agréables, saura perdurer avec le temps.

Je remercie mes cobureaux qui ont réussi à me supporter pendant 3 ans : Anthony S., Mohammed, Claire, Murielle, Odile, Areti, Jérome B., David A., David C., Alban, Eric V.. Nous avons partagé de bons moments et je me rappellerai de vous longtemps.

Je remercie l'encadrement administratif et technique, Céline A., Fabien Sp., Christine, sans lesquels ces années de travail auraient été bien moins faciles.

Je remercie l'ensemble de mes collègue de Lagadic et de IAM, en espérant n'en oublier aucun parmi ceux que je n'ai pas encore cité : Christophe, Nicolas Me., Céline T., Anthony R., Roméo, Fabien D., Alexandre, Xiang, Andréa, Raffik, Caroline, Olivier, Ryuta, Albert, Hai, Sinisa, Nicolas Ma., Amaury, Jean, Guillaume, David F, Christian, Gaspard, Noémie, Olivier A., David C., Jérome R., Luc, Loic, Nicolas, Sébastien, Oxanna et Philippe.

Je tiens à remercier mes deux encadrants de master, Cyril Fonlupt et Denis Robillard, pour m'avoir formé pendant mon cursus et pour avoir soutenu ma candidature pour cette thèse.

Enfin et bien sur, je tiens tout spécialement à remercier mes parents, ma soeur préférée et toute ma famille en général ainsi que mes amis pour leur soutien affectif de tous les instants. Merci à Mahendra et Marie pour avoir pu se libérer et assister à ma soutenance afin de m'aider à passer cette épreuve.

Table des matières

Notations

Règles générales

x minuscule	:	scalaire
$\mathbf{x}$ minuscule gras	:	vecteur
$\mathbf{X}$ majuscule gras	:	matrice

Mathématiques

$\mathbf{A}^\intercal$	:	matrice transposée de $\mathbf{A}$
$[\mathbf{x}]_\times$	:	matrice de préproduit vectoriel associée au vecteur $\mathbf{x}$
$\mathbf{A} \otimes \mathbf{B}$	:	produit de Kronecker entre $\mathbf{A}$ et $\mathbf{B}$
$(\mathbf{A}^s)$	:	superposition des colonnes de $\mathbf{A}$ dans un même vecteur
$det(\mathbf{A})$	:	determinant de la matrice $\mathbf{A}$
$ker(\mathbf{A})$	:	noyau de la matrice $\mathbf{A}$
$\mathbf{O}_{n\times m}$	:	matrice nulle de dimension $n \times n$ ($n = 3$ par défaut)
$\mathbf{I}_n$	:	matrice identité de dimension $n \times n$ ($n = 3$ par défaut)
$\frac{\partial \mathbf{A}}{\partial \mathbf{B}}$	:	dérivée de $\mathbf{A}$ en fonction de $\mathbf{B}$
$\mathbf{A}_{k\|y}$	:	variable aléatoire à l'instant k en fonction de l'état à l'instant y
$\mathbf{A}_{\mathbf{xy}}$	:	terme de $\mathbf{A}$ à l'intersection de la ligne x et de la colonne y

Géométrie

$\mathcal{F}_x$	:	repère cartésien x
${}^y\mathbf{x}$	:	entité géométrique $\mathbf{x}$ définie dans $\mathcal{F}_y$
${}^y\mathbf{X}_z$	:	Transformation géométrique $\mathbf{X}$ entre $\mathcal{F}_z$ et $\mathcal{F}_y$
${}^y\mathbf{R}_z$	:	matrice de rotation
${}^y\mathbf{r}_z$	:	vecteur de rotation (notation $\theta\mathbf{u}$)
${}^y\mathbf{t}_z$	:	vecteur de translation
${}^y\mathbf{E}_z$	:	matrice essentielle
${}^y\mathbf{F}_z$	:	matrice fondamentale
${}^y\mathbf{H}_z$	:	matrice d'homographie

Problématique

Introduction

L'être humain conçoit le monde extérieur et sa propre présence dans ce monde à travers des stimuli appréhendés par divers capteurs. La vue, l'ouïe et le toucher sont autant de sens permettant à notre cerveau de se faire une représentation subjective de son environnement. Nous avons donc une lecture restreinte de notre environnement qui correspond aux propriétés des stimuli mis en œuvre. Notre cerveau est incapable de discriminer deux informations dont les différences se situent en dehors de son domaine de perception. De plus, si l'information est légèrement différente ou dégradée par rapport à la réalité, le cerveau est capable inconsciemment de la classifier et de l'associer à une information précédemment acquise.

Cette limitation cognitive est utilisée depuis longtemps par l'homme pour créer des reproductions de son environnement qui soit compréhensible par les autres. Peintures et photographies sont interprétées par notre cerveau comme si les informations qui y sont inscrites étaient réelles. Dès lors, l'humanité n'a eu de cesse d'améliorer le réalisme de ces reproductions afin de tendre vers une parfaite symétrie entre la réalité et l'œuvre. Choix des couleurs, étude de la lumière et de la perspective sont autant d'avancées dans l'histoire de l'art qui ont permis d'atteindre cet objectif.

Avec l'avènement de machines capables de calculs systématiques et précis est née l'idée de la réalité virtuelle. En se basant sur les modèles physiques simplifiés du monde réel et de l'œil humain, il est possibble de synthétiser sur un écran une représentation visuelle d'un environnement donné. Bien qu'approximative et biaisée par de nombreux facteurs, nos limitations cognitives nous permettent de confondre cette représentation virtuelle avec la réalité. Les avancées dans le domaine de la réalité virtuelle ont ouvert la voie à de nombreuses innovations dans la majeure partie des domaines industriels.

Hormis les évidentes possibilités artistiques dans le domaine du cinéma et des loisirs, la réalité virtuelle permet d'élargir les possibilités d'interaction entre l'homme et la machine en fournissant une représentation adaptée et configurable des informations à manipuler. En représentant des informations n'existant pas réellement ou existant de manière différente, la réalité virtuelle peut être un outil décisionnel ou d'informations capitales.

La réalité augmentée peut-être appréhendée comme une modification de la réalité virtuelle. Alors que le principe de la réalité virtuelle est de synthétiser entièrement un environnement donné, la réalité augmentée se "contente" de synthétiser une partie de l'environnement et de l'intégrer en quelque sorte dans le monde réel. Le concept est donc d'intégrer le mieux possible l'information virtuelle afin qu'elle soit la plus informative possible pour l'utilisateur. La plupart du temps, on souhaitera intégrer un objet virtuel dans la réalité de telle manière qu'il semble toujours être à un endroit donné.

FIGURE 1 – Vues d'artistes de la réalité augmentée et ses applications

La fusion du virtuel et de la réalité implique que deux sources d'informations soient affichées simultanément. Dans le cas d'application temps-réel, certains utilisent des lunettes see-through permettant de voir la réalité à travers des verres semi-transparents, les éventuelles informations virtuelles étant projetées sur ces lunettes par un dispositif interne. La plupart du temps cependant, pour les applications non temps-réel, ou pour des raisons de praticité, la scène est filmée avec une caméra et l'image est augmentée par les informations virtuelles avant d'être affichée et/ou stockée.

La réalité augmentée permet donc comme la réalité virtuelle de faciliter l'interaction entre l'homme et la machine, en lui fournissant des informations supplémentaires qu'il peut interpréter sans nécessité d'abstraction comme une description textuelle. Encore une fois, nos capacités cognitives nous permettent d'appréhender ces informations visuelles de manière inconsciente comme si elles étaient réelles et donc de les interpréter sans difficultés supplémentaires, à condition, évidemment, que l'intégration des données soit suffisamment bonne.

Problématique industrielle

Cette thèse est effectuée dans le cadre d'une convention CIFRE entre l'IRISA (en tant que laboratoire universitaire) et la société Orange Labs, filiale de recherche et développement du groupe France Telecom. Cette dernière société est, à l'origine, dédiée à la mise en place de moyens de communications permettant le transfert de voix et de

données. Avec l'arrivée à la fin du XXème siècle de l'ADSL et de la téléphonie mobile grand public, le groupe a procédé à une mutation de ses activités pour cesser d'être uniquement un support technique pour services. Le groupe s'est attaché à développer les services utilisant ces nouvelles technologies. Dans le cadre des services mobiles, il est important de pouvoir fournir en temps-réel des informations localisées et adaptées à l'utilisateur, que ce soit dans un but commercial (telle que l'affichage d'informations publicitaires) ou non.

La réalité augmentée mobile est un moyen intéressant d'offrir à l'utilisateur du contenu localisé sous une forme adaptée. La société d'études Gartner a d'ailleurs identifié la réalité augmentée comme l'une des dix "disruptive technologies" pour la période 2008-2012 (*Gartner defines a disruptive technology as one that causes major changes in 'the accepted way of doing things', including business models, processes, revenue streams, industry dynamics and consumer behaviour*). Les appareils mobiles haut de gamme disposant déjà de caméras de qualité correcte, il ne reste qu'a attendre une augmentation des capacités calculatoires des appareils mobiles qui devrait survenir très prochainement (avec notamment l'arrivée de coprocesseurs graphiques intégrés et de processeurs à virgule flottante).

L'objectif industriel est donc de trouver de nouvelles méthodes permettant l'augmentation de la réalité sur appareils mobiles dans des environnements urbains ou intérieurs. Les contraintes apportées étaient entre autres que la méthode fonctionne avec des caméras de qualité moyenne, avec un minimum de ressources calculatoires et qu'elle permette un fonctionnement dans un environnement synthétique (par opposition à naturel). En partenariat avec la ville de Rennes et de son musée des beaux arts, la première application grand public devrait être constituée par le projet ANR RIAM GAMME. Ce projet consiste en la réalisation d'un guide augmenté pour la visite de musées.

Problématique scientifique

De multiples questions se posent quant à la réalisation et à l'utilisation d'un système de réalité augmentée, notamment au niveau des interactions utilisateurs, des contraintes techniques et d'ergonomie. Malgré tout, quel que soit le type d'application ciblée, le problème majeur est de savoir comment augmenter la séquence vidéo avec les informations souhaitées. En effet, cette dernière doit être intégrée de manière à avoir une signification évidente.

Dans le type d'application qui nous intéresse, on souhaitera simuler le fait que l'information est située à un point donné de l'environnement comme s'il s'agissait d'un objet réel immobile. L'affichage de l'information (sa position et son orientation) dépendra donc du point de vue de l'utilisateur. Si on considère que l'objet est affiché suivant les paramètres de position et d'orientation d'une caméra virtuelle, la problématique principale (et celle qui nous intéresse parmi d'autres non moins importantes comme la gestion de l'occlusion) et donc d'aligner cette caméra virtuelle avec la caméra réelle. Cela revient à répondre à la question "**Quelle est la position et l'orientation (la**

pose) de la caméra dans un repère donné ?" à chaque image de la séquence capturée par la caméra en temps réel. Par l'analyse du flux d'images et des mouvements dans celles-ci, l'objectif est d'en extraire les mouvements de la caméra elle-même.

La problématique industrielle est de faire fonctionner l'application dans un environnement urbain connu pour être majoritairement constitué de surfaces planes. L'idée de départ est de profiter de ces plans pour faciliter l'estimation de la pose de la caméra.

Le travail de cette thèse peut être découpé en 4 parties principales qui sont autant d'étages dans le processus aboutissant à l'estimation de la pose.

- Le suivi dans l'image de certaines zones planaires. Cette étape est l'interface entre le flux d'images et l'estimation de la pose. Elle fournit à ce dernier les paramètres du déplacement dans l'image des zones suivies.
- L'estimateur de pose en lui-même est décrit. Conçu pour appréhender les incertitudes dans les mesures et dans la connaissance de l'environnement, l'estimateur statistique développé est basé sur les principes du SLAM monoculaire et adapté à l'utilisation de plans.
- La thèse décrit comment sélectionner les zones planaires de l'image avec différentes méthodes qui dépendent des connaissances *a priori* de la scène. Cette partie décrit également comment initialiser ces zones afin qu'elles puissent être introduites automatiquement dans l'estimateur.
- Dans une optique de robustesse, la quatrième partie s'attachera à décrire l'utilisation d'une centrale inertielle en complément de la caméra. Cette centrale inertielle permet de compenser un certain nombre de problèmes liés à l'utilisation de la caméra. Les méthodes de calibration et de fusion des informations seront détaillées.

Cette thèse propose une solution complètement automatique ne nécessitant pas d'intervention de l'utilisateur pour son initialisation. Il sera montré que sa conception est axée sur la robustesse aux bruits des mesures et sur la réduction des coûts calculatoires par rapport aux méthodes de même type.

Chapitre 1

Éléments de la vision par ordinateur

La problématique que l'on souhaite étudier est le calcul de la pose du système, ou plus précisément l'estimation de son déplacement en temps réel par l'analyse d'images capturées par une caméra rigidement liée à ce système.

L'estimation du déplacement nécessite par construction que le système soit à même de mesurer son mouvement soit par observation de son environnement, soit en utilisant des capteurs de mouvements. On verra dans le chapitre 5 que les capteurs de mouvements abordables d'un point de vue financier et portables ne peuvent fournir une estimation du déplacement adéquate et fiable s'ils sont utilisés seuls. Il reste donc les capteurs exteroceptifs. Quel que soit le type de capteur utilisé, l'estimation du déplacement consistera à analyser les variations des mesures au cours du temps. La question du choix du capteur et donc de la mesure nécessite de prendre en compte les contraintes liées à l'application. Dans un contexte robotique/industriel, les capteurs de prédilection sont les lasers et autres lidars qui fournissent une mesure riche en informations 3D et de bonne qualité. Nonobstant ces qualités, ce type de capteurs n'est pas acceptable dans notre contexte applicatif pour plusieurs raisons primordiales : le coût et la difficulté d'utilisation. Cette difficulté est causée par le fait que ces capteurs émettent des signaux potentiellement destructeurs/dangereux et consomment plus qu'un capteur passif. De plus, ces capteurs sont difficilement utilisables en extérieur puisqu'ils nécessitent une calibration *a priori* qui dépend des dimensions mesurées. Par opposition, les caméras possèdent les qualités nécessaires à l'utilisation du système par une personne en situation de mobilité.

Première caractéristique et non des moindres, la caméra a un très faible coût. *A fortiori* dans notre cas où elle est nécessaire au bon fonctionnement du pipeline de réalité augmentée, ses qualités économiques sont incomparables aux autres capteurs. Il faudra cependant utiliser une caméra de qualité suffisante avec le moins de défauts images possible pour faciliter le suivi et donc éviter les capteurs caméra de très faible

coût. Un autre avantage est que la caméra est un capteur passif : elle n'émet directement aucune énergie pour son fonctionnement. Ce type de capteur est donc plus économe en énergie mais aussi moins délicat à utiliser dans des environnements fragiles comme des musées.

Quant aux mesures fournies par la caméra, elles possèdent plusieurs propriétés intéressantes. Habituellement 15/30 fois par seconde, une information sur tout ce qui se situe devant la caméra est obtenue simultanément. Ces mesures donnent non seulement des informations géométriques (indirectement) comme sur les autres capteurs mais également des informations sur les propriétés des matériaux que l'on observe. On verra que cela facilite la reconnaissance et le suivi d'éléments dans l'image. Pour des raisons pratiques et techniques, une seule caméra est utilisée. Les problématiques de vision multi-caméras ne sont donc pas traitées dans ce document.

Ce chapitre va permettre de présenter au lecteur les principes du capteur caméra et les modélisations mathématiques associées. Ce chapitre introduit également la problématique de la géométrie multi-vues qui découle de l'utilisation de deux caméras ou d'un flux d'images. Tout ceci permettra de comprendre les méthodes développées et présentées dans les chapitres ultérieurs. Les problématiques de traitement d'image (comme l'extraction des régions d'intérêt) à proprement parler seront traitées dans un autre chapitre et sont considérées, pour le moment, comme acquises.

1.1 Contraintes et limitations des caméras

Une caméra est en quelque sorte une matrice rectangulaire de capteurs qui mesurent la lumière réfléchie par les objets de son environnement. Cette matrice est la mesure image fournie par la caméra : chaque cellule contient une information sur la couleur reçue par un capteur à cette coordonnée. Aucune information sur la profondeur de l'élément réfléchissant n'est fournie par la mesure, on parle de projection en deux dimensions. C'est une problématique de première importance pour nous puisque cela signifie que la localisation par rapport à un objet mesuré n'est pas possible directement. La problématique théorique de la vision par ordinateur est principalement de contourner cette limitation en utilisant soit plusieurs caméras, soit plusieurs vues d'un même objet afin d'obtenir plus d'informations sur celui-ci.

On verra par la suite que l'estimation du déplacement de la caméra nécessite de suivre ou de localiser dans l'image des "objets". Pour que ce suivi soit possible, il faut que la représentation de cet objet dans le flux d'image reste cohérent dans le temps (par exemple que sa couleur ne change pas). Cette consistance est mise en défaut par les propriétés intrinsèques de la caméra et des modèles d'illumination. Premièrement, la caméra fournit une mesure bruitée. Ces bruits sont liés à différents facteurs, notamment aux défauts de certains éléments de la caméra.

Deuxièmement, la mesure fournie par la caméra est une mesure discrète dont la précision dépend non seulement de la résolution de la caméra mais également de la distance de la scène. Ce manque de précision peut provoquer des changements dans la

projection d'un objet suivant le point de vue.

Troisièmement, chaque cellule de la caméra doit mesurer pendant un laps de temps la couleur reçue ("le temps d'ouverture"). Si, par un mouvement de la caméra ou de l'objet trop rapide, l'élément mesuré par la cellule change pendant ce laps de temps, cette dernière mesure en réalité plusieurs choses et mélange les informations. On parle alors de flou de bougé. Ce flou rend les objets de la scène difficilement identifiables et va la plupart du temps conduire à une erreur de suivi dans l'image.

Dernier point de cette liste non exhaustive : la caméra mesurant la lumière, la mesure est donc très sensible aux conditions d'éclairage. Une scène peut devenir très différente si les sources de lumières changent. Cette difficulté est encore plus prononcée dans les cas où l'objet dans la scène a une surface spéculaire. Dans ce cas, sa couleur change partiellement en fonction du point de vue de la caméra, ce qui est bien évidemment peu souhaitable. Ces changements sont de plus difficilement modélisables puisqu'ils dépendent d'un nombre important de variables. Les problèmes d'intensité lumineuse sont renforcés par le fait qu'une caméra standard est calibrée pour mesurer une certaine étendue d'intensité lumineuse. Les endroits dont l'intensité lumineuse est trop importante vont être écrasés (saturés) à la valeur maximale admise par la caméra. Ce type de phénomène de saturation de l'image se produit fréquemment avec les caméras bas coût que nous utilisons lorsque la caméra fixe une surface réfléchissante ou une source lumineuse. Cela a pour effet de modifier de manière destructrice (non inversible) la représentation de l'objet dans l'image.

Toutes ces difficultés soulevées ne rendent pas caduc le choix de l'utilisation de la caméra dans notre contexte applicatif. Ils démontrent toutefois qu'il est nécessaire d'utiliser dans la mesure du possible des algorithmes robustes qui prennent en compte les éventuelles pertes de suivi. On considérera dans la suite de ce chapitre que la mesure fournie par la caméra ne souffre pas des défauts décrits et qu'on est capable d'observer le même objet à travers plusieurs vues.

1.2 Géométrie de la prise de vue

L'utilisation des mesures d'une caméra pour la compréhension géométrique de la scène filmée nécessite de savoir d'où ces mesures proviennent, ou plus précisément comment on a obtenu ces mesures. Si on est capable de modéliser la manière dont l'environnement se projette et devient une image, il est alors plus facile de déduire des mesures des informations sur l'environnement en inversant ce modèle.

Une caméra peut-être modélisée en simplifiant par un modèle de projection perspective. C'est le modèle le plus couramment utilisé dans la littérature et modélise relativement bien le principe de fonctionnement d'une caméra. L'objectif de la caméra est considéré alors comme un point $\mathcal{C}$, le centre optique, par lequel passe l'ensemble des rayons lumineux réfléchis par la scène qui vont impacter notre capteur. Chaque point $\mathbf{X}$ de la scène se situant devant la caméra est alors représenté par un rayon qui passe

par $\mathbf{X}$ et $\mathcal{C}$. L'intersection de ce rayon et de l'image $\mathcal{I}$ est la projection de ce point dans l'image. La figure 1.1 illustre ce modèle.

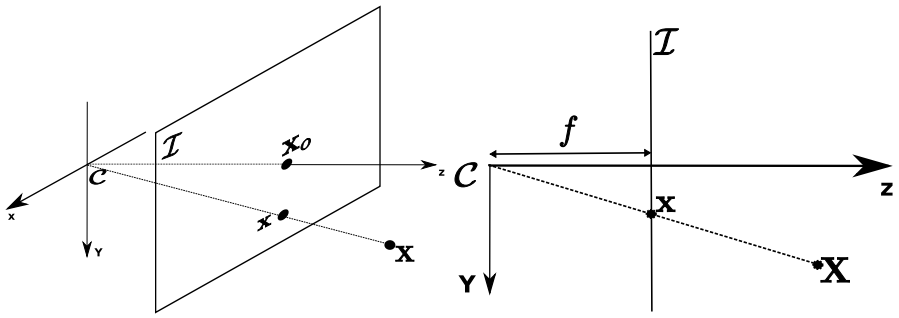


FIGURE 1.1 – Modèle Pin-Hole

La caméra peut être considérée comme un repère cartésien $\mathcal{F}_c$ d'origine $\mathcal{C}$ et de base $(\mathbf{x}, \mathbf{y}, \mathbf{z})$. Le point $\mathbf{X}$ est alors un vecteur de coordonnées

$$\mathbf{X} = \begin{bmatrix} {}^cX & {}^cY & {}^cZ \end{bmatrix}^{\mathtt{T}} \tag{1.1}$$

La point $\mathbf{x}$ est une projection de $\mathbf{X}$ vers $\mathcal{C}$. $\mathbf{x}$ a donc la même direction que $\mathbf{X}$.

$$\mathbf{x} = \lambda \begin{bmatrix} {}^cX & {}^cY & {}^cZ \end{bmatrix}^{\mathtt{T}} \tag{1.2}$$

On voit que la profondeur de $\mathbf{x}$ n'est plus ${}^cZ$ mais la focale de la caméra f qui détermine la distance de l'image à $\mathcal{C}$. On en déduit que

$$\lambda = \frac{f}{{}^cZ} \tag{1.3}$$

Les coordonnées du point projeté dans $\mathcal{F}_c$ sont donc

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1.4}$$

$$= \begin{bmatrix} f\frac{{}^cX}{{}^cZ} \\ f\frac{{}^cY}{{}^cZ} \\ 1 \end{bmatrix} \tag{1.5}$$

$\mathbf{x}$ représente la mesure en mètres du point $\mathbf{X}$. Pour connaitre la mesure fournie par la caméra, on doit convertir le point en pixel. En effet, la mesure image a pour unité de taille les pixels et non plus les mètres comme l'espace mesuré. On doit donc mettre à

l'échelle les informations afin qu'ils utilisent la bonne métrique. Chaque pixel correspond à un nombre de mètres s_x le long de l'axe x et s_y le long de l'axe y. L'information en mètre ayant pour origine le point principal $\mathbf{x}_0$ (l'intersection de $\mathcal{I}$ et de l'axe z), on note les coordonnées en pixel de $\mathbf{x}_0$ par rapport à l'origine de $\mathcal{I}$ (u_0,v_0). Le passage en mètre s'écrit alors

$$u = u_0 + f\frac{x}{s_x} \tag{1.6}$$

$$v = v_0 + f\frac{y}{s_y} \tag{1.7}$$

On regroupe les informations propres à la caméra dans la matrice des paramètres intrinsèques $\mathbf{K}$. Soit $p_x = \frac{f}{s_x}$ et $p_y = \frac{f}{s_y}$.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1.8}$$

$$\mathbf{K} = \begin{bmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.9}$$

Cette modélisation de la caméra est une modélisation basique qui ne prend pas en compte les déformations et défauts éventuels de l'objectif. Elle suppose que les axes de l'image sont parallèles à ceux de la scène et qu'il n'y a pas de distorsion. Elle ne prend pas du tout en compte les distorsions non linéaires comme la distorsion radiale. Cependant on observe qu'elle est suffisante dans un grand nombre d'applications. La mesure caméra étant discrète, toutes les informations projetées dans la même coordonnée pixel (quelles que soient les décimales) sont mélangées. L'inversion est rendue d'autant moins précise par cette discrétisation que l'objet mesuré est distant de la caméra. Dans la plupart des méthodes d'estimation de déplacement (tout comme dans notre méthode) la matrice $\mathbf{K}$ est supposée connue à l'avance. On parle d'étalonnage hors ligne de la caméra [Tsai 86, Brown 71].

1.3 Transformation de repère

Dans l'équation (1.5), on observe que l'objet doit être défini dans le repère de la caméra courante. Il arrive le plus souvent que ce ne soit pas le cas, l'objet étant souvent défini dans un repère global qui ne dépend pas de la position de la caméra. Il est alors nécessaire de pouvoir transformer l'information géométrique du repère global à celui de la caméra. D'une manière générale, il est régulièrement souhaitable de pouvoir transformer des données d'un repère à un autre. La transformation d'un point dans un espace euclidien de dimension 3 est étudiée, les autres formes géométriques pouvant en être déduites dans le même type d'espace.

Soit un repère global $\mathcal{F}_o$ et le repère caméra $\mathcal{F}_c$. Un point ${}^o\mathbf{X}$ défini dans $\mathcal{F}_o$ peut être exprimé dans $\mathcal{F}_c$ à travers la fonction affine

$$ {}^c\mathbf{X} = {}^c\mathbf{R}_o {}^o\mathbf{X} + {}^c\mathbf{t}_o \tag{1.10} $$

ou, en notation homogène :

$$ {}^c\mathbf{X} = {}^c\mathbf{M}_o {}^o\mathbf{X} \tag{1.11} $$

avec la matrice homogène ${}^c\mathbf{M}_o$ définie par :

$$ {}^c\mathbf{M}_o = \begin{bmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{1.12} $$

Cette transformation est valide pour tout point défini dans le repère $\mathcal{F}_o$. Le point subit d'abord une rotation autour de l'origine de $\mathcal{F}_o$ à travers la matrice de rotation ${}^c\mathbf{R}_o$ et est déplacé par le vecteur de translation (de dimension 3) ${}^c\mathbf{t}_o$. Alors que ce dernier est un vecteur quelconque défini dans $\mathbb{R}_3$, la matrice de rotation doit respecter une structure spéciale.

En effet, une rotation doit conserver l'orthogonalité et les dimensions de l'espace concerné. Plus directement, la norme du vecteur sur lequel on applique la rotation doit rester inchangée. la matrice de rotation appartient au groupe *spécial orthogonal* de dimension 3, ou $\mathbf{SO}(3)$, qui respecte ces contraintes :

$$ det(\mathbf{R}) = 1 \tag{1.13} $$

$$ \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I} \tag{1.14} $$

Une rotation peut être obtenue à l'aide de 3 angles et donc seulement 3 paramètres indépendants. De plus, les contraintes sur la forme de la matrice rendent difficile la manipulation numérique de celle-ci notamment dans les problématiques d'estimation puisque la valeur de chaque paramètre dépend évidemment des autres paramètres.

La représentation utilisée dans cette thèse est celle communément appellée $\theta\mathbf{u}$. Soit un vecteur $\mathbf{u}$, unitaire, de dimension 3 et θ un scalaire. Toute rotation peut être considérée comme la rotation de θ radians autour d'un axe de rotation $\mathbf{u}$. Le vecteur de rotation est alors obtenu en multipliant $\mathbf{u}$ par θ. On parle alors de l'algèbre du groupe *spécial orthogonal* ($\mathbf{so}(3)$).

Le passage de $\mathbf{so}(3)$ à $\mathbf{SO}(3)$ se fait par l'exponentielle

$$ \mathbf{R} = e^{[\theta\mathbf{u}]_\times} \tag{1.15} $$

où $[\mathbf{a}]_\times$ est la matrice antisymmétrique associée à $\mathbf{a}$ et est donné par :

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -\mathbf{a}_3 & \mathbf{a}_2 \\ \mathbf{a}_3 & 0 & -\mathbf{a}_1 \\ -\mathbf{a}_2 & \mathbf{a}_1 & 0 \end{bmatrix} \tag{1.16}$$

L'exponentielle d'une matrice générique est une série infinie dont le calcul numérique est relativement complexe et surtout approximatif. Dans le cas de $\mathbf{so}(3)$, un résultat exact est obtenue à travers la formule de Rodrigues. Cette méthode, qui s'explique géométriquement, est définie par la fonction R :

$$R(\theta\mathbf{u}) = \mathbf{I} + \frac{sin(\theta)}{\theta}[\theta\mathbf{u}]_\times + \frac{1 - cos(\theta)}{\theta^2}[\theta\mathbf{u}]_\times^2 \tag{1.17}$$

On observe que lorsque $||\theta\mathbf{u}||$ est nulle, la fonction R est mal définie numériquement. On considère alors que

$$R(\mathbf{0}) = \mathbf{I} \tag{1.18}$$

Inversement, il est parfois nécessaire de connaitre la représentation $||\theta\mathbf{u}||$ d'une matrice de rotation $\mathbf{M}$ donnée. Ce passage de $\mathbf{SO}(3)$ à $\mathbf{so}(3)$ se fait par l'inverse de l'exponentielle, le logarithme matriciel. Dans le cas de $\mathbf{SO}(3)$, ce logarithme est défini par la fonction ϕ :

$$\theta = cos^{-1}(\frac{(trace(\mathbf{M}) - \mathbf{1})}{2}) \tag{1.19}$$

$$\phi(\mathbf{M}) = \frac{\theta}{2sin(\theta)} \begin{bmatrix} \mathbf{M}_{3,2} - \mathbf{M}_{2,3} \\ \mathbf{M}_{1,3} - \mathbf{M}_{3,1} \\ \mathbf{M}_{2,1} - \mathbf{M}_{1,2} \end{bmatrix} \tag{1.20}$$

1.4 Transformation entre plusieurs vues

Il peut être parfois utile de déduire d'une mesure prise dans une première vue (repère $\mathcal{F}_{c1}$) un a priori sur la position du même objet dans une seconde vue (repère $\mathcal{F}_{c2}$). La projection dans l'image d'un point $\mathbf{X}$ a pour conséquence de perdre l'information sur sa profondeur. Les paramètres de $\mathbf{X}$ ne sont donc pas connus dans $\mathcal{F}_{c1}$ et il est donc impossible d'estimer une transformation de ce point dans $\mathcal{F}_{c2}$ si d'autres informations sur celui-ci ne sont pas disponibles.

1.4.1 La contrainte épipolaire

Une mesure ${}^{c1}\mathbf{x}$ dans une image étant une projection dans un plan d'un point 3D $\mathbf{X}$ vers le centre de projection $\mathcal{C}$, tout point $\mathbf{X}$ situé sur la droite reliant $\mathcal{C}$ et $\mathbf{x}$ peut-être

considéré comme le point mesuré. Il est donc possible de considérer la mesure $\mathbf{x}$ comme une droite reliant $\mathcal{C}$ et $\mathbf{x}$ dans $\mathcal{F}_{c1}$. Cette droite peut-être reprojetée dans $\mathcal{F}_{c2}$ puisque ses paramètres sont connus dans $\mathcal{F}_{c1}$.

En conséquence, si le point est visible dans l'image, sa position ${}^{c2}\mathbf{x}$ est nécessairement le long de la projection de cette droite dans l'image. Le fait que le point soit forcément devant l'objectif de la caméra permet d'éliminer tous les points situés avant la projection e de l'origine de $\mathcal{F}_{c1}$ dans l'image de $\mathcal{F}_{c2}$. La figure 1.2 illustre cette contrainte.

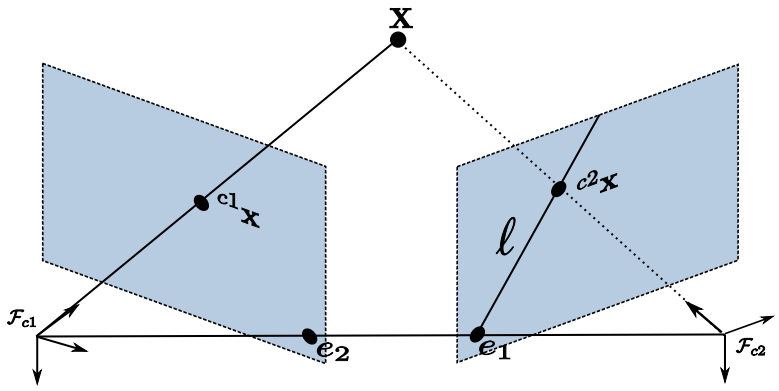


FIGURE 1.2 – La contrainte épipolaire

Cette contrainte, communément appelée contrainte épipolaire, permet généralement de limiter l'espace de recherche lors de la mise en correspondance d'un point dans deux images. Cette contrainte peut être vérifiée analytiquement par l'équation

$$ {}^{c2}\mathbf{x}^{\mathrm{T}}{}^{c2}\mathbf{E}_{c1}{}^{c1}\mathbf{x} \;=\; 0 \tag{1.21} $$

${}^{c2}\mathbf{E}_{c1}$ est appelée la matrice essentielle et est construite en fonction des paramètres de transformation entre ${}^{c1}\mathbf{x}$ et ${}^{c2}\mathbf{x}$

$$ {}^{c2}\mathbf{E}_{c1} \;=\; [{}^{c2}\mathbf{t}_{c1}]_{\times}{}^{c2}\mathbf{R}_{c1} \tag{1.22} $$

Notons que dans le cas d'une rotation pure, ${}^{c2}\mathbf{t}_{c1}$ est nul. Dans ce cas, la matrice essentielle n'est pas définie. Cette matrice essentielle est définie pour les points mesurés en mètres. Pour connaitre la même relation appliquée aux points dans l'image, et donc en pixels, la matrice fondamentale ${}^{c2}\mathbf{F}_{c1}$ doit être utilisée en lieu et place de la matrice essentielle :

$$ {}^{c2}\mathbf{F}_{c1} \;=\; \mathbf{K}^{\mathrm{T}}{}^{c2}\mathbf{E}_{c1}\mathbf{K} \tag{1.23} $$

1.4.2 La relation d'homographie

L'impossibilité d'estimer la mesure ${}^{c2}\mathbf{x}$ tient dans la perte de l'information de profondeur. Si le point est situé sur un plan Π dont les paramètres sont connus dans $\mathcal{F}_{c1}$, le point 3D ${}^{c1}\mathbf{X}$ peut être estimé comme l'intersection de Π et de la droite passant par ${}^{c1}\mathbf{X}$ et $\mathcal{C}$. Le point 3D étant connu, on peut alors estimer la mesure dans $\mathcal{F}_{c2}$.

Considérons Π défini par un vecteur normal ${}^{c1}\mathbf{n}$ qui détermine l'orientation du plan, et par ${}^{c1}d$ la distance orthogonale du plan à l'origine du repère. Puisque ${}^{c1}\mathbf{X}$ appartient au plan Π, l'équation suivante est valide :

$$
\begin{aligned}
{}^{c1}\mathbf{n}^{\mathsf{T}}{}^{c1}\mathbf{X} + {}^{c1}d &= 0 && (1.24)\\
\frac{-1}{{}^{c1}d}{}^{c1}\mathbf{n}^{\mathsf{T}}{}^{c1}\mathbf{X} &= 1 && (1.25)
\end{aligned}
$$

L'équation de transformation de ${}^{c1}\mathbf{X}$ dans $\mathcal{F}_{c2}$ est alors transformée en

$$
\begin{aligned}
{}^{c2}\mathbf{X} &= {}^{c2}\mathbf{R}_{c1}{}^{c1}\mathbf{X} + {}^{c2}\mathbf{t}_{c1}(\frac{-1}{{}^{c1}d}{}^{c1}\mathbf{n}^{\mathsf{T}}{}^{c1}\mathbf{X}) && (1.26)\\
{}^{c2}\mathbf{X} &= {}^{c2}\mathbf{H}_{c1}{}^{c1}\mathbf{X} && (1.27)
\end{aligned}
$$

où

$$
{}^{c2}\mathbf{H}_{c1} = {}^{c2}\mathbf{R}_{c1} - \frac{1}{{}^{c1}d}{}^{c2}\mathbf{t}_{c1}{}^{c1}\mathbf{n}^{\mathsf{T}} \quad (1.28)
$$

Puisque $\mathbf{x}c1$ et ${}^{c2}\mathbf{x}$ sont les projections perspectives de ${}^{c1}\mathbf{X}$ et ${}^{c2}\mathbf{X}$:

$$
\begin{aligned}
{}^{c1}\mathbf{X} &= \lambda_1{}^{c1}\mathbf{x} && (1.29)\\
{}^{c2}\mathbf{X} &= \lambda_2{}^{c2}\mathbf{x} && (1.30)\\
\lambda_2{}^{c2}\mathbf{x} &= {}^{c2}\mathbf{H}_{c1}\lambda_1{}^{c1}\mathbf{x} && (1.31)
\end{aligned}
$$

La mesure ${}^{c2}\mathbf{x}$ est donc connue à un facteur d'échelle près.

$$
{}^{c2}\mathbf{x} \sim {}^{c2}\mathbf{H}_{c1}{}^{c1}\mathbf{x} \quad (1.32)
$$

${}^{c2}\mathbf{H}_{c1}$ est appelée la matrice d'homographie. L'intérêt de définir cette matrice est qu'elle pourra être estimée par divers moyens quand bien même les paramètres de déplacement de la caméra et du plan ne sont pas connus. En connaissant cette matrice d'homographie, la transformation des mesures d'un point appartenant à un plan entre deux vues est possible alors que la transformation entre les deux caméras n'est pas directement connue. Notons que dans le cas où la transformation entre $\mathcal{F}_{c1}$ et $\mathcal{F}_{c2}$ est une rotation pure ou que le plan est à l'infini (${}^{c1}d$ est très grand par rapport à ${}^{c2}\mathbf{t}_{c1}$), le terme $\frac{1}{{}^{c1}d}{}^{c2}\mathbf{T}_{c1}{}^{c1}\mathbf{n}^{\mathsf{T}}$ s'annule et l'homographie est indépendante des paramètres du plan.

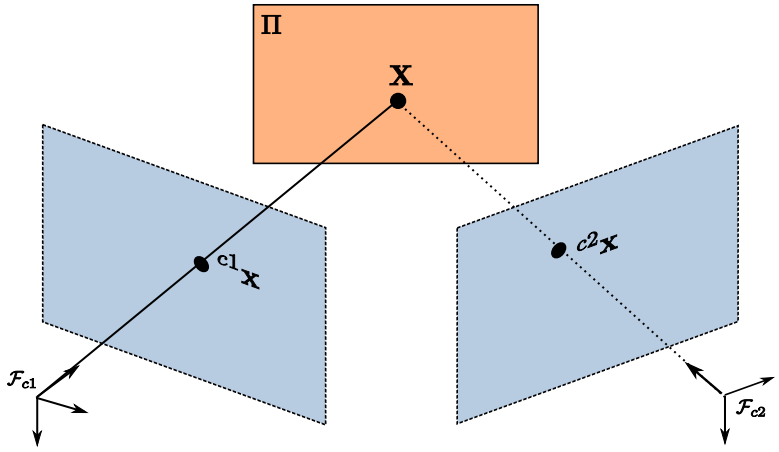


FIGURE 1.3 – La relation d'homographie

Cette matrice d'homographie est fondamentale dans la compréhension de cette thèse puisqu'on s'intéresse à des objets plans.

1.5 Le calcul de pose

Le calcul de pose consiste à connaitre la transformation entre un repère global $\mathcal{F}_o$ et le repère de la caméra $\mathcal{F}_c$ en fonction des mesures fournies par la caméra. En connaissant cette transformation, les paramètres des points de l'objet dans le repère $\mathcal{F}_o$ et le modèle de projection défini dans ce chapitre, nous avons vu que la mesure pouvait être estimée par

$$ {}^c\mathbf{x} \quad = \quad {}^c\mathbf{R}_o \, {}^o\mathbf{X} + {}^c\mathbf{t}_o \tag{1.33} $$

Dans le cas où la mesure et les paramètres de l'objet dans $\mathcal{F}_o$ sont connus, la problématique est de trouver les paramètres de transformation qui permettent d'obtenir ${}^c\mathbf{x}$ en fonction de ${}^o\mathbf{X}$. Cette transformation est estimable par la minimisation

$$ {}^c\mathbf{M}_o \quad = \quad \arg\min_{{}^c\mathbf{R}_o, {}^c\mathbf{t}_o} \sum_{i=1}^{N} ({}^c\mathbf{x}_i - \mathbf{K}\mathbf{C}({}^c\mathbf{R}_o \, {}^o\mathbf{X}_i + {}^c\mathbf{t}_o))^2 \tag{1.34} $$

$$ \mathbf{C} \quad = \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.35} $$

où N étant le nombre de points. Par souci de clarté, on s'intéresse aux primitives géométriques les plus simples et les plus utilisées : les points. Il est cependant possible d'appliquer cette logique à des primitives plus complexes [Dhome 89, Lowe 87, Safaee-Rad 92,

Dhome 90] voire à différents types de primitives simultanément [Phong 95, Marchand 02a] ou même à des objets 3D. Cette dernière catégorie consiste à utiliser un modèle CAO de l'objet et à comparer les distances entre les contours du modèle reprojeté et les contours détectés dans l'image [Lowe 91, Drummond 02, Martin 02, Simon 98, Comport 06].

La résolution du système (1.34) est une problématique importante qui a fait l'objet d'un grand nombre de recherches. Historiquement, des méthodes analytiques ou "linéaires" existent et permettent de résoudre le calcul de pose [Fischler 81, Faugeras 87, Horaud 89, Dhome 89, Liu 90] mais sont très sensibles aux bruits de mesures.

Dans un souci de robustesse et d'optimalité, les méthodes de minimisation non linéaires de type Newton doivent être préférées. La minimisation de la différence (entre la projection des primitives et les mesures) est la plupart du temps effectuée à l'aide d'un algorithme du type Newton-Raphson ou Levenberg-Marquardt [Fischler 81, Horaud 89, Dhome 89]. D'autres méthodes similaires, mais basées sur les principes et la littérature de l'asservissement visuel, permettent également d'effectuer cette minimisation [Marchand 02a, Comport 06].

Comme toute méthode de minimisation non linéaire, ces méthodes assurent un bon fonctionnement mais nécessitent d'être initialisées avec des paramètres proches de la solution. En effet, ces méthodes sont peu robustes aux problèmes de minimums locaux. Une solution est proposée par [Dementhon 95, Oberkampf 96] : l'algorithme s'initialise en considérant dans un premier temps le modèle de projection non plus perspective mais orthographique à l'échelle (la minimisation est alors linéaire puisqu'il s'agit d'un modèle affine).

Géométriquement, la difficulté tient dans la nécessité de connaitre les paramètres géométriques de l'objet dans $\mathcal{F}_o$. Cela nécessite premièrement un modèle de l'objet et que l'on connaisse la transformation entre le repère de l'objet (noté $\mathcal{F}_b$) et le repère $\mathcal{F}_o$. Dans le cas où la pose est toujours calculée à l'aide du même objet, on peut aligner $\mathcal{F}_b$ et $\mathcal{F}_o$. La pose sera alors relative au repère $\mathcal{F}_b$. Dans les autres cas, la nécessité de connaitre la pose relative des objets peut-être très contraignante. Une erreur dans la modélisation de la scène implique directement une erreur dans le calcul de la pose. La réduction des besoins de précision du modèle voire la suppression de tout modèle peut donc être un atout non négligeable dans le cadre d'un système mobile et fait d'ailleurs partie des apports de la méthode proposée dans la suite de cette thèse.

Le calcul de pose est une problématique intrinsèquement mono-image. La pose est recalculée pour chaque image. Une autre possibilité est de ne plus chercher à connaitre une pose par rapport à un repère fixe, mais de connaitre la pose relative – la transformation – entre une vue d'origine $\mathcal{F}_{c1}$ et une vue courante $\mathcal{F}_{c2}$ au fur et à mesure. Si on considère une séquence continue d'images, l'intégration des transformations entre chaque image de la séquence permet d'estimer la transformation entre la première et la dernière vue. On pourra alors éventuellement connaitre la pose de $\mathcal{F}_{c2}$ si la pose de $\mathcal{F}_{c1}$ est connue *a priori*.

L'intégration des transformations entre chaque image provoque inévitablement une accumulation des erreurs dans le résultat final. Lorsque le même "objet" est observé, cette accumulation peut être corrigée en vérifiant la cohérence de la transformation de la caméra par rapport à une image de référence. Lorsque les "objets" se multiplient, cette correction n'est plus forcément possible à l'échelle de la séquence. Une accumulation d'erreurs difficile à prendre en compte vient entacher l'estimation.

Le chapitre suivant explique comment notre méthode évite cet écueil en prenant en compte l'erreur statistique des différentes transformations estimées au cours du temps.

Chapitre 2

Localisation et cartographie simultanées basé sur des structures planes

Cette thèse adresse le problème de la localisation d'un système sur le long terme dans un environnement urbain ou d'intérieur dont la structure est inconnue *a priori*. Ce système, porté à la main par un utilisateur humain, doit être capable de continuer à fonctionner quels que soient les mouvements effectués par celui-ci. Le fonctionnement doit donc être entièrement automatique et nécessite donc que l'algorithme soit capable non seulement de chercher tout seul dans les images les informations nécessaires mais également de corriger si nécessaire ses éventuelles erreurs d'estimation.

L'estimation du déplacement de la caméra implique le suivi dans l'image d'un ou plusieurs éléments de l'environnement. Un contexte d'utilisation réaliste implique une utilisation sur le long terme et, implicitement, que les objets observés par la caméra ne soient pas les mêmes tout le long de l'exécution. L'algorithme utilisé doit donc être capable d'appréhender l'apparition de nouveaux éléments à suivre et d'intégrer les informations qu'ils fournissent pour le processus de localisation. En étant à même d'estimer l'incertitude sur l'estimation des paramètres de l'environnement, les méthodes de type SLAM (Simultaneous Localization and Mapping) fournissent une solution qui permet de propager l'incertitude dans le temps. Cette prise en compte de l'incertitude à chaque instant va permettre d'appréhender les nouvelles informations avec une connaissance *a priori* "souple" qui va s'adapter en fonction des nouvelles mesures.

Nous verrons que des implémentations du SLAM existent et donnent des résultats très intéressants. Alors que la vaste majorité de ces implémentations appliquées à la vision monoculaire représentent l'environnement et le mesure à travers des points, cette thèse s'intéresse à la conception d'un système dérivé du SLAM qui utilise une primitive géométrique plus complexe : le plan. Dans notre environnement quotidien, en milieu urbain ou en intérieur, le plan est une primitive abondante.

Ce chapitre décrit comment l'intégration des plans dans un algorithme de SLAM est effectué et quels problèmes classiques du SLAM monoculaire cette intégration des structures planes peux résoudre. Afin d'introduire notre méthode, on s'attachera à décrire dans un premier temps le SLAM (générique et appliqué à la vision monoculaire), ses problèmatiques et les principes théoriques sous-jacents statistiques et mathématiques. L'étude des problématiques de traitement d'image, fournissant une mesure d'entrée et d'initialisation de nouveaux éléments de la carte, est en grande partie reportée à un chapitre ultérieur.

2.1 Fondamentaux de l'estimation statistique

Les méthodes de type SLAM sont des méthodes d'estimation statistiques. Afin de mieux appréhender la suite de ce chapitre, un court rappel des principes utilisés est nécessaire. Le filtrage statistique consiste à estimer, en fonction d'un flux de mesures provenant de capteurs, l'état d'un système dynamique. On souhaite connaitre l'état courant qui correspond le mieux aux mesures collectées et aux informations que l'on connait *a priori*, sachant que ces données peuvent être bruitées et que les mesures peuvent être une projection de l'état dans une dimension de taille inférieure.

Les informations du système sont considérées comme des variables aléatoires au sens statistique du terme. On considérera donc l'état et la mesure comme deux variables aléatoires pour lesquelles on devra attribuer ou estimer une fonction de densité de probabilité. Le travail de l'estimateur statistique va donc être –pour simplifier– d'estimer la fonction de densité de probabilité (nommée par la suite pdf pour Probability Density Function) de l'état qui corresponde le mieux à celle des informations *a priori* et de celle de la (ou des) mesure(s).

2.1.1 Filtrage bayésien

Dans le cadre théorique qui nous intéresse, les mesures que l'on obtient sont discrètes et doivent être traitées au fur et à mesure qu'elles arrivent (cadre temps réel). Le problème est donc pour chaque nouvelle mesure de mettre à jour les caractéristiques de l'état. L'historique des mesures précédentes n'est pas reconsidéré ni conservé par souci de performance, seule est connue la dernière estimation. Soit $\mathbf{x}$ l'état du système, $\mathbf{z_k}$ une mesure à l'instant k et $\mathbf{Z_k}$ l'ensemble des mesures accumulées à l'instant k. Si ces trois variables sont des variables aléatoires [1], on peut parler alors d'estimation bayésienne récursive :

$$p(\mathbf{x}|\mathbf{Z_k}) = \frac{\mathbf{p}(\mathbf{z_k}|\mathbf{x})\mathbf{p}(\mathbf{x}|\mathbf{Z_{k-1}})}{\mathbf{p}(\mathbf{z_k}|\mathbf{Z_{k-1}})} \tag{2.1}$$

Etant donné la vraisemblance de la mesure par rapport à l'état $p(\mathbf{z_k}|\mathbf{x})$, la vraisemblance globale de la mesure $p(\mathbf{z_k}|\mathbf{Z_{k-1}})$ et l'*a priori* $p(\mathbf{x}|\mathbf{Z_{k-1}})$ estimé à l'itération

1. Une variable aléatoire est une variable dont la valeur est incertaine mais modélisée sous forme de fonction de densité de probabilité.

précédente, le maximum *a posteriori* de l'itération courante est $p(\mathbf{x}|\mathbf{Z_k})$.

L'ensemble des méthodes présentées par la suite utilisent cette notion de filtrage bayésien comme fondement théorique. Il n'existe pas de méthode algébrique qui permette d'implémenter le filtrage bayésien dans tous les cas. Suivant les besoins de l'application visée, on utilisera diverses simplifications pour trouver une solution optimale. Deux de ces simplifications applicables au problème du SLAM sont présentées dans ce chapitre.

2.1.2 *A priori* et prédiction

L'information *a priori* est primordiale dans le cadre d'une estimation statistique. Elle permet de contraindre l'espace de solutions et de contourner le fait que les mesures à chaque itération ne permettent pas forcément de contraindre l'état complètement. L'information *a priori* permet donc d'effectuer la mise à jour connaissant l'ensemble des informations acquises à travers les anciennes mesures.

L'utilisation d'une phase de prédiction permet d'améliorer cette information à priori. En effet, il est souvent possible de fournir des informations sur les changements de l'état en fonction du temps, que ce soit par l'odométrie ou par un modèle (dans notre cas de déplacement). Dans le cas d'applications où les mesures sont discrètes et espacées dans le temps, la prédiction permet de "compenser" le manque de mesure dans l'intervalle et de prédire la mesure à venir.

2.1.3 Filtre de Kalman

Les différentes variables aléatoires présentées plus haut peuvent être de n'importe quelle forme. Une simplification possible est de trouver une loi de probabilité générique appliquable. Lorsqu'on ne dispose que de peu d'informations sur les caractéristiques d'un bruit de mesure, une simplification générique est d'utiliser une loi normale Gaussienne. La loi Gaussienne est reconnue pour être la meilleure représentation d'une variable aléatoire lorsqu'aucune information (à la conception du système) n'est connue sur la forme de cette variable.

Une loi normale Gaussienne $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ est définie par une moyenne μ et une dispersion autour de cette moyenne, la variance $\boldsymbol{\Sigma}$.

$$p(x) = \frac{1}{(2\pi)^{N/2}\,|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{2.2}$$

La principale restriction de la loi normale Gaussienne est qu'elle est unimodale et qu'elle ne peut donc pas représenter plusieurs solutions distinctes. En effet, elle est constituée d'un pic centré sur la moyenne. Si l'on considère que $p(\mathbf{z_k})$ et $p(\mathbf{z_k}|\mathbf{x})$ suivent toutes deux une loi normale, on peut utiliser le filtre de Kalman [BarShalom 93] comme estimateur.

On représente donc l'état du système par une variable aléatoire suivant une loi normale $\mathcal{N}(\mathbf{x}, \mathbf{P})$.

À l'initialisation du filtre, on détermine l'état initial du système $\mathbf{x}_{0|0}$ et la matrice de covariance $\mathbf{P}_{0|0}$ en fonction des informations dont on dispose.

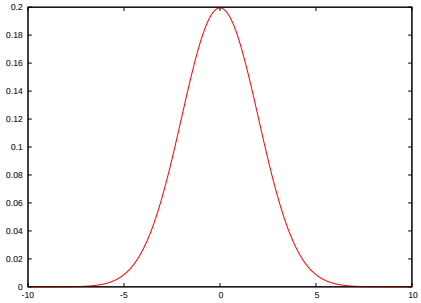


FIGURE 2.1 – Exemple de loi normale $\mathcal{N}(0, 2)$

On décrira ici le filtre de Kalman étendu (EKF), qui est sensiblement identique au filtre de Kalman classique si ce n'est qu'il permet d'utiliser des modèles d'évolution et de mesure non linéaires. Les phases décrites par la suite sont répétées de manière itérative pour chaque occurence d'une nouvelle mesure.

Prédiction de l'état. On considère que l'on dispose d'un modèle de déplacement f de notre système qui nous permet, connaissant l'état du système à l'itération précédente ($\mathbf{x}_{k-1|k-1}$ et un intervalle de temps δt de prédire en "aveugle" le nouvel état du système $\mathbf{x}_{k|k-1}$. Le bruit $\mathbf{v}_k$ de loi $\mathcal{N}(\mathbf{0}, \mathbf{Q})$ représente l'incertitude de la prédiction.

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1|k-1}, \mathbf{v}_k, \delta t) \tag{2.3}$$

On met à jour la matrice de covariance en utilisant

$$\mathbf{P}_{k|k-1} = \mathbf{F_x}\mathbf{P}_{k-1|k-1}\mathbf{F_x}^T + \mathbf{F_v}\mathbf{Q}\mathbf{F_v}^T \tag{2.4}$$

où $\mathbf{F_x}$ correspond à $\frac{\partial \mathbf{f}(\mathbf{x_{k-1|k-1}}, \mathbf{0}, \delta \mathbf{t})}{\partial \mathbf{x_{k-1|k-1}}}$ et $\mathbf{F_v}$ à $\frac{\partial \mathbf{f}(\mathbf{x_{k-1|k-1}}, \mathbf{0}, \delta \mathbf{t})}{\partial \mathbf{v_k}}$. D'une manière générale, on observera que les Jacobiennes sont toujours calculées en utilisant la meilleure estimation disponible de l'état.

Prédiction de la mesure. La mesure $\mathbf{z}$ (qui suit une loi $\mathcal{N}(\mathbf{z}_k, \mathbf{R})$) est fournie par un élément extérieur au système comme un capteur et doit pouvoir être estimée par le filtre afin d'être comparée pendant la mise à jour. Il est donc nécessaire de posséder un modèle numérique h du capteur qui permette, en fonction de l'état du système et d'éventuelles variables extérieures, de prédire la mesure. Cette mesure prédite est représentée par la variable Gaussienne $\mathcal{N}(\mathbf{z}_{k|k-1}, \mathbf{S})$ où :

$$\mathbf{z}_{k|k-1} = h(\mathbf{x}_{k|k-1}) \tag{2.5}$$

La covariance $\mathbf{S}$ est calculée à l'aide de la Jacobienne $\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x_{k|k-1}})}{\partial \mathbf{x_{k|k-1}}}$.

$$\mathbf{S} = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R} \tag{2.6}$$

Mise à jour de l'état. La phase de mise à jour correspond à l'estimation du maximum *a posteriori* (2.1). La corrélation introduite par la matrice de covariance de l'état va nous permettre de mettre à jour l'ensemble de l'état même si la mesure ne le définit pas complètement. Bien sûr, il sera nécessaire d'observer l'ensemble des dimensions de l'état au fur et à mesure des itérations si l'on souhaite l'estimer correctement. La matrice de gain $\mathbf{W}$ répartit l'information contenue dans la différence entre l'estimation et la mesure (l'innovation) entre les variables de l'état.

$$\begin{aligned} \mathbf{W} &= \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}^{-1} && (2.7) \\ \mathbf{x}_{k|k} &= \mathbf{x}_{k-1|k-1} + \mathbf{W}(\mathbf{z}_k - \mathbf{z}_{k|k-1}) && (2.8) \end{aligned}$$

La matrice de covariance est mise à jour suivant le même principe. Cette phase permet de réduire les incertitudes du système et est logiquement explicable par le fait que chaque mesure rajoute de l'information. Les équations de mise à jour de l'EKF sont dérivées de la minimisation de la trace de la matrice $\mathbf{P}_{k|k-1}$.

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{W}\mathbf{S}\mathbf{W}^T \tag{2.9}$$

La dernière étape ne fait pas partie des équations théoriques du filtre de Kalman mais s'avère nécessaire dans une implémentation informatique, à cause des problèmes de précision et d'arrondis. Elle consiste à s'assurer que la matrice de covariance est bien symétrique. On observe de temps en temps des problèmes de divergence si cette étape n'est pas appliquée.

$$\mathbf{P}_{k|k} = \frac{\mathbf{P}_{k|k} + \mathbf{P}_{k|k}^T}{2} \tag{2.10}$$

Outil statistique Une méthode importante dans l'utilisation d'un filtre de Kalman est le test du χ^2. Cette méthode permet de tester si une valeur donnée peut correspondre statistiquement à une variable aléatoire Gaussienne. C'est en quelque sorte une mesure de distance. Considérons une variable aléatoire $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ de dimension n et une valeur λ. Si l'inégalité

$$(\lambda - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\lambda - \boldsymbol{\mu})^T \leqslant \chi^2(n, p) \tag{2.11}$$

est vérifiée, cela signifie que l'on considère que la valeur λ peut être une observation de la variable aléatoire. La constante $\chi^2(n, p)$ correspond à la distance maximale qui devrait être obtenue dans p% des observations. On utilise cette méthode, par exemple, pour vérifier que la mesure obtenue n'est pas aberrante et peut-être utilisée.

Limitations. Par construction, le filtre de Kalman se limite à prendre en compte des pdf Gaussiennes, ce qui peut être évidemment inadapté dans le cas où la pdf réelle est radicalement différente (multimodale par exemple). Dans le cas de l'EKF, qui concerne les modèles non linéaires, il est nécessaire de linéariser au premier ordre ces modèles. La preuve de convergence du filtre de Kalman peut alors ne plus être valide si le modèle est très fortement non linéaire. Enfin, dernière limitation majeure,

la complexité calculatoire est quadratique par rapport à la dimension de l'état, ce qui peut être problématique dans le cadre du SLAM comme nous pourrons le voir.

2.1.4 Filtre particulaire

Une autre méthode de filtrage très utilisée, notamment dans le domaine du suivi est le filtre particulaire. Le filtre particulaire [Isard 98, Rekleitis 03] est fortement similaire, dans son principe, au filtre de Kalman puisqu'il s'agit aussi d'un filtre statistique bayésien qui fonctionne séquentiellement. Alors que l'EKF se "limite" à l'estimation des deux premiers moments statistiques (ce qui est une approximation si l'état n'a pas réellement une fonction de densité de probabilité Gaussienne), le filtre particulaire va permettre d'éliminer cette contrainte et d'estimer des pdf beaucoup plus complexes. Il permet entre autres d'estimer des fonctions de densité de probabilité multi-modales. Au lieu d'estimer les paramètres d'une fonction, ce qui peut être délicat dans le cas de fonctions complexes, le principe du filtre particulaire est de discrétiser celle-ci et d'estimer l'espace de recherche ponctuellement.

Comme toute méthode de recherche opérationnelle, le rôle du filtre particulaire est d'évaluer au moins localement l'espace de recherche afin de trouver le ou les points optimaux dans celui-ci. La dimension de l'espace correspond à celle de l'état estimé. Dans le cadre d'un filtrage statistique, l'espace de recherche est évalué par la densité de probabilité en chaque point. Une particule va donc représenter un point de cette espace de recherche et son poids (sa taille) la densité de probabilité associée à ce point. Le filtre particulaire va donc s'attacher à rechercher les meilleures solutions en plaçant et déplaçant judicieusement les particules afin d'obtenir en quelque sorte les pics de la pdf de l'état. On note $\mathbf{x}_i$ une particule et w_i le poids, représentant la densité de probabilité, qui lui est associé.

Prediction. Comme pour le filtre de Kalman, la prédiction consiste à transformer l'état, et donc ici l'état de chaque particule. Cette transformation est effectuée en utilisant un modèle de prédiction sélectionné par le concepteur du filtre. Cette prédiction va permettre de parcourir l'espace.

Mise à jour. Lorsqu'une mesure est disponible, celle-ci va servir à mettre à jour notre estimation de l'espace de recherche, et donc du poids des particules qui décrivent cet espace. La mise à jour du poids w_i d'une particule $\mathbf{x}_i$ est effectuée par

$$w_i \quad = \quad w_i p(\mathbf{x_i}|\mathbf{z}) \tag{2.12}$$

Le poids de chaque particule est ensuite normalisé par la somme des poids de l'ensemble des particules afin de conserver une cohérence numérique. L'utilisation des particules pour en déduire un résultat dépend de l'application. On pourra considérer tout simplement que la particule dont le poids est le plus fort est l'état du système à considérer. Une autre solution est de prendre la moyenne des particules pondérée par le

poids de chacune, en éliminant les particules dont le poids est inférieur à un certain seuil. L'utilisation de plusieurs particules comme autant de résultats différents est, bien évidemment, également possible.

Redistribution. La mise à jour tend à réduire le poids des particules. Afin de permettre une continuité de l'estimation, il est nécessaire d'éliminer les particules qui décrivent un point de l'espace peu probable. Celles-ci consomment des ressources calculatoires sans être d'un grand avantage descriptif. Elles sont donc remplacées par de nouvelles particules disposées à des endroits plus propices (par exemple en clonant les meilleures particules).

Limitations Les principaux avantages du filtre particulaire tiennent non seulement dans l'appréciation de pdf multimodales mais aussi dans la non linéarisation des modèles (qui peut être un facteur dépréciatif de la qualité dans l'EKF). L'inconvénient majeur du filtre particulaire tient dans le fait qu'il est nécessaire d'évaluer chaque particule pour connaître sa densité de probabilité. Cela implique des coûts calculatoires non négligeables dans le cas où la dimension de l'espace de recherche est élevée. Des solutions existent – comme l'utilisation des principes de réduction de l'espace type Rao-Blackwell – afin de réduire ce problème sans toutefois le supprimer totalement. La qualité de l'estimation est, par construction, corrélée à la finesse de la discrétisation de l'espace et donc au nombre de particules qui définissent l'espace à un instant donné t.

2.2 Modèle de prédiction et représentation de la caméra

Le modèle de prédiction utilisé dans le filtre de Kalman étendu (EKF) dépend du contexte applicatif et des connaissances que l'on a sur les mouvements du système : degrés de libertés, amplitudes des changements possibles, etc. Dans le cadre d'une application robotique par exemple, on connait les degrés de libertés autorisés par ses moteurs, les vitesses maximum qu'il peut atteindre, et surtout on peut connaitre les commandes qui sont données aux servo-moteurs afin d'avoir une prédiction relativement précise. Ces informations sont bien sûr insuffisantes pour une estimation à long terme, mais facilitent la prise en compte des informations extéroceptives lors de la mise à jour. Dans le cas qui nous intéresse, le système est une caméra (ou un bloc IMU et caméra) porté par un utilisateur. On n'a donc que très peu d'informations permettant de prédire le mouvement à l'avance puisque l'utilisateur va pouvoir bouger comme il le souhaite. On doit donc trouver un modèle de prédiction générique qui corresponde le plus à la réalité en n'ayant comme seule information que les mouvements précédents. On doit trouver un compromis entre un modèle trop simple qui impliquerait une grande incertitude et un modèle trop strict qui risquerait de ne jamais être valide.

Les modèles classiques de mouvement basés uniquement sur le passé sont définis dans la littérature [Zhang 92] :

– modèle de position constante

- modèle de vitesse constante en translation et rotation
- modèle d'accélération constante en translation, vitesse constante en rotation
- modèle de translation générale, vitesse constante en rotation

Le modèle le plus approprié dans le cadre de notre application semble être celui qui considère l'accélération en translation comme constante. Elle permet d'appréhender un plus grand nombre de mouvements sans alourdir significativement la complexité et de plus facilitera l'intégration de l'IMU comme décrit dans le chapitre 5.
Contrairement à la plupart des références dans la littérature qui utilisent ce modèle dans une implémentation du EKF-SLAM, les paramètres de pose et de mouvement du système sont définis dans le repère du système $\mathcal{F}_c$ et non dans un repère fixe $\mathcal{F}_o$.${}^c\mathbf{M}_o$ sera donc estimée en lieu et place de ${}^o\mathbf{M}_c$, ce qui d'après nos expériences permet d'obtenir une estimation moins bruitée.
On a donc une prédiction basée sur un modèle utilisant l'accélération et la vitesse de rotation. Puisque la prédiction se fait suivant l'estimation précédente de l'état, on doit intégrer ces termes dans l'état. On peut donc définir le vecteur d'état de la caméra comme

$$\mathbf{x_c} = \begin{bmatrix} \mathbf{t} & \mathbf{r} & \mathbf{v} & \boldsymbol{\omega} & \mathbf{a} \end{bmatrix}^{\mathsf{T}} \tag{2.13}$$

$\mathbf{t}$ est la position ${}^c\mathbf{t}_o$ et $\mathbf{r}$ l'orientation ${}^c\mathbf{r}_o$ représentée sous la forme angle/axe de rotation. $\mathbf{v}$, $\boldsymbol{\omega}$ représentent la vitesse de déplacement instantanée de la caméra. le terme $\mathbf{a}$ correspond quant à lui à l'accélération en translation.

Le modèle de prédiction [Bourquardez 04] est donc défini comme

$$f(\mathbf{x_c}, \delta t) = \begin{bmatrix} f_{\mathbf{t}}(\mathbf{x_c}, \delta t) \\ f_{\mathbf{r}}(\mathbf{x_c}, \delta t) \\ f_{\mathbf{v}}(\mathbf{x_c}, \delta t) \\ f_{\boldsymbol{\omega}}(\mathbf{x_c}, \delta t) \\ f_{\mathbf{a}}(\mathbf{x_c}, \delta t) \end{bmatrix} \tag{2.14}$$

Bruit. Puisqu'on considère un modèle avec des termes d'accélération et de vitesse de rotation constants dans le temps, le bruit associé au modèle (la différence entre l'état prédit et l'état réel) correspond à une erreur sur ces termes qui est propagé sur les autres paramètres de la caméra. On considère donc que l'accélération en translation et la vitesse de rotation sont toutes deux entachées d'un bruit additif. Ce bruit peut être modélisé comme un terme supplémentaire d'ordre supérieur (un "jerk" en translation et une accélération en rotation) :

$$\mathbf{a} = \mathbf{a} + \mathbf{j}\delta t \tag{2.15}$$

$$\boldsymbol{\omega} = \boldsymbol{\omega} + \boldsymbol{\omega}_{acc}\delta t \tag{2.16}$$

Cependant, pour simplifier les calculs dans l'intégration de ces bruits dans la prédiction de la position et de l'orientation, le bruit suivant est considéré :

$$\mathbf{a} = \mathbf{a} + \mathbf{n_a} \tag{2.17}$$
$$\boldsymbol{\omega} = \boldsymbol{\omega} + \mathbf{n_\omega} \tag{2.18}$$

Le bruit n'est donc plus considéré comme un terme d'ordre supérieur mais comme un simple biais.

Prédiction de l'accélération et de la vitesse de rotation. Ces deux termes sont considérés comme constants, ils ne sont donc pas affectés par la prédiction

$$f_{\boldsymbol{\omega}}(\mathbf{x_c}, \delta t) = \boldsymbol{\omega} \tag{2.19}$$
$$f_{\mathbf{a}}(\mathbf{x_c}, \delta t) = \mathbf{a} \tag{2.20}$$

Prédiction de la vitesse de déplacement La vitesse est prédite en intégrant directement l'accélération en fonction de l'intervalle de temps δt :

$$f_{\mathbf{v}}(\mathbf{x_c}, \delta t) = \mathbf{v} + \mathbf{a}\delta t \tag{2.21}$$

Prédiction de l'orientation :

$${}^{c2}\mathbf{R}_{c1} = R(\boldsymbol{\omega}\delta t) \tag{2.22}$$
$${}^{c2}\mathbf{R}_{o} = R(\mathbf{r}) \tag{2.23}$$
$$f_{\mathbf{r}}(\mathbf{x_c}, \delta t) = \phi({}^{c2}\mathbf{R}_{c1}^{T}\,{}^{c2}\mathbf{R}_{o}) \tag{2.24}$$

où les fonctions$R$ et ϕ sont respectivement définies par les équations 1.17 et 1.20.

Prédiction de la nouvelle position. Puisque la rotation estimée est ${}^{c2}\mathbf{R}_o$, le changement d'orientation rentre en compte dans le calcul de la nouvelle position tout comme les paramètres de déplacement en translation $\mathbf{v}$ et $\mathbf{a}$. On base notre modèle de prédiction sur les équations de cinématique pour une cible fixe et une caméra mobile.

$$\dot{\mathbf{t}}(\delta t) = -\mathbf{v} - \boldsymbol{\omega} \times \mathbf{t} - \mathbf{a}\delta t \tag{2.25}$$

L'intégration [Koller 97, Zhang 92] de l'équation (2.25) dans le temps permet de déterminer la prédiction de la nouvelle position après un intervalle de temps δt.

$$f_{\mathbf{t}}(\mathbf{x_c}, \delta t) = R(-\boldsymbol{\omega}\delta t)\mathbf{t} - S(-\boldsymbol{\omega}dt)\mathbf{v}\delta t - T(-\boldsymbol{\omega}dt)\mathbf{a}\frac{\delta t^2}{2} \tag{2.26}$$

où

$$
\begin{aligned}
R(\theta\mathbf{u}) &= \mathbf{I} + \frac{sin(\theta)}{\theta}[\theta\mathbf{u}]_\times + \frac{1-cos(\theta)}{\theta^2}[\theta\mathbf{u}]_\times^2 && (2.27)\\
S(\theta\mathbf{u}) &= \int_0^{\delta t} R(\theta\mathbf{u}\delta t)\mathbf{v} && (2.28)\\
&= \mathbf{I} + \frac{1-cos(\theta)}{\theta^2}[\mathbf{u}]_\times + \frac{\theta - sin(\theta)}{\theta^3}[\mathbf{u}]_\times^2 && (2.29)\\
T(\theta\mathbf{u}) &= \int_0^{\delta t} R(\theta\mathbf{u}\delta t)\mathbf{a}\delta t && (2.30)\\
&= \mathbf{I} + 2\frac{\theta - sin(\theta)}{\theta^3}[\mathbf{u}]_\times + \frac{\theta^2 - 2(1-cos(\theta))}{\theta^4}[\mathbf{u}]_\times^2 && (2.31)
\end{aligned}
$$

Jacobiennes. L'étape de prédiction nécessite (voir la section 2.1.3) de connaître les matrices Jacobiennes $\mathbf{F_x}$ et $\mathbf{F_v}$ afin de calculer la nouvelle variance de l'état. On cherche à connaitre la Jacobienne de l'ensemble des paramètres de la caméra en fonction de ces mêmes paramètres

$$
\frac{\partial \mathbf{f}(\mathbf{x_c}, \delta\mathbf{t})}{\partial \mathbf{x_c}} = \begin{bmatrix}
R(-\boldsymbol{\omega}\delta t) & \mathbf{0} & -S(-\boldsymbol{\omega}\delta t)\delta t & \frac{\partial \mathbf{f_t}(\mathbf{x_c},\delta\mathbf{t})}{\partial \boldsymbol{\omega}} & -T(-\boldsymbol{\omega}\delta t)\frac{\delta t^2}{2} \\
\mathbf{0} & \frac{\partial \mathbf{f_r}(\mathbf{x_c},\delta\mathbf{t})}{\partial \mathbf{r}} & \mathbf{0} & \frac{\partial \mathbf{f_r}(\mathbf{x_c},\delta\mathbf{t})}{\partial \boldsymbol{\omega}} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I}\delta t \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}
\end{bmatrix} \tag{2.32}
$$

mais également la Jacobienne de l'ensemble des paramètres de la caméra en fonction des bruits $\mathbf{n_a}$ et $\mathbf{n_\omega}$

$$
\frac{\partial \mathbf{f}(\mathbf{x_c}, \delta\mathbf{t})}{\partial \mathbf{n}} = \begin{bmatrix}
-T(-\boldsymbol{\omega}\delta t)\frac{\delta t^2}{2} & \frac{\partial \mathbf{f_t}(\mathbf{x_c},\delta\mathbf{t})}{\partial \boldsymbol{\omega}} \\
\mathbf{0} & \frac{\partial \mathbf{f_r}(\mathbf{x_c},\delta\mathbf{t})}{\partial \boldsymbol{\omega}} \\
\mathbf{I}\delta t & \mathbf{0} \\
\mathbf{0} & \mathbf{I} \\
\mathbf{I} & \mathbf{0}
\end{bmatrix} \tag{2.33}
$$

Dérivations. la matrice Jacobienne (2.32) est constituée en partie de sous matrices non encore définies :

$$
\begin{aligned}
\frac{\partial \mathbf{f_t}(\mathbf{x})}{\partial \boldsymbol{\omega}} &= -\mathbf{A} + \mathbf{B} + \mathbf{C} && (2.34)\\
\mathbf{A} &= (\mathbf{t}^\mathsf{T} \otimes \mathbf{I})\frac{\partial \mathbf{R}(-\boldsymbol{\omega}\delta\mathbf{t})}{\partial(-\boldsymbol{\omega}\delta\mathbf{t})}\delta\mathbf{t} && (2.35)\\
\mathbf{B} &= (\mathbf{v}^\mathsf{T}\delta t \otimes \mathbf{I})\frac{\partial \mathbf{S}(-\boldsymbol{\omega}\delta\mathbf{t})}{\partial(-\boldsymbol{\omega}\delta\mathbf{t})}\delta\mathbf{t} && (2.36)\\
\mathbf{C} &= (\mathbf{a}^\mathsf{T}\frac{\delta t^2}{2} \otimes \mathbf{I})\frac{\partial \mathbf{T}(-\boldsymbol{\omega}\delta\mathbf{t})}{\partial(-\boldsymbol{\omega}\delta\mathbf{t})}\delta\mathbf{t} && (2.37)\\
& && (2.38)
\end{aligned}
$$

$$\frac{\partial \mathbf{f_r}(\mathbf{x})}{\partial \mathbf{r}} = \frac{\partial \phi(\mathbf{R}(\boldsymbol{\omega}\delta\mathbf{t})^{\mathsf{T}}.\mathbf{R}(\mathbf{r}))}{\partial \mathbf{R}(\boldsymbol{\omega}\delta\mathbf{t})^{\mathsf{T}}.\mathbf{R}(\mathbf{r})}(\mathbf{I} \otimes \mathbf{R}(-\boldsymbol{\omega}\delta\mathbf{t}))\frac{\partial \mathbf{R}(\mathbf{r})}{\partial \mathbf{r}} \quad (2.39)$$

$$\frac{\partial \mathbf{f_r}(\mathbf{x})}{\partial \boldsymbol{\omega}} = -\frac{\partial \phi(\mathbf{R}(\boldsymbol{\omega}\delta\mathbf{t})^{\mathsf{T}}.\mathbf{R}(\mathbf{r}))}{\partial \mathbf{R}(\boldsymbol{\omega}\delta\mathbf{t})^{\mathsf{T}}.\mathbf{R}(\mathbf{r})}(\mathbf{R}(\mathbf{r})^{\mathsf{T}} \otimes \mathbf{I})\frac{\partial \mathbf{R}(-\boldsymbol{\omega}\delta\mathbf{t})}{\partial -\boldsymbol{\omega}\delta\mathbf{t}}\delta\mathbf{t} \quad (2.40)$$

2.2.1 Conclusion

Par la suite, le terme "modèle de prédiction du déplacement de la caméra" sera associé au "modèle d'accélération constante et de vitesse de rotation constante" décrit précédemment. On verra par la suite (chapitre 5) que le choix de ce modèle n'est pas sans conséquence sur les autres parties de cette thèse et permettra de faciliter l'intégration des capteurs inertiels. Ce modèle est légèrement plus complexe que ceux utilisés habituellement dans la littérature, mais il a pour avantage de définir correctement la corrélation entre le déplacement de la caméra et la vitesse de rotation. Dans la majorité des implémentations en effet, cette composante introduite par les équations de la dynamique des solides est ignorée et les 6 paramètres de la pose sont considérés comme indépendants.

2.3 Localisation et cartographie simultanées

« La problématique du Simultaneous Localization and Mapping (S.L.A.M.) est de savoir s'il est possible qu'un robot mobile soit placé à une position inconnue dans un environnement inconnu et que celui-ci construise de manière incrémentale une carte cohérente de cet environnement tout en déterminant sa position simultanément dans cette carte. Une solution au problème du SLAM est vu comme le Graal par la communauté de la robotique mobile puisque cela permettrait d'avoir un robot complètement autonome. »

Introduction de [DurrantWhyte 06]

Dans cette section, on introduira les concepts du SLAM nécessaires à la compréhension des travaux effectués. Pour une introduction plus complète au SLAM, le lecteur pourra se diriger vers deux tutoriaux [DurrantWhyte 06, Mei 07].
La traduction « Localisation et cartographie simultanée »du terme SLAM en français est assez explicite par elle-même. Le concept vient du monde robotique mais peut-être appliqué à tout système équipé de capteurs extéroceptifs qui souhaite se localiser dans un environnement inconnu. Basé sur des travaux précédents sur les relations statistiques entre les éléments mesurés par un même système [Smith 91], le concept de SLAM est apparu dans le milieu des années 90 [H.DurrantWhyte 96, Csorba 97, Thrun 98]. Le terme de SLAM correspond à une problématique générique qui consiste à estimer à la fois un ensemble de paramètres lié à l'environnement et un ensemble de paramètres lié au système à localiser (en temps réel et au fur et à mesure que l'information arrive). Il n'est pas associé à une quelconque théorie ou méthode mais cependant la littérature

montre une large domination des méthodes probabilistes. En prenant en compte les diverses incertitudes introduites par les capteurs, ces méthodes probabilistes fournissent des solutions plus robustes dans le temps.

2.3.1 Principes probabilistes

Soit $\mathbf{p}_{0:k}$ l'ensemble des localisations 0 à k du système, $\mathbf{z}_{0:k}$ l'ensemble des mesures effectuées sur les données de l'environnement, $\mathbf{f}_{0:n}$ l'ensemble des n données de l'environnement. la problématique classique de la localisation est de déterminer

$$P(p_k|\mathbf{z}_{0:k}, \mathbf{f}_{0:n}, \mathbf{p}_0) \tag{2.41}$$

et la problématique de la cartographie est de déterminer la meilleure solution pour

$$P(f_{0:n}|\mathbf{z}_{0:k}, \mathbf{p}_{0:k}) \tag{2.42}$$

Le principe du SLAM est donc de réunir sous une seule et même estimation la localisation et le mapping, ce qui revient à chercher

$$P(p_k, f_{0:n}|\mathbf{z}_{0:k}, \mathbf{p}_0) \tag{2.43}$$

Comme cette distribution doit être évaluée à chaque nouvelle information, les méthodes utilisées dans la littérature sont récursives et basées sur le théorème de Bayes. Diverses solutions génériques au SLAM ont été proposées en utilisant des estimateurs statistiques.

2.3.1.1 EKF-SLAM

La plus étudiée est basée sur un filtre de Kalman [Csorba 97] dont le vecteur d'état contient à la fois les informations sur le système mais également sur la carte (les paramètres des éléments de la carte). La composante de localisation et la composante de mapping sont donc estimées dans le même vecteur d'état. Comme dans tout EKF, cet état est représenté par une variable aléatoire Gaussienne, et donc par un vecteur moyenne $\mathbf{x}$ et une matrice de covariance associée $\mathbf{P}$. On notera P_{x1x2} la covariance entre la variable $x1$ et la variable $x2$. Soit $\mathbf{p} = \mathbf{x_c}$ l'état du système et $\mathbf{f_i}$ un élément de la carte :

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} & \mathbf{f}_1 & \cdots & \mathbf{f}_n \end{bmatrix}^T \tag{2.44}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P_{pp}} & \mathbf{P_{pf_1}} & \cdots & \mathbf{P_{pf_n}} \\ \mathbf{P_{f_1p}} & \mathbf{P_{f_1f_1}} & \cdots & \mathbf{P_{f_1f_n}} \\ \vdots & \vdots & \ddots & \\ \mathbf{P_{f_np}} & \mathbf{P_{f_nf_1}} & \cdots & \mathbf{P_{f_nf_n}} \end{bmatrix} \tag{2.45}$$

La matrice de covariance représente à la fois les incertitudes sur chaque élément et les relations entre eux. Chaque terme en dehors de la diagonale peut en effet être perçu

comme la corrélation entre différents termes. Chaque itération va permettre d'augmenter les différentes corrélations tout en diminuant les termes d'incertitudes. Cette notion de corrélation va servir à propager les informations dans l'espace : si un élément de la carte est mesuré, cela va fournir au système de nouvelles informations sur son état, et comme il est relié à d'autres éléments par corrélation, cela fournit également de nouvelles informations sur l'ensemble des éléments du vecteur d'état. L'objectif étant de tendre vers un système dont les corrélations sont maximum, et donc où les paramètres relatifs sont quasi-certains.
Cette méthode a pour avantage d'être facilement adaptable et implémentable avec une grande efficacité si la carte n'est pas trop dense. Dans le cas de modèles de prédiction et de mesure linéaires, la convergence a été prouvée par [Newman 00]. De plus, cette méthode est richement étudiée. Enfin, elle facilite la fusion de multiples sources de données. Elle a par contre de multiples inconvénients (qui correspondent à ceux de l'EKF, comme la nécessité de linéariser les modèles, voir la section 2.3.3). Une difficulté supplémentaire propre au SLAM tient dans le fait qu'il est nécessaire, quand on observe à nouveau un élément de la carte, de l'associer avec l'élément précédemment enregistré. Comme l'EKF considère des pdf Gaussiennes, donc unimodales, il n'est capable de prendre en compte qu'une seule possibilité. Cela signifie qu'il est moins tolérant aux mesures aberrantes et donc aux mauvaises associations.
Considérant les contraintes matérielles, nos besoins de fusion, cette méthode a été retenue comme base de travail. On détaillera donc par la suite plus particulièrement les tenants et aboutissants spécifiques à l'EKF-SLAM.

2.3.1.2 FAST-SLAM

L'autre solution majeure au problème du SLAM est connue sous le nom de FAST-SLAM [Montemerlo 03a] (ou FAST-SLAM 2.0 [Montemerlo 03b]) qui utilise un filtre particulaire spécial. Filtre spécial puisque l'on ne peut pas utiliser une méthode classique de filtre particulaire et discrétiser un espace de recherche de dimension aussi importante (contenant les paramètres de chaque élément de la carte). L'idée de cette méthode est donc de réduire l'espace estimé par le filtre particulaire en utilisant la méthode de Rao-Blackwell. Au lieu de représenter l'espace de solutions entier avec des particules, ces dernières ne sont utilisées que pour décrire le chemin parcouru par le système (on s'intéresse alors non plus à estimer uniquement la dernière pose p_k mais l'ensemble du chemin $p_{0:k}$). Chaque particule contient non seulement un chemin estimé, mais également une carte des N éléments qui la constituent, chaque élément étant estimé à son tour par un filtre de Kalman très simple. Cette simplification est possible en observant que si on connait le chemin du robot, chaque élément de la carte peut être estimé de manière indépendante :

$$P(p_{0:k}, f_{0:n}|\mathbf{z}_{0:k}, \mathbf{p}_0) = P(p_k|\mathbf{z}_{0:k}, \mathbf{p}_0) \prod_i P(f_i|p_k, \mathbf{z}_{0:k}, \mathbf{p}_0) \tag{2.46}$$

L'algorithme crée donc pour chaque mode de l'estimation du chemin du système une carte locale indépendante. Des expériences ont démontré le très bon fonctionnement de

cette méthode et la convergence en pratique. Contrairement à la méthode basée EKF, chaque particule effectue sa propre association entre les différentes mesures d'un même élément de la carte, ce qui rend plus robuste l'algorithme. Cependant, comme expliqué dans le paragraphe sur la méthode précédente, la méthode basée EKF est plus adéquate dans notre cadre applicatif, notamment pour des raisons de facilité d'implémentation, d'optimisation et de fusion.

2.3.2 Précisions sur la méthode EKF-SLAM

Le choix de l'EKF-SLAM étant fait, cette section va détailler les différentes étapes de fonctionnement de cette méthode.

Initialisation. L'état initial doit être initialisé par une méthode externe. Cet état initial représente nos connaissances *a priori* sur le système et son environnement ainsi que la confiance dont on dispose sur chacune de ces informations. L'environnement du système à localiser étant inconnu, la carte de l'état initial est vide. Ce dernier ne contient donc que les informations sur le système.

$$\mathbf{x}_{0|0} = \left[\begin{array}{c} \mathbf{x}_c \end{array} \right] \tag{2.47}$$

$$\mathbf{P}_{0|0} = \left[\begin{array}{c} \mathbf{P}_{pp} \end{array} \right] \tag{2.48}$$

Si la pose du système est estimée par rapport à la pose initiale $\mathbf{x}_{0|0}$, l'initialisation est triviale puisque l'incertitude est nulle. Dans le cas où la pose est estimée par rapport à un repère global fixe, il faut déterminer à l'initialisation la pose relative initiale et l'incertitude de cette dernière.

Prédiction. La carte de l'environnement exprimée dans un repère fixe est considérée comme statique (les propriétés des différents éléments de la carte ne changent pas dans le temps). La prédiction, qui estime les changements *a priori* pour un intervalle de temps donné, n'affecte donc que le système et non la carte (la matrice de covariance est bien sûr affectée entièrement à cause des corrélations). L'étape de prédiction peut donc être simplifiée et donc le nombre d'éléments à calculer réduit. Soit $\mathbf{m}$ l'ensemble des éléments de la carte, les autres variables sont définies dans la section 2.2.

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1|k-1}, \delta t) \tag{2.49}$$

$$= \left[\begin{array}{c} f(\mathbf{x_c}_{k-1|k-1}, \delta t) \\ \mathbf{m} \end{array} \right] \tag{2.50}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F_x} \mathbf{P}_{k-1|k-1} \mathbf{F_x}^T + \mathbf{F_v} \mathbf{Q} \mathbf{F_v}^T \tag{2.51}$$

$$= \left[\begin{array}{cc} \mathbf{F}_{x_p} \mathbf{P_{pp}} \mathbf{F}_{x_p}^T + \mathbf{F_{v_p}} \mathbf{Q} \mathbf{F_{v_p}}^T & \mathbf{F}_{x_p} \mathbf{P_{pm}} \\ \mathbf{P_{mp}} \mathbf{F}_{x_p}^T & \mathbf{P_{mm}} \end{array} \right] \tag{2.52}$$

Augmentation. La carte est créée au fur et à mesure que de nouvelles mesures arrivent. Si l'algorithme estime qu'une mesure est pertinente et qu'il ne peut l'associer à aucun élément de la carte courante, cette information est alors ajoutée à la carte.

Il est à noter que les éléments de la carte et les informations sur le système doivent permettre d'estimer la mesure, fournie par un capteur, d'un élément donné. A condition que les informations soient suffisantes pour permettre cette estimation, la carte peut être représentée de multiples façons. Par conséquent, il est nécessaire que les informations mesurées soient suffisamment complètes pour que l'on puisse ajouter un élément à la carte avec une incertitude Gaussienne.

Dans le cas des capteurs du type "Profondeur et attitude (Range and Bearing)" (comme les lasers, sonars et autres systèmes de vision stéréo) il est possible d'inverser directement le modèle de mesure et d'obtenir directement les caractéristiques approximatives de l'élément mesuré. Pour d'autres types de capteurs (comme la vision monoculaire), il est impossible d'inverser le modèle de mesure et on doit utiliser des méthodes plus complexes, pouvant nécessiter plusieurs itérations avant l'ajout de l'information à la carte.

Lorsque l'élément peut être ajouté, l'état est augmenté en utilisant le "modèle d'augmentation" g qui dépend du type de capteur et du type d'élément cartographique que l'on choisit de représenter. Cette augmentation va transformer le plus souvent l'élément mesuré afin qu'il utilise le même repère que le reste de la carte. Cette augmentation modifie naturellement la matrice de covariance associée. On définit là encore l'information à ajouter comme une variable aléatoire Gaussienne $\mathcal{N}(\mathbf{l}, \mathbf{D})$.

$$\begin{aligned}
\mathbf{x} &= g(\mathbf{x}, \mathbf{l}) && (2.53)\\
&= \begin{bmatrix} \mathbf{p} \\ \mathbf{m} \\ g_l(\mathbf{x}, \mathbf{l}) \end{bmatrix} && (2.54)\\
\mathbf{P} &= \mathbf{G} \begin{bmatrix} \mathbf{P_{pp}} & \mathbf{P_{pm}} & 0 \\ \mathbf{P_{mp}} & \mathbf{P_{mm}} & 0 \\ 0 & 0 & \mathbf{D} \end{bmatrix} \mathbf{G}^T && (2.55)\\
&= \begin{bmatrix} \mathbf{P_{pp}} & \mathbf{P_{pm}} & \mathbf{P_{pp}}\mathbf{G}_p^T \\ \mathbf{P_{mp}} & \mathbf{P_{mm}} & \mathbf{P_{mp}}\mathbf{G}_p^T \\ \mathbf{G}_p\mathbf{P_{pp}} & \mathbf{G}_p\mathbf{P_{pm}} & \mathbf{G}_p\mathbf{P_{pp}}\mathbf{G}_p^T + \mathbf{G}_l\mathbf{D}\mathbf{G}_l^T \end{bmatrix} && (2.56)
\end{aligned}$$

où $\mathbf{G}$ représente $\frac{\partial \mathbf{g}(\mathbf{x},\mathbf{l})}{\partial \mathbf{x}}$, $\mathbf{G}_p$ représente $\frac{\partial \mathbf{g_l}(\mathbf{x},\mathbf{l})}{\partial \mathbf{p}}$ et enfin $\mathbf{G}_l$ représente $\frac{\partial \mathbf{g_l}(\mathbf{x},\mathbf{l})}{\partial \mathbf{l}}$. On peut observer que cette mise à jour crée une corrélation entre le nouvel élément et le reste de l'état, ce qui parait logique puisque la nouvelle information dépend de l'état du système qui lui-même a été antérieurement corrélé à la carte. L'essence du SLAM réside dans cette méthode d'ajout d'information : l'incertitude sur la mesure et sur l'ensemble de l'état vont être utilisées pour construire celle du nouvel élément de la carte. À travers la création des corrélations, les différentes incertitudes vont être propagées dans

l'espace (les éléments de la carte) et dans le temps (prises en compte dans les itérations ultérieures) et permettre ainsi une estimation durable de la pose et de la carte.

Mise à jour. La mise à jour de l'état s'effectue à l'aide des équations classiques de l'EKF. La Jacobienne **H** est évidemment définie uniquement pour les paramètres de l'état qui correspondent à la mesure, en l'occurence les éléments de la carte mesurés et les paramètres de localisation du système. L'innovation – la différence entre la mesure et son estimation – va donc permettre de corriger la localisation en même temps que la carte de l'environnement (à travers les corrélations) en prenant en compte l'incertitude des mesures. On notera que la phase de prédiction de la mesure fournit une matrice de covariance de l'innovation qui pourra être utilisée pour vérifier statistiquement que la mesure n'est pas aberrante.

Suppression. Lorsqu'un élément de la carte n'est plus utilisé ou devient obsolète (par une stratégie dépendante de l'application), il est nécessaire de supprimer cet élément de l'état afin de ne pas alourdir inutilement le calcul. Cette suppression se fait simplement par la suppression des lignes correspondantes dans le vecteur d'état et des lignes et colonnes correspondantes dans la matrice de covariance.

2.3.3 Problémes rencontrés lors de l'utilisation de l'EKF-SLAM

L'EKF-SLAM possède des avantages non négligeables qui rendent intéressante son utilisation dans notre contexte applicatif. Il n'est cependant pas exempt de défauts qu'il est nécessaire de connaitre pour éviter les écueils lors de la conception de l'estimateur présenté dans ce document.

Complexité. La complexité calculatoire de l'EKF-SLAM est un problème majeur dans l'implémentation du SLAM. Induite principalement par l'étape de mise à jour de l'état, la complexité est d'ordre $\mathcal{O}(N^2)$ où N est le nombre d'éléments de la carte. N est dépendant de la taille de l'environnement, du type d'élément et du temps passé à naviguer. Il est bien sûr possible d'utiliser différentes méthodes d'optimisation logicielle pour réduire les temps de calculs sans toutefois modifier cette complexité calculatoire. Plusieurs articles décrivent des solutions qui permettent de modifier structurellement l'algorithme afin de réduire les coûts dans les cas où N est grand.
Une première approche heuristique consiste à ne plus considérer en permanence la carte complète (et donc la matrice de covariance entière). Toutes les méthodes utilisant ce genre d'approche partitionnent en effet la carte globale en carte(s) locale(s). La manière dont les informations locales sont utilisées pour retrouver une information globale constitue la différence majeure entre les différentes solutions de la littérature :

- la première catégorie [Estrada 05] considère un ensemble de cartes locales avec leurs propres repères reliées de manière hiérarchique à un repère global ;
- la seconde catégorie [Chong 99, Williams 01] considère un réseau de cartes locales définies les unes par rapport aux autres sans repère global fédérateur ;

– la dernière catégorie [Guivant 01, Knight 01, Williams 01] utilise des états locaux et un état global. À chaque itération, l'état local correspondant à l'élément de la carte mesurée est mis à jour. De temps en temps (moins régulièrement), l'état global est mis à jour en utilisant les états locaux.

Une autre approche est d'essayer de rendre creuse la matrice de covariance afin de réduire le nombre d'opérations à effectuer à chaque itération. Cette méthode utilise non plus le filtre de Kalman sous sa forme habituelle mais le filtre d'information [F. 06, Eustice 05] afin d'obtenir une matrice dont les termes en dehors des diagonales sont proches de 0 et peuvent donc ne pas être considérés.

Problémes structurels. L'EKF-SLAM introduit des erreurs d'estimation liées à la linéarisation des modèles. Ces problèmes, liés au comportement intrinsèque du filtre de Kalman étendu, sont étudiés dans [Julier 01, Bailey 06]. On y découvre notamment que l'EKF-SLAM a une tendance optimiste : il a tendance à réduire l'incertitude trop rapidement (Ce qui peut être résolu artificiellement en ajoutant un bruit plus important que la réalité à chaque prédiction). [Julier 01] montre que l'estimation de l'EKF-SLAM a également tendance à devenir incohérente sur le long terme si le système ne bouge pas ou lorsqu'on utilise des variances trop larges. Cet article démontre cependant qu'en choisissant correctement les paramètres et les modèles utilisés, ces problèmes disparaissent. L'auteur souligne également l'inutilité des méthodes dérivées de l'EKF (comme l'UKF ou l'IEKF) pour améliorer l'estimation.

Problème de visibilité. La construction de la carte et la mise à jour de la pose se font à travers les corrélations entre les différents éléments de l'état. L'innovation apportée par la mesure va permettre de corriger la disposition relative des éléments de la carte. Elle ne permet cependant pas de corriger la dérive globale du système liée à la permissivité de l'incertitude.

La carte est en effet construite en fonction des mesures et de la pose du système, et cette dernière en fonction des mesures et de la carte. L'ajout d'éléments est donc une fonction de la mesure et de l'état actuel du système. L'erreur accumulée, même si elle est prise en compte statistiquement, va être propagée aux nouveaux éléments de la carte. Une dérive naturelle de la carte est alors observée et ne peut être corrigée qu'en ré-observant de nouveau les anciens éléments de la carte.

Cette dérive implique une nécessité de pouvoir ré-observer d'anciens éléments de la carte. Mais la ré-observation nécessite à son tour de pouvoir retrouver ces anciens éléments dans les nouvelles mesures. On parle d'association des mesures. Cette association est d'autant plus difficile que l'intervalle de temps est grand depuis la dernière mesure. En temps normal, on peut estimer géométriquement s'il est raisonnable de penser qu'on mesure un élément existant de la carte à partir de l'estimation de la pose. Dans le cas où l'intervalle de temps est grand, la dérive est trop importante pour utiliser ce type d'approche naïve. La littérature s'intéresse abondamment à ce problème du "loop-closing" [Surmann 04, Newman 06].

D'une manière générale, ce problème de visibilité décrit la tendance du SLAM à être

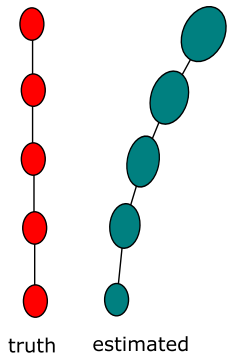


FIGURE 2.2 – Illustration de la dérive de l'estimation de la pose

plus correct (à court terme) dans l'estimation des informations relatives (sous-cartes proches et déplacement de la caméra) que dans l'estimation globale de la pose.

2.3.4 Conclusion

Le SLAM est un concept qui commence à devenir mature avec des méthodes dont le potentiel a pu être prouvé dans de nombreux types d'environnements et d'applications. On citera bien sûr des environnements intérieurs [Tardos 02], mais aussi en extérieur [Newman 06] et même en environnement sous-marin [Eustice 06]. Il reste évidemment des efforts de recherche à effectuer pour rendre le SLAM plus à même de fonctionner dans des environnements de très grande taille et réduire certains problèmes de stabilité dans le temps. Le SLAM a été originellement conçu pour fonctionner avec des capteurs robotiques classiques comme des LIDAR, des Lasers ou autres sonars qui fournissent des informations en trois dimensions sur l'environnement. Plus intéressant dans le contexte de cette thèse, le SLAM visuel utilisant un système stéréoscopique [Se 02, Davison 02, Jung 04] a permis de démontrer la faisabilité d'un système SLAM utilisant des caméras. La troisième dimension est alors obtenue par estimation et non plus par mesure directe et nécessite donc des méthodes (et donc des calculs supplémentaires) qui permettent de mettre en correspondance les informations visuelles des deux caméras. Le passage à une caméra, le SLAM monoculaire, fait l'objet de contraintes particulières qui ont été seulement récemment abordées dans la littérature.

2.4 Le SLAM Monoculaire

Le système de mesure dans nos travaux est une caméra. Le SLAM classique étant adapté à l'utilisation de capteurs différents, cette section va permettre de montrer les spécificités du SLAM monoculaire. De nouvelles contraintes, qu'il est nécessaire d'appréhender, sont en effet induites par l'utilisation d'une caméra.

2.4.1 La carte

Comme démontré dans la section 2.3.2, la carte est utilisée comme support à la localisation et doit permettre avec l'aide de la pose courante d'estimer, a priori, une mesure fournie par le capteur. Le choix des éléments constitutifs de la carte est une problématique majeure dans la définition d'un système SLAM puisqu'il déterminera le modèle de mesure, la densité nécessaire de la carte, et bien sûr les algorithmes de traitement d'images permettant d'extraire les mesures voulues. Il faut trouver un équilibre entre la complexité calculatoire (introduite par exemple par une carte trop dense), la richesse de l'information fournie par chaque élément et bien sûr la difficulté à extraire l'information des images. Pour permettre une localisation à long terme, il est également nécessaire que l'élément puisse être "reconnaissable" s'il est remesuré longtemps après avoir été perdu. La plupart du temps, on retrouvera dans la littérature des primitives géométriques simples comme des points [Davison 03] ou des segments de droite [Smith 06]. Les segments de droite permettent de fournir plus d'informations à chaque mesure puisqu'elles contraignent deux dimensions supplémentaires mais les points posent beaucoup moins de problèmes pour être extraits de l'image.

2.4.2 Initialisation des éléments de la carte

La mesure fournie par une caméra est la projection d'un élément de la scène 3D sur le plan image. À cause des imperfections des éléments constituant la caméra (comme la lentille ou le capteur CCD) mais également à cause des algorithmes d'extraction et de mesures, elles peuvent être bruitées. Cela signifie que la position qu'elle devrait avoir dans l'image en considérant un modèle de caméra pin-hole n'est pas exactement celle que l'on mesure. Cette erreur peut être représentée par un bruit additif Gaussien défini dans l'espace image.

Par souci de simplification, on considèrera par la suite qu'on mesure un point 2D, les problématiques décrites s'appliquant aux autres primitives image. Dans ce cas, la variable aléatoire est bien sûr à deux dimensions (les coordonnées image du point). Elle peut être considérée comme une zone ellipsoïdale dans l'image qui pourrait statistiquement contenir ce point (pour un pourcentage de confiance donné).

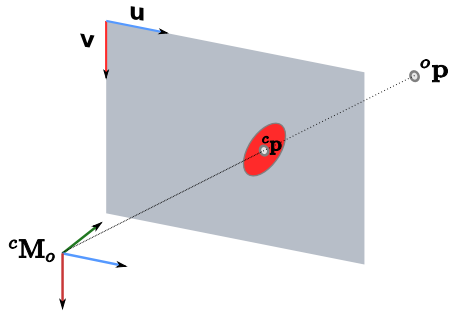


FIGURE 2.3 – Projection d'un point et incertitude de la mesure à une variance donnée

La reprojection est une caractéristique intrinsèque et physique du fonctionnement d'une caméra. Elle ne fournit aucune information sur la profondeur des points mesurés dans l'image. On sait simplement que le point se situe devant la caméra et on peut éventuellement supposer une profondeur maximum si l'on dispose d'*a priori* sur la scène. La probabilité que le point soit à une profondeur donnée est *a priori* uniforme sur tout l'intervalle. La position 3D correspondant à un point dans l'image que l'on peut reconstruire avec une seule image suit donc une loi normale sur deux dimensions et une loi uniforme sur la dimension de profondeur. Cette variable aléatoire peut être représentée intuitivement par un cône partant du repère de la caméra en direction du point 3D.

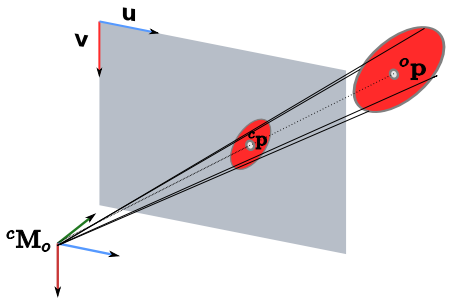


FIGURE 2.4 – Cône d'incertitude de la position 3D d'un point image

Comme cela a été expliqué précédemment, le SLAM ajoute dans son état les éléments mesurés éventuellement transformés pour compléter la carte. Par construction, les variables enregistrées dans l'état doivent être Gaussiennes. Il est évidemment difficile d'approximer une variable aléatoire uniforme par une Gaussienne. La problématique majeure propre au SLAM monoculaire est donc de trouver une solution à l'initialisation de nouvelles données dans la carte et fait l'objet d'un certain nombre de propositions dans la littérature que nous étudierons séparément ainsi que notre solution dans le chapitre 4. On considèrera par la suite que les informations sont connues complètement *a priori* par souci de simplification.

2.4.3 Problème du facteur d'échelle

La problématique du facteur d'échelle a été expliquée de manière générale dans le chapitre 1. Cette problématique s'applique bien évidemment au SLAM également à moins de disposer d'informations *a priori* sur les objets mesurés et/ou de disposer de mesures autres que celles basées vision. Dans le SLAM monoculaire, cette inconnue se répercutera inévitablement sur la carte en même temps que sur la localisation de la carte (garantissant par la même occasion la cohérence globale).

2.4.4 Conclusion

Cette application du SLAM aux capteurs caméra a prouvé dans la littérature son bon fonctionnement et permet d'effectuer avec une simple caméra une localisation pendant de longues sessions d'utilisation. Permettant de prendre en compte les incertitudes des mesures, ce concept de SLAM monoculaire va être étendu dans cette thèse à l'utilisation des plans. Une grande partie des problématiques traitées ultérieurement dans ce document permettront d'expliquer comment les problématiques classiques du SLAM en général et du SLAM monoculaire en particulier peuvent être résolues dans le cas spécifique où l'on considère des régions planes.

De nombreuses études théoriques et pratiques ont été effectuées sur le SLAM monoculaire. En réutilisant au maximum les solutions éprouvées pour les étendre quand cela est nécessaire, notre application peut profiter des preuves de bon fonctionnement et des heuristiques d'optimisation existant dans la littérature. Cette thèse s'appliquera donc à conserver la compatibilité de l'application décrite avec les méthodes traditionnelles, permettant par la même occasion d'utiliser simultanément les points et les plans quand le besoin s'en fera sentir.

2.5 Utilisation des régions planes

Contexte d'utilisation. Notre environnement quotidien est de plus en plus modelé par l'homme. Alors que la nature à tendance à créer des formes complexes et variées, l'homme a tendance à construire des formes simples. Par facilité de conception, on retrouve autour de nous une vaste majorité d'éléments constitués de surfaces planes ou qui peuvent être assimilées comme telles à une certaine distance. En extérieur on trouve les murs, les façades des bâtiments, les toits, les routes et trottoirs. En intérieur, on retrouve les murs, les décorations murales, le mobilier. Dans les magasins, les présentoirs peuvent souvent être assimilés à des plans. Même si les surfaces ne sont pas parfaitement planes, l'imperfection du capteur et les distances de mesure permettent souvent de réaliser cette approximation planaire. On peut donc imaginer que l'utilisation de surfaces planaires est intéressante dans de nombreux environnements.

Sources d'informations. De nombreuses sources fournissent des informations sur les surfaces planaires. En intérieur, il est possible de mesurer facilement les dimensions des pièces et les emplacements des fournitures à l'aide de mètres laser ou autres moyens de mesure. Plus simplement, on trouve dans les plans d'architectes (données CAO) les empreintes au sol des murs (segments de droites) et les dimensions des pièces et couloirs. L'information est alors sous forme $2D^{\frac{1}{2}}$. Il n'y a pas d'information sur l'inclinaison ou la courbure des murs, ce qui n'est pas forcément très contraignant dans la majorité des situations. En extérieur, il existe des informations sur les emplacements des bâtiments, des routes et autres murs dans les bases de données des systèmes d'informations géographiques (SIG) qui peuvent être acquises auprès de fournisseurs comme l'IGN. Elles sont représentées de la même manière que les plans d'architectes.

Ces informations fournies sont évidemment entachées d'erreurs. Problèmes de capteurs, interpolations erronées, résolution des documents et simplification du modèle sont autant de sources qui provoquent des biais dans les informations obtenues. Il est donc nécessaire de prendre en compte cette erreur si l'on souhaite utiliser efficacement les plans.

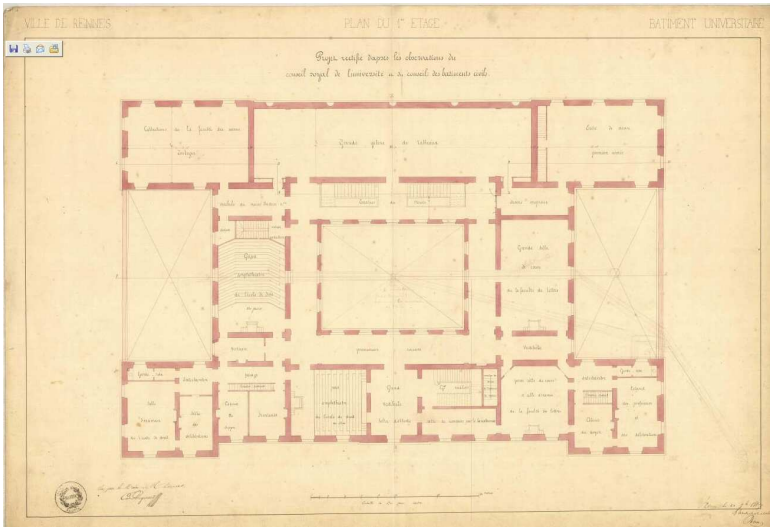


FIGURE 2.5 – Un plan d'architecte du musée de Rennes datant du XIXème siècle. Cette carte est ancienne, très peu précise et donc incertaine.

Intéret et informations utiles : les surfaces planes ont plusieurs avantages non négligeables en vision par ordinateur. On considère un ensemble de points appartenant à une surface plane. Le principal avantage est qu'il n'existe pas de discontinuité de profondeur dans les différents points. On sait donc que quel que soit le déplacement de la caméra, une partie de la surface ne pourra pas recouvrir une autre. Cela permet de faciliter entre autres le suivi basé image de zones dans l'image. Une autre propriété est que l'ensemble des points aura un déplacement dans l'image cohérent : le mouvement des points sera corrélé à celui des autres points dans l'image. En effet, le déplacement de deux points entre un repère $c1$ et un repère $c2$ appartenant au même plan est déterminé par la même matrice d'homographie (voir chapitre 1). Ceci permet de factoriser les données à estimer et de faciliter le suivi basé image en imposant des contraintes.
Les sources d'informations permettent de reconstruire un polygone à 4 cotés défini dans un repère global donné (Les bases de données étant constituées d'empreintes au sol sous la forme de segment 2D plus une hauteur associée). On peut extraire de ce polygone le plan qui le contient et les informations sur les bords du polygone. Seul le plan est utilisé dans les estimations puisqu'il contient toutes les informations qui contraignent le déplacement dans l'image des points. Les informations sur les bords et autres dimensions seront utilisées comme positions pour valider la cohérence des mesures et pour trouver

des zones dans l'image qui pourraient appartenir à ce polygone.
Le plan sera représenté sous la forme d'un vecteur **pl**

$$\mathbf{n} = \begin{bmatrix} a & b & c \end{bmatrix}^T \tag{2.57}$$

$$\mathbf{pl} = \begin{bmatrix} \mathbf{n} & d \end{bmatrix}^T \tag{2.58}$$

où $\mathbf{n}$ est le vecteur normal au plan et d est la distance orthogonal du plan à l'origine. Si le point 3D $\mathbf{X}$ (défini dans le même repère) appartient au plan, alors l'équation du plan est donnée par :

$$\mathbf{n}^{\mathrm{T}}\mathbf{X} - d = 0 \tag{2.59}$$

2.6 Les plans et l'estimation de pose dans la littérature

Par leur abondance dans notre environnement quotidien et les simplifications qu'ils permettent en vision par ordinateur, les plans sont une source d'inspiration majeure dans le domaine de la vision et de la géométrie multi-vues. L'homographie (voir chapitre 1) qui résulte du déplacement entre deux images d'une région plane est largement utilisée dans cette littérature. Calculée par le déplacement de points [Hartley 01] ou estimée directement à partir de l'information de luminance [Benhimane 07], de cette matrice peuvent être extraits des informations sur le déplacement de la caméra [Hartley 01]. L'approximation liée au calcul de l'homographie peut être compensée en utilisant de multiples homographies et donc de multiples plans pour calculer la pose [Simon 02]. Ces homographies permettent d'estimer, en plus de la pose, les paramètres des plans suivis simultanément [Mei 07].

Ce dernier article s'approche du principe du SLAM si ce n'est qu'il suit toujours les mêmes plans, et ne traite pas des problématiques de l'ajout dynamique de nouveaux éléments. A notre connaissance, le premier papier faisant état de l'utilisation de la notion de plan dans le SLAM est [Molton 04]. Celui-ci considère les régions autour des points suivis comme localement planaires et estime les normales de ces plans (en dehors du vecteur d'état cependant) pour faciliter le suivi des points. Plus proche des travaux de cette thèse, les travaux décrits dans [Pietzsch 08] permettent d'estimer dans le vecteur d'état les normales des plans. Contrairement à cette thèse, chaque point suivi possède son propre plan et aucune notion de mutualisation des plans n'est évoquée.

Cette mutualisation des plans, permettant de réduire la taille du vecteur d'état, est utilisée dans [Gee 07]. Dans cet article, un point est enregistré non plus en 3D mais en 2D dans l'espace du plan auquel il est rattaché. Contrairement à cette thèse, cet article conserve la notion de point et utilise "simplement" le plan pour mutualiser l'information de profondeur de chaque point dans le vecteur d'état. Pour finir, on citera [Silveira 08] qui décrit un système de SLAM utilisant des plans. Utilisant une méthode analytique proche de [Mei 07], elle est différente des méthodes statistiques habituelles comme cette thèse.

2.7 Intégration des plans dans le SLAM

L'implémentation la plus courante d'un système de vision SLAM repose sur l'utilisation d'une carte de points 3D définis dans un repère global. En reprenant les notations précédemment définies du SLAM (section 2.3), les éléments de la carte sont

$$\mathbf{f}_i = \begin{bmatrix} {}^oX \\ {}^oY \\ {}^oZ \end{bmatrix} \tag{2.60}$$

L'espace est considéré naturellement comme un espace Euclidien de dimension 3. Un point 3D étant mesuré dans l'image comme un point 2D, le modèle de mesure associé est le modèle de projection perspective. Chaque point est défini géométriquement indépendamment des autres. La mesure peut alors être fournie par un tracking de point classique comme le KLT [Shi 94]. La mesure sera comparée à son estimée (calculée à l'aide de la carte de points et la pose de la caméra) pour mettre à jour le système. De nombreux travaux ont porté sur l'étude théorique d'un tel système et des améliorations possibles [Sola 05, Montiel 06, Davison 07, Pinies 08]. Il semble raisonnable de partir de ce concept, éprouvé, pour tenter d'introduire notre notion de plan dans le SLAM.

Notre souhait d'appréhender l'environnement, au moins partiellement, comme un ensemble de surfaces planes peut être traduit de différentes façons. On peut considérer que ces surfaces planes sont un ensemble fini de points qui forment une structure géométrique : le plan. Inversement, et si ce plan est connu, la surface plane peut être définie comme l'ensemble des points situés sur ce plan, c'est-à-dire qui respectent la contrainte définie par l'équation (2.59). De cette contrainte est déduite la profondeur correspondante à chaque point **mesuré** appartenant au plan.

$$a\frac{X}{Z} + b\frac{Y}{Z} + c\frac{Z}{Z} = \frac{d}{Z} \tag{2.61}$$

$$ax + by + c = \frac{d}{Z} \tag{2.62}$$

$$\frac{1}{Z} = \frac{ax + by + c}{d} \tag{2.63}$$

Le plan factorise la troisième dimension des points. Au lieu d'estimer l'ensemble des profondeurs des points, il est tout à fait possible d'estimer les paramètres du plan pour obtenir les mêmes résultats. L'estimation des paramètres du plan profite de plus à tous les points lui appartenant mais non visibles pour le moment.

2.7.1 Une première solution basée sur les points

Une première solution imaginable est donc de n'enregistrer dans le vecteur d'état que la mesure d'un point ainsi que le plan sur lequel il est disposé. La carte étant de préférence indépendante de la caméra, la mesure est transformée de manière à être définie dans $\mathcal{F}_o$. Un point est alors défini par deux éléments dans la carte :

$$\mathbf{f}_i = \begin{bmatrix} \frac{^oX}{^oZ} \\ \frac{^oY}{^oZ} \end{bmatrix} \tag{2.64}$$

et

$$\mathbf{f}_j = \begin{bmatrix} \mathbf{n} \\ d \end{bmatrix} \tag{2.65}$$

où $\mathbf{f}_i$ peut-être déterminé en utilisant la mesure des coordonnées du point dans l'image et l'homographie entre la pose actuelle et le repère global ${}^o\mathbf{H}_c$. Cette dernière homographie est connue puisqu'elle se construit à partir des paramètres du plan $\mathbf{f}_j$ et de la pose courante. Cette solution est très similaire à la solution générique du SLAM puisqu'il suffit d'estimer la profondeur de chaque point à l'aide du plan pour retrouver le point 3D. Cependant cette solution induit des difficultés pour l'initialisation puisqu'elle nécessite de connaitre de manière relativement précise les paramètres des plans *a priori* si l'on souhaite éviter des problèmes de linéarisation. Cette solution nécessite de plus de rajouter des informations au vecteur d'état pour chaque point, ce qui n'est pas souhaitable compte tenu de la complexité en $O(n^2)$ de l'EKF. Notre objectif est de réduire au maximum la taille de ce vecteur d'état et donc la complexité calculatoire.

2.7.2 Une solution basée sur les homographies

Notre solution est de ne plus considérer la carte comme une localisation d'éléments géométriques à proprement parler mais comme un ensemble de variables aléatoires permettant de définir les mouvements mesurés. Prenons un point ${}^{c1}\mathbf{p}$ dans l'image $\mathcal{I}$ (la reprojection dans le repère $c1$) appartenant à un plan $\mathbf{pl}$: ce point, puisqu'il est choisi arbitrairement, est connu parfaitement. L'estimation des coordonnées ${}^{c1}\mathbf{p}$ dans l'image $\mathcal{I}'$ (la reprojection dans le repère $c2$) est comme nous l'avons vu, l'application de l'homographie ${}^{c2}\mathbf{H}_{c1}$ sur le point ${}^{c1}\mathbf{p}$. Si erreur dans l'estimation il y a, on peut considérer qu'elle provient d'une erreur dans l'estimation de l'homographie, et donc dans l'estimation des paramètres de transformation ${}^{c2}\mathbf{R}_{c1}$, ${}^{c2}\mathbf{t}_{c1}$ et des paramètres du plan ${}^{c1}\mathbf{pl}$.

Afin de faciliter le processus d'estimation, il est donc préférable que chaque élément estimé soit indépendant des autres afin de ne pas créer de redondance inutile dans les informations et les incertitudes. Ce qui nous amène à exprimer les informations nécessaires à l'estimation dans un repère commun et non plus dans un repère, lié a la caméra, lui même estimé.

2.7.2.1 Les paramètres de transformation

L'homographie nécessite la connaissance de ${}^{c2}\mathbf{R}_{c1}$ et de ${}^{c2}\mathbf{t}_{c1}$. Il est possible d'obtenir ces deux informations à partir d'informations exprimées dans un repère global

$$
\begin{aligned}
{}^{c2}\mathbf{M}_{c1} &= {}^{c2}\mathbf{M}_{o}\,{}^{o}\mathbf{M}_{c1} && (2.66)\\
& && (2.67)
\end{aligned}
$$

et donc

$$
\begin{aligned}
{}^{c2}\mathbf{R}_{c1} &= {}^{c2}\mathbf{R}_{o}\,{}^{o}\mathbf{R}_{c1} && (2.68)\\
&= {}^{c2}\mathbf{R}_{o}\,{}^{c1}\mathbf{R}_{o}^{\mathsf{T}} && (2.69)\\
{}^{c2}\mathbf{t}_{c1} &= {}^{c2}\mathbf{R}_{o}\,{}^{c1}\mathbf{t}_{o} + {}^{c2}\mathbf{t}_{o} && (2.70)
\end{aligned}
$$

2.7.2.2 Les paramètres du plan

L'homographie nécessite la connaissance des paramètres de ${}^{c1}\mathbf{pl}$. Ceux-ci peuvent être exprimés à partir de ${}^{o}\mathbf{pl}$.

$$
\begin{aligned}
{}^{c1}\mathbf{n} &= {}^{c1}\mathbf{R}_{o}\,{}^{o}\mathbf{n} && (2.71)\\
{}^{c1}d &= {}^{o}d - {}^{c1}\mathbf{n}^{\mathsf{T}}\,{}^{c1}\mathbf{t}_{o} && (2.72)
\end{aligned}
$$

Considérons maintenant le repère $\mathcal{F}_{c2}$ comme étant notre pose actuelle, $\mathcal{F}_{c1}$ le repère où un point a été détecté la première fois et ${}^{o}\mathbf{pl}$ le plan associé au point (connu ou non à l'avance dans une base de données). ${}^{o}\mathbf{R}_{c2}$ et ${}^{o}\mathbf{t}_{c2}$ sont d'ores et déja estimés dans notre vecteur d'état (la partie localisation du SLAM). Il reste donc à estimer ${}^{o}\mathbf{R}_{c1}$, ${}^{o}\mathbf{t}_{c1}$ et ${}^{o}\mathbf{pl}$. C'est précisément ces 3 paramètres qui constitueront les éléments de la carte pour un point suivi. Une région est alors déterminée dans le vecteur d'état par 9 paramètres mais dont l'utilisation est mutualisée sur de nombreuses régions.

2.7.2.3 Ajout d'un plan

Lorsque un plan ${}^{o}\mathbf{pl}$ est utilisé pour la première fois, il est ajouté au vecteur d'état une seule et unique fois et utilisé pour toutes les mesures qui le concerne. On suppose pour l'instant que lorsqu'un nouveau plan est utilisé, on connait ses paramètres par "magie" (la section 4.3 expliquera comment connaître ces paramètres). Le plan est défini entre autres par un vecteur, la normale au plan. Ce vecteur normal doit avoir une norme de 1. Si celui-ci est estimé à travers un filtre de Kalman, il est nécessaire de procéder à la renormalisation de ce vecteur et à la mise à jour correspondante de la matrice de covariance de l'état. Pour éviter ces calculs inutiles, le vecteur normal au plan est remplacé par une orientation sous forme de coordonnées sphériques (ϕ et θ).

$$
\mathbf{n} = \begin{bmatrix} sin(\phi)cos(\theta) \\ sin(\phi)sin(\theta) \\ cos(\phi) \end{bmatrix} \qquad (2.73)
$$

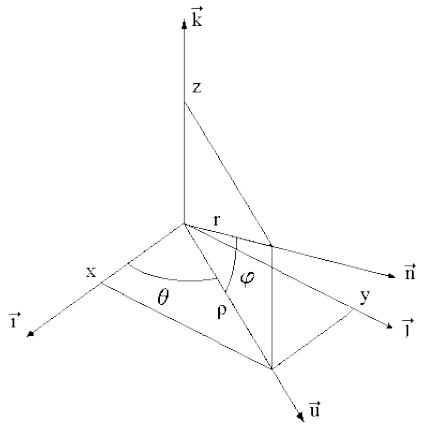


FIGURE 2.6 – Illustration des coordonnées sphériques

La fonction d'augmentation g_l est simplement l'application identité et le paramètre l le vecteur du plan. la covariance associée $\mathbf{D}$ est donc ajoutée simplement à l'intersection des lignes et des colonnes de $\mathbf{P}$ correspondant au plan.

$$
\begin{aligned}
l &= {}^{o}\mathbf{pl} && (2.74)\\
&= \begin{bmatrix} {}^{o}\phi \\ {}^{o}\theta \\ {}^{o}d \end{bmatrix} && (2.75)\\
\mathbf{G}_p &= \mathbf{0} && (2.76)\\
\mathbf{G}_l &= \mathbf{I} && (2.77)\\
& && (2.78)
\end{aligned}
$$

À condition de disposer d'un plan totalement défini comme support, un plan peut être exprimé comme une transformation constante d'un autre. Le plan est alors fonction d'un plan de référence et d'une transformation. De cette manière, on peut exprimer des contraintes sur la structure de la scène sans complexifier davantage notre algorithme. Citons par exemple la possibilité d'imposer la notion d'orthogonalité entre les plans dans une pièce d'intérieur. Les paramètres absolus des plans sont inconnus ou du moins incertains, mais on sait qu'ils forment 2 par 2 des angles droits. Hormis le premier plan, il n'est alors plus nécessaire d'estimer l'orientation des autres plans de la pièce. Ceci réduit le nombre de paramètres et renforce la cohérence du calcul.

2.7.2.4 Ajout d'une caméra de référence

Il a été expliqué que le calcul de l'homographie nécessite l'estimation d'une pose de référence $\mathcal{F}_{c1}$. Cette pose de référence peut être choisie comme la pose à laquelle on a observé pour la première fois un élément dans l'image. Lorsque un élément est

mesuré pour la première fois, il est alors nécessaire d'initialiser dans le vecteur d'état cette pose de référence. Cette dernière est à cet instant la pose actuelle et peut donc être initialisée en augmentant le vecteur d'état avec une copie de la pose courante.

$$l = \begin{bmatrix} {}^{c2}\mathbf{t}_o \\ {}^{c2}\mathbf{r}_o \end{bmatrix} \tag{2.79}$$

$$\mathbf{G}_p = \mathbf{I} \tag{2.80}$$

$$\mathbf{G}_l = \mathbf{0} \tag{2.81}$$

Cette caméra de référence est commune à tous les élements détectés dans l'image mesurée à cette pose. Le vecteur d'état résultant de ces opérations peut par exemple devenir :

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ {}^{o}\mathbf{pl}_1 \\ {}^{c1}\mathbf{t}_o \\ {}^{c1}\mathbf{r}_o \end{bmatrix} \tag{2.82}$$

2.7.2.5 Modèle de mesure basé point

Chaque élément détecté est enregistré avec le couple d'éléments (caméra de référence + plan) de la carte qui permettent de le décrire ainsi que ses propres informations d'origine (coordonnées ${}^{c1}\mathbf{p}$, texture, etc.). On considère toujours pour la démonstration que le repère $\mathcal{F}_{c1}$ est la pose de référence pour le point qui fait l'objet de l'estimation et que $\mathcal{F}_{c2}$ est la pose courante.

Une possibilité pour notre estimateur est de comparer une estimation des coordonnées d'un point avec la mesure de ce même point dans l'image courante. Le modèle de mesure est alors considéré comme une application de l'homographie sur ${}^{c1}\mathbf{p}$.

$$\mathbf{z} = w(\mathbf{K}\,{}^{c2}\mathbf{H}_{c1}\mathbf{K}^{-1}\,{}^{c1}\mathbf{p}) \tag{2.83}$$

où w(.) est la fonction de projection perspective, l'homographie étant valide à un facteur d'échelle près. Rappelons que $\mathbf{K}$ est la matrice de passage de mètres en pixels.

$$w(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}) = \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{bmatrix} \tag{2.84}$$

La Jacobienne du modèle de mesure $\mathbf{Z}$, qui va – rappelons le – permettre de répartir l'innovation de la mesure dans les différentes composantes du vecteur d'état, est décomposable en autant de parties qu'il y a d'éléments utilisés pour la construction de l'homographie dans le vecteur d'état.

$$\mathbf{Z} = \left[\begin{array}{ccccccc} \frac{\partial \mathbf{h}}{\partial \mathbf{x_{c2}}} & \cdots & \frac{\partial \mathbf{h}}{\partial \mathbf{pl}} & \cdots & \frac{\partial \mathbf{h}}{\partial \mathbf{x_{c1}}} & \cdots \end{array} \right] \tag{2.85}$$

Seule la matrice d'homographie dépend des paramètres en cours d'estimation du vecteur d'état. Cette Jacobienne peut donc être réécrite en factorisant certains calculs :

$$\mathbf{Z} = \left[\begin{array}{c} \frac{\partial \mathbf{w(M)}}{\partial \mathbf{M}} \frac{\partial \mathbf{M}}{\partial \mathbf{G}} \frac{\partial \mathbf{G}}{\partial \mathbf{H}} \frac{\partial \mathbf{H}}{\partial \mathbf{x_{c2}}} \\ \vdots \\ \frac{\partial \mathbf{w(M)}}{\partial \mathbf{M}} \frac{\partial \mathbf{M}}{\partial \mathbf{G}} \frac{\partial \mathbf{G}}{\partial \mathbf{H}} \frac{\partial \mathbf{H}}{\partial \mathbf{pl}} \\ \vdots \\ \frac{\partial \mathbf{w(M)}}{\partial \mathbf{M}} \frac{\partial \mathbf{M}}{\partial \mathbf{G}} \frac{\partial \mathbf{G}}{\partial \mathbf{H}} \frac{\partial \mathbf{H}}{\partial \mathbf{x_{c1}}} \\ \vdots \end{array} \right]^{\mathtt{T}} \tag{2.86}$$

où

$$\mathbf{G} = \mathbf{K}^{c2}\mathbf{H}_{c1}\mathbf{K}^{-1} \tag{2.87}$$

$$\mathbf{M} = \mathbf{G}^{c1}\mathbf{p} \tag{2.88}$$

La matrice $\mathbf{R}$ (la matrice de covariance de la mesure) se définit comme l'incertitude des coordonnées mesurées et donc en quelque sorte l'ellipse dans lequel le point mesuré peut être localisé. L'information de transformation étant projetée sur des points, il est bien évidemment nécessaire de procéder à l'étape de mise à jour de l'EKF pour de multiples points (même pour un seul plan). Cette mise à jour peut également être faite en une seule fois en agrégeant les différentes matrices utilisées pour les points. On se retrouve de toute manière confronté à un alourdissement des besoins en ressources calculatoires pour chaque image.

2.7.2.6 Modèle de mesure basé sur les homographies

Un chapitre ultérieur sera consacré à l'étude des méthodes utilisées pour le suivi dans l'image des informations d'intéret. Sans rentrer dans des considérations détaillées et redondantes, l'utilisation de structures planes incite évidemment à utiliser des méthodes de suivi dans l'image qui profitent de cette contrainte géométrique. De cette méthode de suivi dans l'image dépend la mesure qui sera fournie à notre estimateur. Le tracker cherche une homographie minimisant une différence d'intensité lumineuse (suivant le même principe que le KLT). Le tracker fournit à chaque image une homographie en pixels. Cette homographie, qui était précédemment en quelque sorte reprojetée dans l'image à travers les points, est donc maintenant directement disponible.

En passant par des points "virtuels" et en les reprojetant avec l'homographie obtenue, on peut obtenir les mêmes mesures et donc utiliser le même modèle de mesure que dans la section précédente. Une solution plus élégante et n'ayant plus les problèmes de coût calculatoire est cependant envisageable. L'homographie retournée par le tracker est la matrice $\mathbf{G}$ précédemment définie et va être utilisée directement comme mesure.

Jacobienne de z en fonction du plan

$$\frac{\partial \mathbf{H}}{\partial \mathbf{pl}} = \frac{\partial \mathbf{H}}{\partial \mathbf{ns(^o pl)}} \frac{\partial \mathbf{ns(^o pl)}}{\partial \mathbf{^o pl}} \tag{2.89}$$

$$\frac{\partial \mathbf{ns(^o pl)}}{\partial \mathbf{^o pl}} = \begin{bmatrix} -sin(\phi)sin(\theta) & cos(\phi)cos(\theta) & 0 \\ sin(\phi)cos(\theta) & cos(\phi)sin(\theta) & 0 \\ -sin(\phi) & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.90}$$

$$\frac{\partial \mathbf{H}}{\partial \mathbf{ns(^o pl)}} = -(\frac{-1}{{}^{c1}d^2}({}^{c1}\mathbf{n}^{\mathsf{T}} \otimes \mathbf{I}){}^{c2}\mathbf{t}_{c1} \frac{\partial \mathbf{^{c1}d}}{\partial \mathbf{^o n}} + (\mathbf{I} \otimes \frac{{}^{c2}\mathbf{t}_{c1}}{{}^{c1}d}) \frac{\partial \mathbf{^{c1}n}}{\partial \mathbf{^o n}}) \tag{2.91}$$

$$\frac{\partial \mathbf{^{c1}d}}{\partial \mathbf{^o n}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} - {}^{c1}\mathbf{t}_o^T \frac{\partial \mathbf{^{c1}n}}{\partial \mathbf{^o n}} \tag{2.92}$$

$$\frac{\partial \mathbf{^{c1}n}}{\partial \mathbf{^o n}} = {}^{c1}\mathbf{R}_o \mathbf{I}_{3\times 4} \tag{2.93}$$

Jacobienne de z en fonction de la caméra de référence

$$\frac{\partial \mathbf{H}}{\partial \mathbf{x_{c1}}} = \frac{\partial \mathbf{^{c2}R_{c1}}}{\partial \mathbf{x_{c1}}} - (({}^{c1}\mathbf{n} \otimes \mathbf{I})(\frac{1}{{}^{c1}d} \frac{\partial \mathbf{^{c2}t_{c1}}}{\partial \mathbf{x_{c1}}} + \frac{-{}^{c2}\mathbf{t}_{c1}}{{}^{c1}d^2} \frac{\partial \mathbf{^{c1}d}}{\partial \mathbf{x_{c1}}}) \tag{2.94}$$

$$+ \quad (\mathbf{I} \otimes \frac{{}^{c2}\mathbf{t}_{c1}}{{}^{c1}d}) \frac{\partial \mathbf{^{c1}n}}{\partial \mathbf{x_{c1}}}) \tag{2.95}$$

$$\frac{\partial \mathbf{^{c2}R_{c1}}}{\partial \mathbf{x_{c1}}} = (\mathbf{I} \otimes {}^{c2}\mathbf{R}_o) \frac{\partial \mathbf{^{c1}R_o^T}}{\partial \mathbf{x_{c1}}} \tag{2.96}$$

$$\frac{\partial \mathbf{^{c1}R_o^T}}{\partial \mathbf{x_{c1}}} = (-{}^{c1}\mathbf{R}_o \otimes {}^{c1}\mathbf{R}_o^{\mathsf{T}}) \frac{\partial \mathbf{^{c1}R_o}}{\partial \mathbf{x_{c1}}} \tag{2.97}$$

$$\frac{\partial \mathbf{^{c2}t_{c1}}}{\partial \mathbf{x_{c1}}} = -(\mathbf{x_{c1}}^T \otimes \mathbf{I}) \frac{\partial \mathbf{^{c2}R_{c1}}}{\partial \mathbf{x_{c1}}} + \mathbf{^{c2}R_{c1}} \frac{\partial \mathbf{^{c1}t_o}}{\partial \mathbf{x_{c1}}} \tag{2.98}$$

$$\frac{\partial \mathbf{^{c1}n}}{\partial \mathbf{x_{c1}}} = ({}^{o}\mathbf{n}^{\mathsf{T}} \otimes \mathbf{I}) \frac{\partial \mathbf{^{c1}R_o}}{\partial \mathbf{x_{c1}}} \tag{2.99}$$

$$\frac{\partial \mathbf{^{c1}d}}{\partial \mathbf{x_{c1}}} = -({}^{c1}\mathbf{t}_o^{\mathsf{T}}(\mathbf{I} \otimes {}^{o}\mathbf{n}^{\mathsf{T}}) \frac{\partial \mathbf{^{c1}R_o^T}}{\partial \mathbf{x_{c1}}} + {}^{c1}\mathbf{n}^{\mathsf{T}} \frac{\partial \mathbf{^{c1}t_o}}{\partial \mathbf{x_{c1}}}) \tag{2.100}$$

Jacobienne de z en fonction de la caméra courante

$$\frac{\partial \mathbf{H}}{\partial \mathbf{x_{c2}}} = \frac{\partial \mathbf{^{c2}R_{c1}}}{\partial \mathbf{x_{c2}}} - \frac{1}{{}^{c1}d}({}^{c1}\mathbf{n} \otimes \mathbf{I}) \frac{\partial \mathbf{^{c2}t_{c1}}}{\partial \mathbf{x_{c2}}} \tag{2.101}$$

$$\frac{\partial \mathbf{^{c2}R_{c1}}}{\partial \mathbf{x_{c2}}} = ({}^{c1}\mathbf{R}_o \otimes \mathbf{I}) \frac{\partial \mathbf{^{c2}R_o}}{\partial \mathbf{x_{c2}}} \tag{2.102}$$

$$\frac{\partial \mathbf{^{c2}t_{c1}}}{\partial \mathbf{x_{c2}}} = -({}^{c1}\mathbf{x}_o^{\mathsf{T}} \otimes \mathbf{I}) \frac{\partial \mathbf{^{c2}R_{c1}}}{\partial \mathbf{x_{c2}}} + \frac{\partial \mathbf{^{c2}t_o}}{\partial \mathbf{x_{c2}}} \tag{2.103}$$

FIGURE 2.7 – Définition des Jacobiennes de la matrice d'homographie en fonction des paramètres du vecteur d'état

Rappelons que le filtre de Kalman va calculer la différence entre deux vecteurs. Une homographie étant valide à un facteur d'échelle près et la comparaison nécessitant que les deux vecteurs soient dans le même espace, la première chose à faire est de normaliser les homographies mesurées et estimées en divisant chaque composante par son dernier terme (La fonction $wm()$ représentera cette normalisation).

Cette normalisation est nécessairement suivie d'une transformation des matrices pour obtenir un vecteur correspondant. Cette vectorisation est effectuée par convenance – et en accord avec la méthode de dérivation – colonne par colonne.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}^s = \begin{bmatrix} a & d & g & b & e & h & c & f \end{bmatrix}^{\mathtt{T}} \tag{2.104}$$

Le dernier terme, inutile, est retiré du vecteur résultant.

$$\mathbf{z} = (wm(\mathbf{K}^{c2}\mathbf{H}_{c1}\mathbf{K}^{-1}))^s \tag{2.105}$$

Le problème de cette solution provient de la "destructuration" des matrices. Il n'est pas possible de calculer une distance entre deux matrices comme s'il s'agissait d'un simple vecteur. Il est nécessaire de rétablir les corrélations et les contraintes structurelles présentes dans la matrice d'homographie. Ceci est fait en choisissant prudemment la matrice de bruit de la mesure, qui contient entre autres choses cette structuration. Le dernier point à résoudre est donc la construction de cette matrice $\mathbf{R}$ de dimension 8×8.

Prenons n points virtuels dans l'image ${}^{c1}\mathbf{p}_i$ dont les coordonnées correspondent à des points suivis dans le repère $\mathcal{F}_{c1}$. Soit ${}^{c2}\mathbf{p}_i$ ces mêmes points reprojetés dans l'image de $\mathcal{F}_{c2}$. ${}^{c1}\mathbf{p}_i$ étant des constantes, l'incertitude $\Sigma_{p'}$ est commme précédemment expliquée le résultat de l'incertitude sur l'homographie. L'équation suivante est donc valide [2] :

$$\Sigma_{p'} = \mathbf{J_h}\Sigma_h\mathbf{J_h}^{\mathtt{T}} \tag{2.106}$$

avec $\mathbf{J_h}$ la Jacobienne $\frac{\partial \mathbf{w}(\mathbf{Hp})}{\partial \mathbf{H}}$ ($\mathbf{H}$ est ici l'homographie mesurée). Cette équation permet de retrouver Σ_h.

$$\Sigma_h = (\mathbf{J_h}^{\mathtt{T}}\Sigma_{p'}^{-1}\mathbf{J_h})^{-1} \tag{2.107}$$

En définissant $\Sigma_{p'}$ arbitrairement pour avoir une incertitude moyenne dans les coordonnnées d'un point reprojeté, et en utilisant suffisamment de points, on peut construire la matrice de corrélation de la mesure de l'homographie. Dans nos essais, on définit $\Sigma_{p'}$ comme une matrice diagonale contenant la même variance pour chaque point. Le lecteur notera qu'on utilise toujours cette notion de point, toutefois en dehors du processus de mise à jour de l'EKF : sans impact sur la taille de la matrice de gain et donc sur la complexité calculatoire.

2. Soit une variable normale $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Si on transforme μ par une fonction $a(\boldsymbol{\mu})$, la covariance est mise à jour en utilisant $\frac{\partial \mathbf{a}}{\partial \mu}\boldsymbol{\Sigma}\frac{\partial \mathbf{a}}{\partial \mu}^{\mathtt{T}}$

2.7.3 Résultats

2.7.3.1 Démonstration de fonctionnement du filtre

Translation. Le premier test réalisé est celui permettant de vérifier le bon fonctionnement des différents modèles utilisés et des paramètrages choisis. Sans penser aux problématiques propres au SLAM, ce test a pour but de valider la convergence dans un cas idéal. On considère un environnement virtuel numérique composé de 3 plans non parallèles. Ces plans sont ajoutés avec une incertitude nulle dans la carte. L'état de départ est également défini comme parfaitement connu et établi avec tous les paramètres à 0 en position et rotation. À chaque itération, une homographie est calculée à partir des positions désirées et ce pour chaque plan. Pendant 100 itérations, la caméra va effectuer une translation vers la gauche, et le même nombre d'itérations vers la droite ensuite.

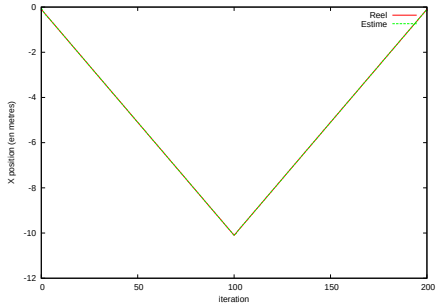


FIGURE 2.8 – Simulation 1 : Position en X de la caméra

La figure 2.9 montre la différence entre l'estimation du filtre et la pose 'réelle' au même instant. Autour de l'itération 0 et de l'itération 100, le lecteur peut observer une erreur soudaine. Ceci est dû au changement dans la direction de la caméra et donc au non respect du modèle d'évolution. Le modèle de mise à jour va permettre alors de corriger les paramètres de déplacement afin de reconverger vers une solution correcte. Ce comportement est commun à tout système d'estimation par filtrage. L'erreur est quasi nulle pour le reste des itérations, ce qui démontre par l'exemple le bon fonctionnement de nos modèles dans le cas d'un mouvement en translation simple.

Rotation La deuxième simulation effectue une rotation de $\frac{\pi}{4}$ de la caméra autour de l'origine du repère $\mathcal{F}_o$, le long de l'axe y puis fait le chemin inverse. Les mêmes commentaires qui s'appliquaient à la figure 2.9 s'appliquent de la même manière à la figure 2.11. Les résultats sont concluants et correspondent aux attentes.

2.7.3.2 Démonstration des avantages avec une carte

Les méthodes d'initialisation de la carte n'ayant pas encore été exposées, la présente démonstration s'attache à montrer le fonctionnement de notre méthode dans le cas où les paramètres des plans sont connus *a priori* et où les sélections des éléments de

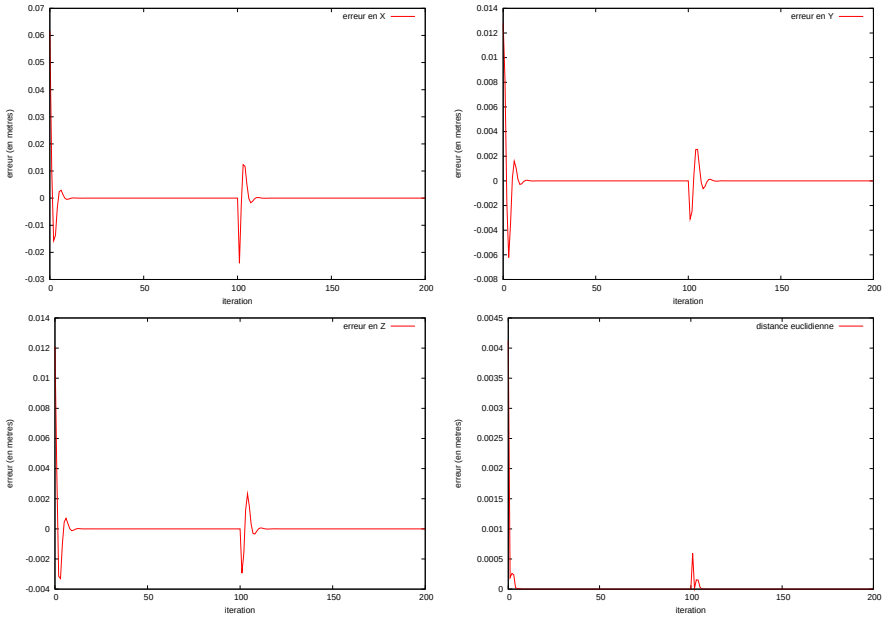


FIGURE 2.9 – Simulation 1 : Erreur d'estimation

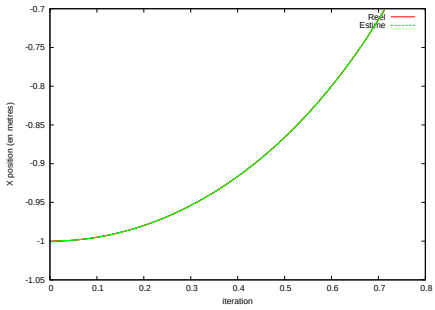


FIGURE 2.10 – Simulation 2 : Position en X de la caméra

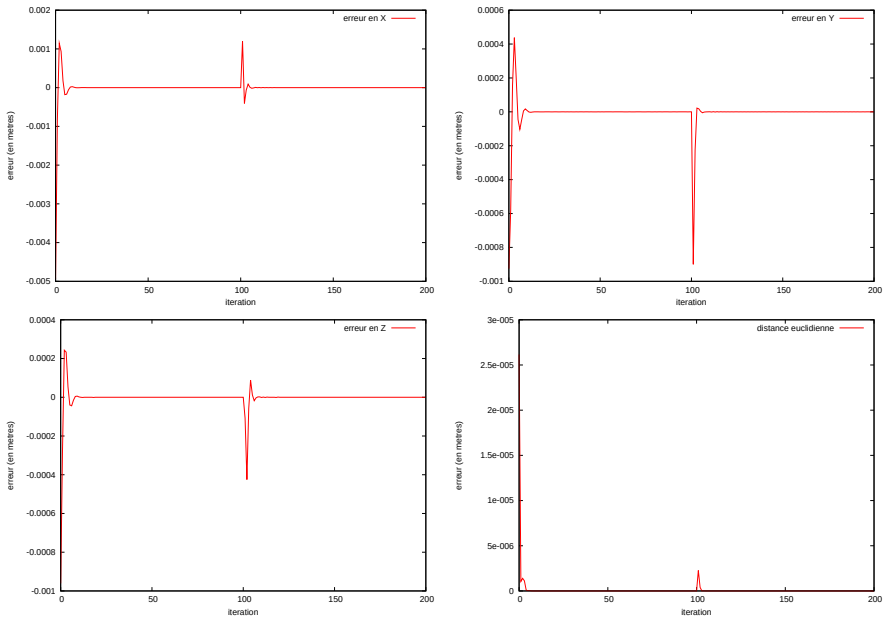


FIGURE 2.11 – Simulation 2 : Erreur d'estimation

la scène mesurés sont faites automatiquement par une méthode externe. Considérons une boite rectangulaire où chaque côté est flanqué d'une image permettant de fournir suffisamment de texture pour le suivi. La séquence consiste en un mouvement circulaire de la caméra autour de la boite, suivi du même type de mouvement dans l'autre sens. La boite est composée de 6 plans dont les paramètres sont connus dans un repère global. Celui-ci est un coin de la boite par convenance. La pose initiale est déterminée par une méthode externe (méthode classique de calcul de pose à partir de 4 points référencés dans l'espace). Chaque texture/image est référencée dans une base de données ainsi que le plan sur lequel elle est située. Les zones à suivre sont détectées en utilisant la reconnaissance de templates (voir section 4.1).

Deux estimations sont effectuées en parallèle. La première estimation profite de notre méthode et considère une incertitude dans les éléments de la carte. La seconde est un filtre de Kalman classique utilisant les mêmes modèles mais considérant la carte comme parfaite et constante. Pour démontrer l'utilité de notre méthode, on introduit une erreur dans l'inclinaison d'un des plans. Cette erreur, de 5 degrés, est volontairement réduite pour pouvoir faciliter la visualisation. La série d'images 2.13 illustre cette séquence dans les moments clés du mouvement. Le repère bleu étant la pose de la caméra estimée sans définir d'incertitudes sur les plans (donc par un EKF classique) et le repère rouge notre méthode de SLAM.

La figure 2.12 permet de remarquer la dérive constante dans l'estimation de la poste avec la méthode standard. Elle s'explique par le fait qu'il trouve une correspondance approximative des mesures pour une pose donnée avec ce modèle de l'environnement.

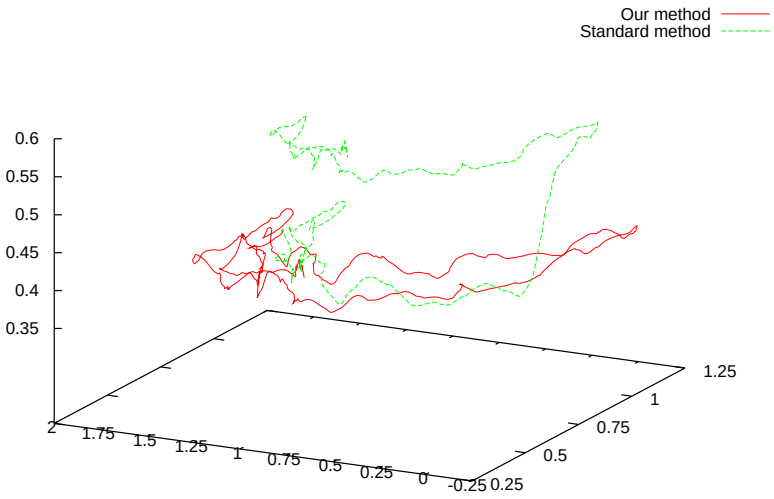


FIGURE 2.12 – Position de la caméra suivant la méthode utilisée. Le repère bleu est la pose de la caméra estimée sans définir d'incertitudes sur les plans et le repère rouge la pose utilisant notre méthode de SLAM.

2.8 Conclusion

Ce chapitre introduit le concept de notre estimateur et les *a priori* effectués. Cet estimateur nécessite de fonctionner dans un cadre applicatif réel, et nécessite donc de prendre en compte son environnement pour fonctionner à travers l'utilisation des mesures images. La suite de cette thèse s'attache à décrire l'interface qui nous permet d'utiliser concrètement cet estimateur.

Dans ce chapitre, une adaptation du principe de la Localisation et Cartographie Simultanée (SLAM) pour les plans a été proposée. Cette méthode, basée sur un filtre de Kalman étendu, permet de localiser une caméra monoculaire dans un environnement constitué de plans dont les paramètres sont incertains tout en affinant ces derniers. Alors que les méthodes traditionnelles de SLAM utilisent un modèle de prédiction de mouvement "basique", nous avons utilisé une méthode plus respectueuse des lois de la dynamique rarement utilisée dans la littérature du SLAM. Notre méthode fournit une manière originale de prendre en compte les plans de l'environnement en utilisant des homographies comme mesures directes de l'environnement. Ce chapitre a permis de prouver le fonctionnement théorique et pratique de notre estimateur à travers quelques démonstrations. Cette méthode profite des nombreuses études effectuées sur le SLAM monoculaire basé point dont nos équations dérivent directement.

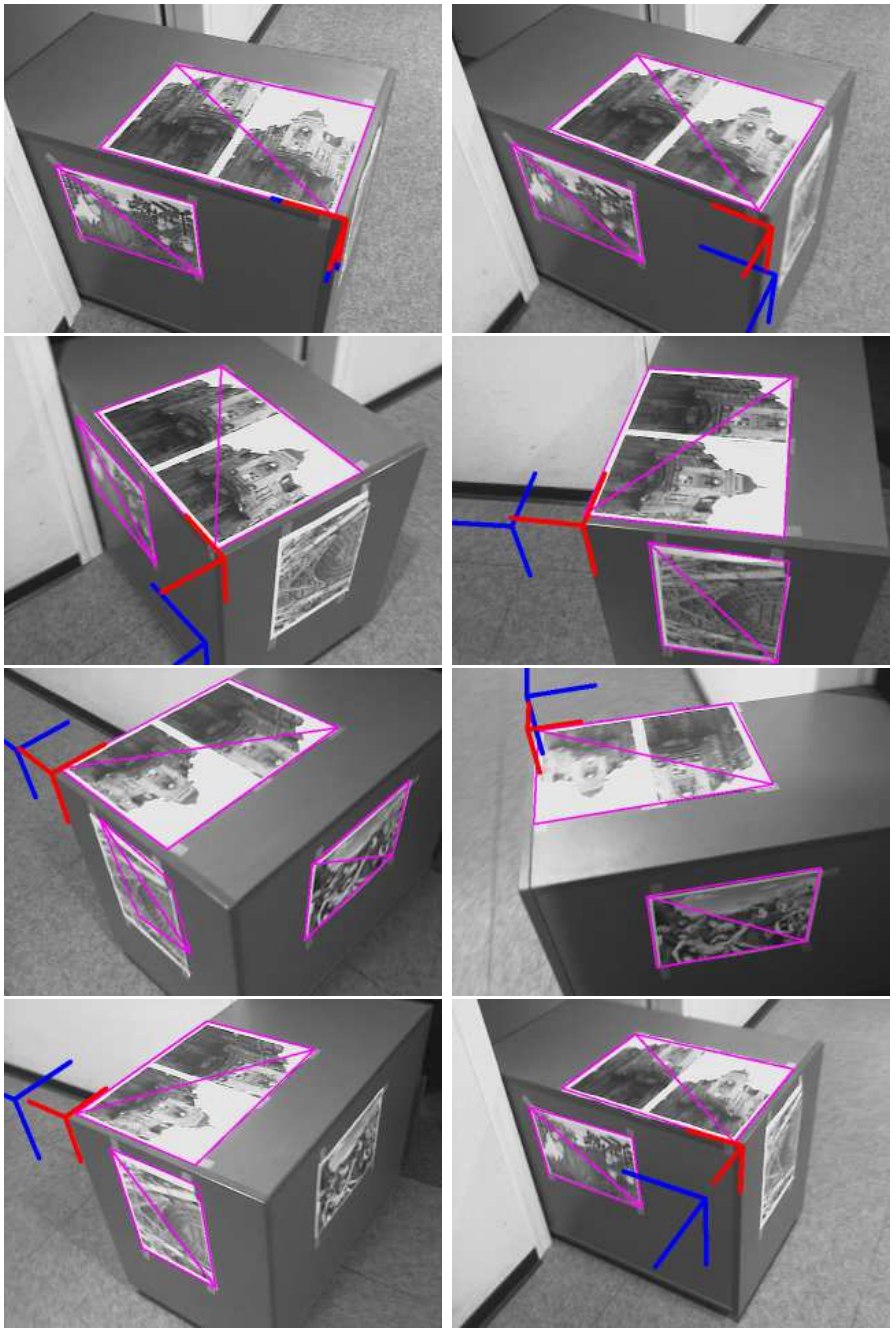


FIGURE 2.13 – Démonstration de l'utilité de l'incertitude

Ce chapitre considérait les mesures des mouvements dans l'image comme acquises et il nous faut maintenant expliquer quelle méthode est utilisée pour mesurer cette homographie. Autre problématique, ce chapitre considérait que les zones à suivre étaient choisies "magiquement". Plusieurs solutions ont été développées suivant le cadre d'utilisation et sont expliquées ultérieurement. Finalement, on expliquera comment notre estimateur pourra être utilisé dans le cas où aucune information n'est connue sur l'environnement *a priori* : Comment estimer les paramètres des plans sans indices autres que le mouvement dans l'image ?

Chapitre 3

Le suivi de formes dans une séquence d'images

L'estimation du déplacement de la caméra nécessite d'appréhender les mouvements de l'environnement dans le repère de la caméra. Cet environnement étant partiellement connu à travers sa projection dans les images, les déplacements successifs dans la séquence d'images permettent de mesurer les mouvements de la scène (dûs aux mouvements de la caméra).

Une image fournit un nombre très important d'informations à une fréquence élevée. Une des problématiques du suivi est de sous-échantillonner l'image, en extrayant les informations les plus discriminantes, de manière à rendre le coût calculatoire abordable. Les parties de l'image – dont les propriétés mathématiques laissent penser qu'elles seront plus aisées à suivre, ou dont le mouvement est modélisable de manière simple – seront choisies en fonction de la méthode de suivi utilisée. L'objectif final étant de pouvoir estimer la transformation géométrique entre cette image et une image de référence. Le choix des éléments suivis et, bien sûr, de la méthode considérée a pour objectif de permettre un suivi le plus fiable et le plus robuste possible.

De la capacité du suivi de région à perdurer va dépendre la qualité de l'application de calcul de pose. En effet, même si les méthodes d'estimation permettent de fusionner et d'agréger les différents suivis d'images pour effectuer leur tâche, les informations obtenues directement sont préférables. Le suivi de région étant directement effectué sur le flux d'images, une transformation estimée par ce suivi est préférable. Il est donc primordial d'utiliser une méthode de suivi de région robuste afin de suivre les régions le plus longtemps possible.

Hormis les problèmes liés aux capteurs caméras que l'on a étudiés dans un chapitre antérieur, de nombreux problèmes doivent être pris en compte par les méthodes de suivi de région pour fonctionner.

3.1 Problématique

Quel que soit le type de suivi choisi, le principe reste *a priori* de suivre une zone de l'image en considérant qu'elle va évoluer dans le temps suivant un modèle de mouvement paramétrique prédéterminé. Les paramètres de ce modèle de mouvement seront estimés afin d'obtenir un résultat concordant avec les mesures dans l'image courante. Savoir comment va évoluer une région de l'image peut être extrêmement complexe et dépend de ses propriétés intrinsèques (forme, matériau) et des propriétés de son environnement (la géométrie des objets de l'environnement qui peuvent provoquer une occlusion, les sources lumineuses, etc.). Les modèles utilisés en suivi sont simplifiés en fonction des connaissances *a priori* que l'on possède sur la scène et des restrictions calculatoires. L'espace de recherche du suivi est réduit de manière plus ou moins importante en supposant que cet espace une fois réduit permette tout de même de trouver des solutions correctes pour les régions sélectionnées et l'importance du déplacement entre l'image de référence et l'image courante. Il n'existe pas, de toute manière, de modèle de déplacement 2D générique qui permette de prendre en compte tous les mouvements 3D possibles de la caméra avec une scène quelconque. Le terme de robustesse à une perturbation est utilisé pour décrire les capacités d'une méthode de suivi à pouvoir suivre un élément dans un flux d'images malgré cette perturbation.

Ce processus de suivi est une interface entre le flux d'images et l'estimation de la pose. Il est donc primordial que le suivi soit à même d'être critique sur son propre résultat. Bien que l'estimateur de pose doive être capable de rejeter les mesures géométriquement aberrantes provenant du processus de suivi, ce processus est le seul à même de vérifier la cohérence de l'estimation du déplacement dans l'image. En effet, il est le seul à même de comparer la région transformée avec les paramètres estimés et la "réalité terrain" provenant de l'image courante. Une mesure invalide prise en compte dans le processus d'estimation fait logiquement dériver le résultat final.

3.2 Les possibilités existantes pour le suivi basé image

Une premier type d'approche [Brown 71, Fischler 81, Lowe 87, Drummond 02] et [Marchand 02a, Marchand 02b, Comport 06] est de fixer un *a priori* sur l'objet suivi en exploitant sa géométrie. Le modèle 3D de l'objet (ou modèle CAO) permet de définir ses faces et arètes. En supposant que l'objet soit rigide, l'ensemble des points de l'image appartenant à cet objet bouge en fonction de leur position 3D sur le modèle et de la pose de la caméra. Le suivi consiste alors à estimer les 6 paramètres de déplacement et de rotation qui minimisent la distance entre la projection du modèle pour la pose estimée et les mesures dans l'image.

Une autre possibilité [Hartley 01, Shi 94, Hager 98] est de simplifier le suivi en supposant que l'élément suit un mouvement de déplacement générique dans l'image. Ce sont les approximations les plus simples en suivi basé image. Divers modèles existent en fonc-

tion du nombre de paramètres utilisés. Les modèles avec de nombreux paramètres sont plus robustes aux changements complexes mais nécessitent plus d'informations pour être estimés, tandis que les modèles simples peuvent suffire lorsque le déplacement est réduit. Considérons le déplacement d'un point de coordonnées ${}^a\mathbf{p}$ dans l'image de référence et de coordonnées ${}^b\mathbf{p}$ dans l'image courante. Les modèles les plus courammment utilisés sont :

- le modèle de translation : la région se déplace dans le plan image sans tourner. 2 paramètres (le vecteur $\mathbf{t}$) suffisent à représenter ce mouvement. Ce type de modèle est utilisé pour suivre un élément entre 2 images consécutives d'une même séquence vidéo. Le peu de mouvement effectué normalement entre ces deux images permet d'ignorer les autres déformations.

$$ {}^b\mathbf{p} = {}^a\mathbf{p} + \mathbf{t} \tag{3.1}$$

- le modèle de déplacement affine : En plus du déplacement sur le plan image est considéré la rotation 2D de la région et des déformations. Le déplacement est décrit par 6 paramètres (La matrice $\mathbf{A}$ et le vecteur de translation $\mathbf{t}$).

$$ {}^b\mathbf{p} = \mathbf{A}{}^a\mathbf{p} + \mathbf{t} \tag{3.2}$$

- le modèle de déplacement homographique : L'élément suivi est considéré être sur un plan qui se déplace suivant les 6 degrés de liberté (rotation + translation 3D). C'est le modèle le plus complexe, qui utilise 8 paramètres pour son fonctionnement (la matrice d'homographie $\mathbf{H}$ étant estimée à un facteur d'échelle près).

$$ {}^b\mathbf{p} = \mathbf{H}{}^a\mathbf{p} \tag{3.3}$$

Le suivi peut-être effectué directement à partir des informations d'intensité lumineuse ou de couleur extraites de l'image, ou être effectué sur les données calculées par un prétraitement de l'image. La solution retenue utilise l'intensité lumineuse de l'image pour effectuer le suivi d'une région de l'image.

3.3 Recherche des paramètres de déplacement

Le mouvement entre l'objet et la caméra entraine un déplacement de la région dans l'image. Tout point dont les coordonnées ${}^a\mathbf{p}$ sont définies dans la première image t_0 se retrouve à l'image t aux coordonnées :

$$ {}^b\mathbf{p} = f({}^a\mathbf{p}, \boldsymbol{\mu}) \tag{3.4}$$

où f est un modèle de déplacement dans l'image comme défini dans la section précédente.

L'intensité lumineuse d'un point à la date t_0 (l'image de référence $\mathcal{I}$) est donnée par $\mathcal{I}(\mathbf{x}, t_0)$. L'intensité lumineuse d'un point à la date t (l'image courante $\mathcal{I}'$) est donnée par $\mathcal{I}(f(\mathbf{x}, \boldsymbol{\mu}_t), t)$.

Soit $\mathcal{R}$ l'ensemble des n points de la région. L'hypothèse de ce type de suivi de région est que la luminance est conservée pour un même point entre deux images :

$$\mathcal{I}(f(\mathbf{x}, \boldsymbol{\mu}_t), t) - \mathcal{I}(\mathbf{x}, t_0) \quad = \quad 0, \forall \mathbf{x} \in \mathcal{R} \tag{3.5}$$

Le vecteur $\mathcal{I}(\boldsymbol{\mu}_0, t_0)$ contient l'intensité lumineuse des points à suivre dans l'image de référence. L'intensité de ces mêmes points à l'image t est donnée par :

$$\mathcal{I}(\boldsymbol{\mu}_t, t) \quad = \quad \mathcal{I}(f(\mathcal{R}, \boldsymbol{\mu}_t, t) \tag{3.6}$$

avec

$$f(\mathcal{R}, \boldsymbol{\mu}_t) \quad = \quad \left[\begin{array}{ccc} f(\mathbf{x}_0, \boldsymbol{\mu}_t) & \ldots & f(\mathbf{x}_n, \boldsymbol{\mu}_t) \end{array} \right]^{\mathtt{T}} \tag{3.7}$$

$$\tag{3.8}$$

L'objectif est d'estimer le vecteur $\boldsymbol{\mu}_t$ afin que l'équation (3.5), la conservation de la luminance, soit respectée entre l'image de référence et l'image courante. L'objectif est donc de déterminer le vecteur $\boldsymbol{\mu}_t$ qui minimise l'expression suivante :

$$\mathcal{O}(\boldsymbol{\mu}_{t+1}) \quad = \quad ||\mathcal{I}(\boldsymbol{\mu}_{t+1}, t+1) - \mathcal{I}(\boldsymbol{\mu}_0, t_0)|| \tag{3.9}$$

La continuité du mouvement nous assure que :

$$\boldsymbol{\mu}_{t+1} \quad = \quad \boldsymbol{\mu}_t \oplus \delta\boldsymbol{\mu} \tag{3.10}$$

où $\oplus$ est un opérateur de composition dépendant de la paramétrisation de $\boldsymbol{\mu}$. Le problème revient donc à déterminer le vecteur $\delta\boldsymbol{\mu}$. La relation 3.9 peut être réécrite par :

$$\mathcal{O}(\delta\boldsymbol{\mu}) \quad = \quad ||\mathcal{I}(\boldsymbol{\mu}_t \oplus \delta\boldsymbol{\mu}, t+1) - \mathcal{I}(\boldsymbol{\mu}_0, t_0)|| \tag{3.11}$$

Lorsque $\delta\boldsymbol{\mu}$ est suffisamment petit, la relation précédente peut être linéarisée en utilisant un développement de Taylor en $\boldsymbol{\mu}$ et en t. Soit *h.o.t.* les termes d'ordre supérieur négligés, la linéarisation de $\mathcal{I}(\boldsymbol{\mu}_t \oplus \delta\boldsymbol{\mu}, t+1)$ donne :

$$\mathcal{I}(\boldsymbol{\mu}_t \oplus \delta\boldsymbol{\mu}, t+1) \quad = \quad \mathcal{I}(\boldsymbol{\mu}_t, t) + \delta\boldsymbol{\mu}\frac{\partial \mathcal{I}(\boldsymbol{\mu}_t, t)}{\partial \boldsymbol{\mu}} + \frac{\partial \mathcal{I}(\boldsymbol{\mu}_t, t)}{\partial t} + h.o.t. \tag{3.12}$$

Considérons l'approximation :

$$\frac{\partial \mathcal{I}(\boldsymbol{\mu}_\mathbf{t}, \mathbf{t})}{\partial \mathbf{t}} \quad \approx \quad \mathcal{I}(\boldsymbol{\mu}_t, t+1) - \mathcal{I}(\boldsymbol{\mu}_t, t) \tag{3.13}$$

L'équation (3.11) peut alors être linéarisée suivant l'équation :

$$\mathcal{O}(\delta\boldsymbol{\mu}) \approx ||\mathcal{I}(\boldsymbol{\mu}_t, t) + \frac{\partial \mathcal{I}(\boldsymbol{\mu}_t, t)}{\partial \boldsymbol{\mu}}\delta\boldsymbol{\mu} + \frac{\partial \mathcal{I}(\boldsymbol{\mu}_t, t)}{\partial t} - \mathcal{I}(\boldsymbol{\mu}_0, t_0)|| \quad (3.14)$$

$$= ||\frac{\partial \mathcal{I}(\boldsymbol{\mu}_t, t)}{\partial \boldsymbol{\mu}}\delta\boldsymbol{\mu} + \mathcal{I}(\boldsymbol{\mu}_t, t+1) - \mathcal{I}(\boldsymbol{\mu}_0, t_0)|| \quad (3.15)$$

Ce terme $\mathcal{O}(\delta\boldsymbol{\mu})$ doit être minimisé pour atteindre l'objectif de superposition des régions courante et de référence. Cette minimisation est effectuée par une méthode de moindres carrés itérée avec :

$$\delta\boldsymbol{\mu} = \ominus \mathbf{J}^{+}(\mathcal{I}(\boldsymbol{\mu}_t, t+1) - \mathcal{I}(\boldsymbol{\mu}_0, t_0)) \quad (3.16)$$

où $\ominus$ est un opérateur de composition dépendant dépendant de la paramétrisation de $\boldsymbol{\mu}$ choisie et $\mathbf{J} = \frac{\partial \mathcal{I}(\boldsymbol{\mu}_t, t)}{\partial \boldsymbol{\mu}}$. La mise à jour du vecteur $\boldsymbol{\mu}$ s'effectue donc à chaque itération par :

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t \ominus \mathbf{J}^{+}(\mathcal{I}(\boldsymbol{\mu}_t, t+1) - \mathcal{I}(\boldsymbol{\mu}_0, t_0)) \quad (3.17)$$

Le processus de minimisation est répété jusqu'à ce que la différence $(\mathcal{I}(\boldsymbol{\mu}_t, t+1) - \mathcal{I}(\boldsymbol{\mu}_0, t_0)$ soit inférieure à un seuil.

3.4 Solution utilisée

Le seul *a priori* disponible sur l'environnement est qu'il est en partie constitué de surfaces planes. A condition de sélectionner une région dans l'image appartenant à un de ces plans (ce que l'on considérera comme acquis pour le moment et jusqu'au chapitre 4), la région subit une déformation suivant un modèle homographique. Un modèle de mouvement homographique peut donc être utilisé pour le suivi de la région.

$$H(\boldsymbol{\mu}) = \begin{bmatrix} \mu_1 & \mu_4 & \mu_7 \\ \mu_2 & \mu_5 & \mu_8 \\ \mu_3 & \mu_6 & 1 \end{bmatrix} \quad (3.18)$$

$$f(\mathbf{x}, \boldsymbol{\mu}_t) = w(H(\boldsymbol{\mu})\mathbf{x}) \quad (3.19)$$

où w est la fonction de projection perspective. L'équation (3.10) peut être réécrite en utilisant la multiplication matricielle comme opérateur de composition :

$$\boldsymbol{\mu}_t \oplus \delta\boldsymbol{\mu} = H(\boldsymbol{\mu}_t)G(\delta\boldsymbol{\mu})^{-1} \quad (3.20)$$

où G est une fonction qui construit une homographie en fonction de $\delta\boldsymbol{\mu}$. L'opérateur $\ominus$ définit dans l'équation (3.17) effectue alors la même multiplication matricielle en inversant le terme de droite :

$$\boldsymbol{\mu}_{t+1} = H(\boldsymbol{\mu}_t)G(\mathbf{J}^{+}(\mathcal{I}(\boldsymbol{\mu}_t, t+1) - \mathcal{I}(\boldsymbol{\mu}_0, t_0))) \quad (3.21)$$

Parmi les diverses possibilités, la solution proposée dans [Benhimane 07] est retenue. Dérivée des méthodes classiques itératives de suivi du type KLT [Shi 94], cette méthode propose une solution élégante et efficace de suivi utilisant un modèle homographique. Cette méthode propose une minimisation approximativement de second ordre qui ne nécessite pas de calculer la matrice hessienne. Elle permet donc une convergence rapide et ce avec un coût calculatoire réduit. Cette méthode sera utilisée dans notre application pour suivre les zones désignées.

Le paramètre de minimisation $\delta\boldsymbol{\mu}$ est un vecteur de taille 8. Ce vecteur n'a ici pas de sens propre si ce n'est qu'il sert à construire la matrice $\mathbf{A}$.

$$\mathbf{A} = \begin{bmatrix} \delta\mu_5 & \delta\mu_3 & \delta\mu_1 \\ \delta\mu_4 & -\delta\mu_5 - \delta\mu_6 & \delta\mu_2 \\ \delta\mu_7 & \delta\mu_8 & \delta\mu_6 \end{bmatrix} \tag{3.22}$$

Comme on peut le remarquer, la trace de cette matrice de taille 3 par 3 est nulle. C'est la caractéristique d'une matrice appartenant à l'algèbre $sl(3)$ (algèbre spécial linéaire). Le passage de l'algèbre au groupe correspondant $SL(3)$ se fait par l'exponentielle matricielle. Le groupe SL(3) étant le groupe des homographies, on peut construire à partir de A

$$G(\delta\boldsymbol{\mu}) = e^{\mathbf{A}} \tag{3.23}$$

L'avantage de cette représentation est qu'elle permet, entre autres, de représenter l'homographie $G(\delta\boldsymbol{\mu})$ par un vecteur $\delta\boldsymbol{\mu}$ qui n'a pas de propriété algébrique particulière et peut donc être manipulé plus aisément qu'une matrice d'homographie (qui doit conserver certaines propriétés).

La matrice jacobienne $\mathbf{J}$ est obtenue pour chaque point de la région (la matrice complète étant une agrégation verticale des jacobiennes pour chaque point) par l'équation suivante :

$$\mathbf{J} = \frac{1}{2}(\mathbf{J}_I + \mathbf{J}_{I'})\mathbf{J}_w\mathbf{J}_G \tag{3.24}$$

$\mathbf{J}_I$ et $\mathbf{J}_{I'}$ sont les gradients dans les images respectives $\mathcal{I}(\boldsymbol{\mu}_t, t+1)$ et $\mathcal{I}(\boldsymbol{\mu}_0, t_0)$. $\mathbf{J}_w$ est la jacobienne $\frac{\partial \mathbf{w}(\mathbf{G})}{\partial \mathbf{G}}$ quand $\mathbf{G} = \mathbf{I}_{3\times 3}$ et $\mathbf{J}_G$ est la jacobienne $\frac{\partial \mathbf{G}(\delta\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}$.

À chaque nouvelle image, la minimisation est effectuée par rapport à la même image de référence. Le processus itératif s'arrête si la norme de $\delta\boldsymbol{\mu}$ est trop petite et qu'il n'y a donc plus de modification dans l'homographie. Des vérifications sont faites pour s'assurer que l'estimation ne diverge pas. Dans le cas contraire, l'estimation est arrêtée

et le suivi considéré comme invalide pour cette image (il ne sera pas utilisé comme mesure). Si l'estimation diverge pendant plusieurs images, la zone est retirée de la liste des régions à suivre. Deux critères sont utilisés pour vérifier que le tracker ne diverge pas. Le premier est simplement une vérification photométrique. Si la différence de luminosité augmente subitement, on considère que le suivi n'a pas fonctionné. Le second critère vérifie que la région ne subit pas une distortion géométrique trop importante par rapport à l'image précédente. Cette conservation des proportions de la région est obligatoire en raison de la faible paralaxe entre les deux prises de vues.

3.5 Améliorations du suivi basé image

3.5.1 Luminosité et robustesse

Notons que ce type de suivi est très sensible aux changements de luminosité puisqu'il repose sur le principe de conservation de la luminance pour les points d'une région de l'image. Ce postulat est évidemment rarement respecté dans des environnements réels où des ombres peuvent apparaitre, les sources lumineuses être modifiées, etc. Pour compenser partiellement les éventuels changements de luminosité, une solution peu coûteuse consiste à prendre en compte uniquement les changements qui affectent toute la région de manière uniforme. Cette solution, si elle ne permet pas de compenser les changements dus aux spécularités, permet toutefois de rendre le suivi beaucoup plus robuste dans la majorité des cas. Une solution consiste à égaliser les histogrammes des deux régions.

Pour compenser ce changement global d'illumination, la région de l'image courante est transformée par la formule suivante :

$$\alpha = \sqrt{\frac{\frac{1}{n}\sum_{k=1}^{n}\mathcal{I}(\mathbf{p}_k)^2}{\frac{1}{n}\sum_{k=1}^{n}\mathcal{I}'(\mathbf{p}'_k)^2}} \tag{3.25}$$

$$\beta = \frac{1}{n}\sum_{k=1}^{n}\mathcal{I}(\mathbf{p}_k) - \alpha * (\frac{1}{n}\sum_{k=1}^{n}\mathcal{I}'(\mathbf{p}'_k)) \tag{3.26}$$

$$\mathcal{I}'(\mathbf{p}'_i) = \alpha\mathcal{I}'(\mathbf{p}'_i) + \beta \tag{3.27}$$

L'illustration 3.1 montre le gain apporté par cette transformation. L'image de gauche représente le patch courant. L'image centrale représente la différence d'intensité entre le patch de référence et le patch courant non corrigé. L'image de droite représente la différence avec le patch courant corrigé (une différence plus grande est représentée par un pixel plus clair). Une diminution de la différence d'intensité lumineuse est observable lorsque le changement global d'illumination est pris en compte.

3.5.2 Minimisation complexe et réduction de l'espace de recherche

La méthode de suivi présentée permet de suivre des régions dans l'image de manière efficace même dans des cas difficiles de par la technique de minimisation différentielle

FIGURE 3.1 – Illustration de la correction du changement d'illumination. Le patch de gauche représente l'image courante. Les deux patchs de droite représentent la différence d'intensité lumineuse entre le patch de référence et le patch courant successivement sans et avec correction du changement global d'illumination.

qui lui permet d'éviter de parcourir l'ensemble de l'espace de recherche. Cependant, certains mouvements brusques dans l'image, de légers flous ou autres changements locaux de lumière rendent souvent plus ardue le suivi. Dans ces cas particuliers, l'espace de recherche est en effet élargi et la fonction de coût est bruitée, provoquant des pertes du suivi. Il est donc nécessaire d'essayer de palier ces problèmes par diverses méthodes.

Création d'une pyramide d'image La première solution est largement utilisée et est très populaire. Il s'agit d'utiliser le principe de la pyramide Gaussienne. Une pyramide d'image (figure 3.2) est, comme son nom l'indique, une série d'images représentant le même contenu à différentes résolutions. Chaque image est un étage de cette pyramide. La variation de la résolution entre chaque étage dépend des paramètres de cette pyramide.

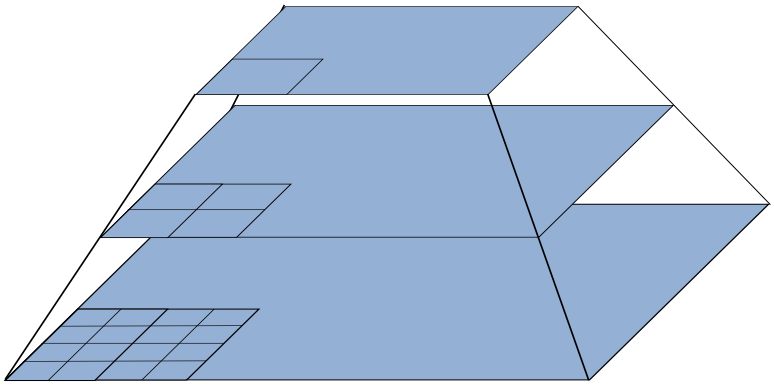


FIGURE 3.2 – Le concept de pyramide d'image

Un filtre Gaussien est une technique de traitement de signal permettant de lisser un signal. Le signal est convolué par un masque Gaussien. Comme toute méthode du traitement de signal, elle peut être appliquée à une image qui est un signal 2D. La Gaussienne utilisée est alors évidemment une Gaussienne 2D. L'objectif est alors de fusionner les pixels proches pour enlever les bruits de haute fréquence et faire du flou

(figure 3.3).

FIGURE 3.3 – Une pyramide gaussienne de 3 étages appliquée à une image

Le concept de pyramide Gaussienne est évidemment un concept hybride qui reprend les deux méthodes précédemment décrites. La pyramide est construite en sous échantillonnant l'image de l'étage précédent. Le caractère discret de l'image induit une modification du signal lors du sous échantillonnage et rajoute notamment du bruit de haute fréquence dû à la diminution de la résolution du signal. Le nouvel étage ainsi créé est lissé en appliquant un filtre Gaussien pour compenser ce bruit artificiel. Ainsi, chaque étage représente la même information à des degrés différents de précision ou de complexité. Il peut être alors compris que l'espace de recherche pour notre suivi est d'autant réduit que l'étage est haut dans la pyramide.

La matrice d'homographie retournée à chaque itération par le tracker représente l'homographie **en pixels** estimée entre l'image de référence et l'image courante pour la zone considérée. Elle dépend donc de la résolution de l'image et donc de l'étage de la pyramide Gaussienne considéré. En commençant par l'étage le plus haut et pour chaque étage, l'homographie est transformée pour correspondre à cette résolution, le tracking est effectué et, si aucune erreur n'est apparue, l'homographie résultante est utilisée pour initialiser le suivi à l'étage inférieur. La transformation de la matrice d'homographie se fait à l'aide de l'équation :

$$\mathbf{L} = \begin{bmatrix} ratio & 0 & 0 \\ 0 & ratio & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.28}$$

$$\mathbf{H}_{level} = \mathbf{L}\mathbf{H}\mathbf{L}^{-1} \tag{3.29}$$

où *ratio* est le coefficient de sous-échantillonnage utilisé pour l'étage courant.

Le suivi est effectué sur un espace de recherche simplifié, ce qui permet de faciliter la convergence et d'éviter certains minimums locaux induits par les hautes fréquences.

Au fur et à mesure que l'on descend dans la pyramide, le suivi permet d'affiner l'homographie en profitant de l'approximation précédente. La complexité calculatoire du suivi étant fonction du nombre de pixels de la région, effectuer les premières itérations sur les premiers étages de la pyramide permet d'accélérer le processus et de le rendre plus robuste à des mouvements importants dans l'image.

Prédiction de l'initialisation. Comme tout processus de minimisation itératif de type Newton, l'estimation des paramètres peut être mise à mal par des mouvements brutaux de la caméra, l'estimation initiale (l'homographie à l'image précédente image) étant trop éloignée du minimum global. Comme toute méthode de minimisation non linéaire, cette condition rend le fonctionnement moins évident. Dans ce cas, une aide extérieure, un *a priori* sur l'homographie, donnée au tracker permet de faciliter la convergence.

Rappelons que notre suivi est utilisé dans le cadre plus global d'un estimateur statistique de pose. Basé sur un filtre de Kalman étendu, cet estimateur nous fournit une prédiction de la mesure. Dans le cas où l'incertitude –la variance– de cette prédiction n'est pas trop forte, cette homographie prédite est utilisée comme initialisation du processus itératif à chaque nouvelle image.

Les capteurs inertiels que nous présenterons dans un chapitre ultérieur permettent d'affiner cette prédiction de la mesure image et ainsi de permettre d'aider le suivi à chaque image encore plus efficacement.

3.5.3 Occlusions et gestion des informations erronées

Cette méthode de suivi nécessite que la région suivie conserve une cohérence sur l'information lumineuse mesurée au fur et à mesure du temps. Elle suppose en effet que la région d'origine puisse être retrouvée entièrement dans la nouvelle image simplement en effectuant une transformation homographique des points de la région (équation 3.5). Parfois, certaines parties de la région ne respectent plus cette équation, de par un changement local d'intensité lumineuse (spécularité) ou par un objet étranger venant occulter cette partie de la région (on parle d'occlusion). La présence de ces informations erronées (outliers) entraine des erreurs dans le processus d'estimation (moindre carré itéré). La problématique est donc de mettre de côté ces pixels afin de ne prendre en compte que les pixels qui respectent le modèle considéré.

l'équation (3.15) revient à la problématique classique de minimisation d'une erreur $\mathcal{O}(\delta\boldsymbol{\mu})$ en fonction des paramètres $\delta\boldsymbol{\mu}$:

$$\mathcal{O}(\delta\boldsymbol{\mu}) \quad = \quad \mathbf{s}(\delta\boldsymbol{\mu}) - \mathbf{s}^* \tag{3.30}$$

où $\mathbf{s}^*$ est la mesure et $\mathbf{s}(\delta\boldsymbol{\mu})$ une fonction d'estimation de la mesure.

Lorsque $\mathcal{O}(\delta\boldsymbol{\mu})$ est partiellemment erroné, la littérature propose, entre autres, l'utilisation d'une classe de méthodes : les M-Estimateurs. Ceux-ci avancent que chaque ligne du système n'a pas la même force de preuve que les autres et que l'information doit être pondérée afin de refléter cet état de fait. Cette pondération peut s'exprimer sous la forme d'une matrice diagonale contenant les poids de chaque ligne 1 à n.

$$\mathbf{W} = \begin{bmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_n \end{bmatrix} \tag{3.31}$$

L'équation (3.30) est pondérée par cette matrice diagonale :

$$\mathcal{O}(\delta\boldsymbol{\mu}) = \mathbf{W}(\mathbf{s}(\delta\boldsymbol{\mu}) - \mathbf{s}^*) \tag{3.32}$$

L'équation à minimiser, linéarisée au premier ordre (3.15), est modifiée pour refléter cette pondération :

$$||\mathbf{W}\mathcal{O}(\boldsymbol{\mu}_t) + \mathbf{W}\mathbf{J}\delta\boldsymbol{\mu}|| \tag{3.33}$$

et les paramètres $\delta\boldsymbol{\mu}$ sont alors obtenus par :

$$\delta\boldsymbol{\mu} = (\mathbf{W}\mathbf{J})^{+}\mathbf{W}\mathcal{O}(\boldsymbol{\mu}_t) \tag{3.34}$$

Le principe des M-estimateurs repose dans le calcul de ces poids. Ce calcul doit permettre de rendre la minimisation robuste aux données aberrantes. Les M-estimateurs considèrent que les données aberrantes sont celles qui s'écartent trop de la médiane des résidus $\mathbf{e}(\delta x)$. La médiane des résidus est calculée ainsi que sa variance. D'une manière générale, les valeurs les plus proches de la valeur médiane auront un poids plus fort. Différents M-estimateurs existent en fonction de la loi utilisée pour attribuer les poids. Dans notre cas nous avons retenu l'estimateur de Tukey. Les poids sont donnés par :

$$\sigma = median(\mathbf{b} - median(\mathbf{b})) * 1.4826 \tag{3.35}$$

$$w_i = \begin{cases} (1 - (\frac{\frac{b_i}{\sigma}}{4.6851})^2)^2 & \text{si } \frac{b_i}{\sigma} <= 4.6851 \\ 0 & \text{sinon} \end{cases} \tag{3.36}$$

On remarque qu'au-delà d'une certaine valeur du résidu, le poids est mis à 0, ce qui signifie qu'on ignore tout simplement cette information.

Ramenée à notre problématique, l'utilisation des M-estimators revient à ignorer les pixels dont la différence d'intensité est trop forte par rapport à celles de la majorité des pixels ou pour le moins à les prendre moins en compte.

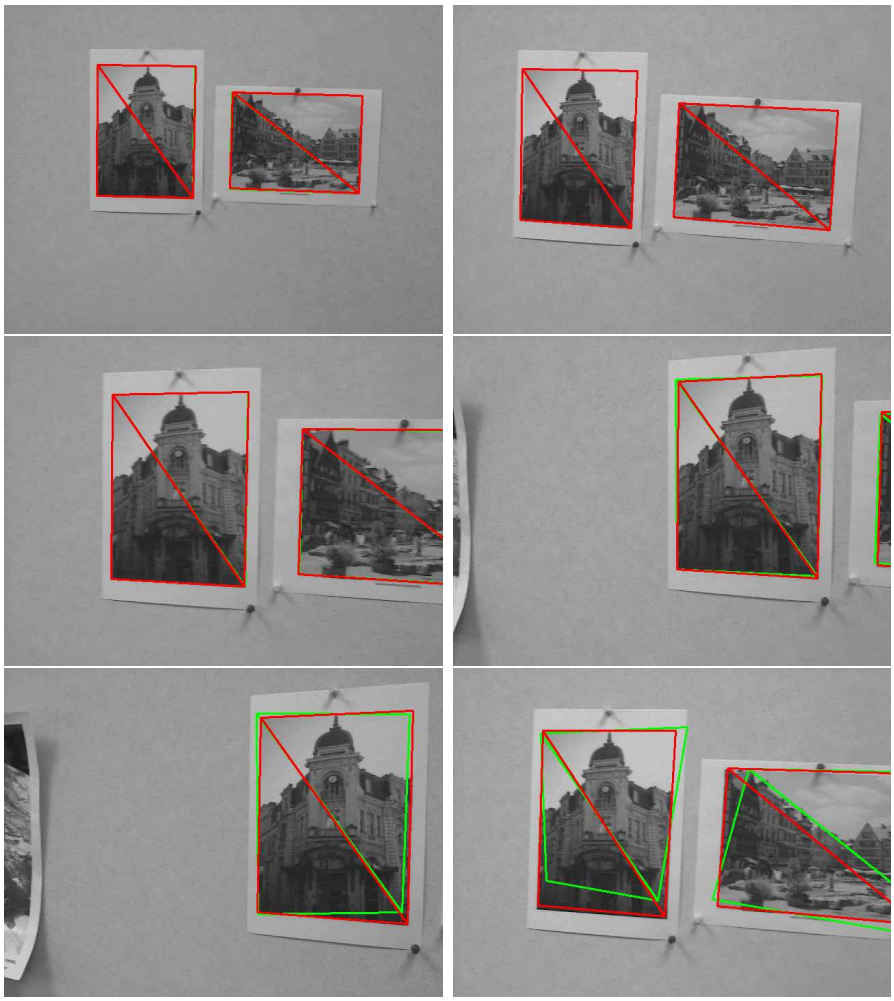


FIGURE 3.4 – Premier exemple d'occlusion avec le suivi Les contours des régions suivies sont affichés en vert et rouge, respectivement sans et avec l'aide des M-Estimateurs. Un meilleur suivi est observable avec les M-estimateurs.

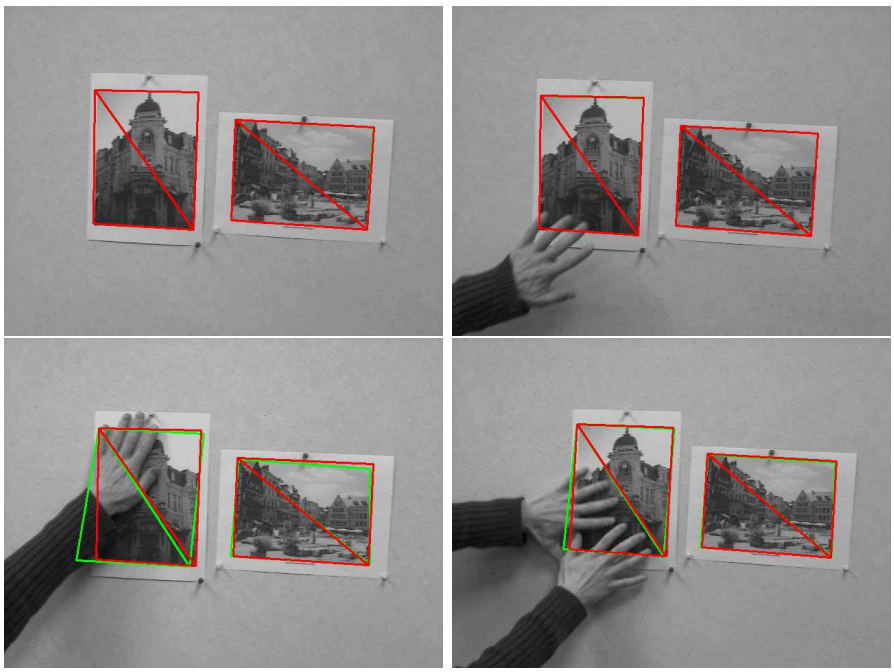


FIGURE 3.5 – Second exemple d'occlusion avec le suivi. Les contours des régions suivies sont affichés en vert et rouge, respectivement sans et avec l'aide des M-Estimateurs. Un meilleur suivi est observable avec les M-estimateurs.

Les figures 3.4 et 3.5 illustrent les résultats obtenus avec le système de suivi. Les lignes délimitent la zone que le suivi estime être la région suivie. En vert la zone estimée par le suivi standard, en rouge la zone estimée à l'aide des M-Estimateurs. Les deux quadrilatères appartiennent au même tracker et participent donc au même processus de minimisation.

L'inconvénient majeur de cette méthode est que le calcul des poids nécessite le calcul de la médiane du vecteur résidu et donc le tri de ce dernier. Ce calcul alourdit considérablement le coût calculatoire. De plus, on observe un ralentissement de la convergence du processus d'estimation lorsqu'on utilise les M-Estimators, dans certains cas, qui découle du lissage des informations. L'utilisation de cette méthode robuste est donc optionnelle et peut être remplacée par notre capacité à remplacer les régions perdues au fur et à mesure à l'aide du SLAM.

3.5.4 Patches multiples et instabilité numérique

Il peut arriver que plusieurs zones distinctes de l'image soient choisies dans une même image et qu'elles appartiennent au même plan. Dans ce cas, il parait logique de vouloir agréger les informations contenues dans les différentes zones pour améliorer la qualité de la minimisation en augmentant le support d'estimation. En effet, plus le nombre de pixels est important, plus le système va être contraint et le suivi facilité. L'article de référence [Benhimane 07] ne démontre que l'utilisation d'une seule région et les coordonnées des pixels sont données localement par rapport au coin haut gauche de la région, les coordonnées du coin étant définies dans l'homographie de départ. Lorsque plusieurs régions sont définies, les coordonnées des points doivent être définies dans le repère de l'image, ce qui provoque de grandes disparités dans les valeurs de la Jacobienne.

La jacobienne (3.24) – qui définit le comportement de la minimisation – peut être exprimée après développement par le vecteur suivant

$$\mathcal{I}_x = \frac{\partial \mathbf{I}(\mathbf{x},\mathbf{y})}{\partial \mathbf{x}} + \frac{\partial \mathbf{I}'(\mathbf{x},\mathbf{y})}{\partial \mathbf{x}} \tag{3.37}$$

$$\mathcal{I}_y = \frac{\partial \mathbf{I}(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}} + \frac{\partial \mathbf{I}'(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}} \tag{3.38}$$

$$\mathbf{J}_i = \left[\begin{array}{cccccc} \mathcal{I}_x & \mathcal{I}_y & \mathcal{I}_x y & \mathcal{I}_y x & \mathcal{I}_x x - \mathcal{I}_y y & -2\mathcal{I}_y y - \mathcal{I}_x x \quad -\mathcal{I}_x x^2 - \mathcal{I}_y xy \end{array} \right] \tag{3.39}$$

Dans le cas où les points sont définis dans le repère de l'image, une divergence de la minimisation a lieu dès les premières itérations.

Malgré un nombre de lignes infiniment supérieur au nombre de colonnes $\mathbf{J}^{+}\mathbf{J} \neq \mathbf{I}$. La décomposition en valeurs singulières de $\mathbf{J}^{\mathsf{T}}\mathbf{J}$ dans les expérimentations donne à chaque fois une très grande différence entre la plus petite et la plus grande valeur. La matrice $\mathbf{J}^{\mathsf{T}}\mathbf{J}$ n'est donc pas de rang plein et n'est donc pas inversible numériquement, ce qui explique la divergence de la minimisation.

Le fait que la jacobienne ne soit pas de rang plein semble s'expliquer par les très grandes différences de valeurs trouvées dans les différentes colonnes lorsqu'on utilise des valeurs pour x et y qui sont importantes. La solution utilisée, qui prouve son fonctionnement dans la pratique de nos expérimentations, est de normaliser chaque colonne et donc de préconditionner la matrice $\mathbf{J}$ avant l'inversion. Ce préconditionnement se fait à travers une matrice carrée diagonale $\mathbf{P}$. L'équation définissant la pseudo-inverse de Moore-Penrose est donc localement redéfinie pour prendre en compte ce préconditionnement.

$$\mathbf{J}^{+} = \mathbf{P}((\mathbf{PJ})^{\mathsf{T}}(\mathbf{PJ}))^{-1}\mathbf{J}^{\mathsf{T}} \tag{3.40}$$

$\mathbf{P}$ est construite de manière à ce que les valeurs de chaque colonne se situent dans les mêmes intervalles pour faciliter l'inversion numérique. m_p est la valeur maximale que peut prendre une coordonnée (le maximum entre la largeur de l'image et la hauteur), m_s est la valeur maximale que peux prendre un gradient d'image.

$$\mathbf{P} = \begin{bmatrix} m_s & & & & & & & & \\ & m_s & & & & & & & \\ & & m_s m_p & & & & & & \\ & & & m_s m_p & & & & & \\ & & & & m_s m_p & & & & \\ & & & & & -m_s m_p & & & \\ & & & & & & m_s m_p^2 & & \\ & & & & & & & m_s m_p^2 & \end{bmatrix}^{-1} \tag{3.41}$$

$(\mathbf{PJ})^{\mathsf{T}}(\mathbf{PJ})$ est alors de rang plein et la minimisation s'effectue normalement. L'utilisation de multiples régions pour le suivi est alors possible.

3.5.5 Simplification du modèle

L'objectif étant de suivre des régions de l'image appartenant à un plan dans l'espace, le choix du modèle homographique est optimal dans le sens où il permet de décrire l'ensemble des transformations possibles de ce plan. L'inconvénient est qu'il nécessite l'estimation de huit paramètres. Les informations lumineuses provenant des images étant fortement bruitées, la minimisation de ce système à huit inconnues rend nécessaire l'utilisation d'un grand nombre de pixels et donc d'une région de l'image suffisamment grande. Afin de réduire le nombre de pixels nécessaire et donc la taille de la zone à suivre, mais également afin de faciliter dans certains cas le suivi lors de mouvements brusques, il est possible d'utiliser un modèle plus simple pour les premières itérations afin d'initialiser les paramètres à estimer. Réduire la taille minimale d'une zone à suivre permet de sélectionner ces zones plus facilement (cf. section 4.4).

Nos expériences montrent que dans cet objectif un modèle de translation est souvent suffisant. Il s'agit de trouver une première estimation du déplacement depuis l'image précédente. Avec seulement deux paramètres, le calcul de la Jacobienne est beaucoup plus rapide qu'avec un modèle plus complexe comme un modèle affine. Ces deux paramètres sont les paramètres de déplacement de la zone dans le plan image tx et ty. Dans ce cas, la mise à jour de l'homographie se fait plus simplement par

$$\mathbf{G} = \mathbf{G} \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \tag{3.42}$$

et la jacobienne $\mathbf{J}$ permettant la minimisation est simplement l'accumulation des lignes

$$\mathbf{J}_i = \begin{bmatrix} \mathcal{I}_x & \mathcal{I}_y \end{bmatrix} \tag{3.43}$$

En utilisant ce modèle pendant les premières itérations, on obtient les améliorations attendues dans de nombreux cas. Il est également possible d'effectuer une sélection de modèle en choisissant le modèle qui minimise le mieux la différence d'intensité, mais les tests ne montrent pas de meilleurs résultats par rapport à l'utilisation automatique du modèle de translation pendant les premières itérations.

3.5.6 Optimisations

Le suivi est de loin le traitement le plus lourd dans notre pipeline de calcul de pose. L'optimisation logicielle de cette partie est donc primordiale pour le bon fonctionnement en temps réel du processus. De multiples optimisations ont été effectuées sur le code, les outils de compilation et les choix des librairies de calculs que l'on ne détaillera pas.

Notons cependant qu'un travail important a été fait sur la vectorisation des calculs à travers l'utilisation des capacités Single Instruction Multiple Data (SIMD) des processeurs modernes. Ceux-ci permettent – sur les processeurs type Intel IA-32 ou IA-64 – d'effectuer la même opération sur 4 variables flottantes à simple précision. Cette technologie SSE (version 3) nécessite l'écriture de codes propriétaires mais permet de multiplier par 3 ou 4 la rapidité d'exécution des fonctions concernées.

Le remplissage de la Jacobienne est une des parties importantes du processus. Une optimisation intéressante peut être obtenue en choisissant de créer la transposée Jacobienne directement. D'une part, la pseudo-inverse utilise 2 fois la transposée et 1 seule fois la Jacobienne normale. Cette dernière est obtenue alors en transposant la 'transposée'. Puisque tous les mêmes éléments d'une même colonne sont remplis de la même manière et accédés séquentiellements, l'utilisation de la transposée permet de transformer les colonnes en lignes et donc de juxtaposer les éléments en mémoire. L'accès mémoire est alors plus rapide et l'utilisation des SSE facilitée.

FIGURE 3.6 – Exemple de suivi d'un tableau dans le musée de Rennes

3.6 Conclusion

Ce chapitre montrait une partie du pipeline de calcul de pose de notre application. Basé sur une méthode proposée par [Benhimane 07], ce suivi est bien adapté à notre problématique puisqu'il permet de suivre des régions de l'image appartenant à un plan et fournit une estimation de l'homographie en sortie. Cette dernière peut alors être directement utilisée comme flux d'entrée dans notre estimateur détaillé dans le chapitre 2.7.

Après avoir décrit le principe de la méthode existante, ce chapitre détaillait les améliorations apportées. Des améliorations permettant de faciliter le suivi ont été proposées comme l'utilisation des pyramides Gaussiennes, la compensation des changements globaux d'illumination ou l'utilisation de la prédiction de la mesure fournie par le SLAM. L'introduction de modèles de mouvements plus simples pendant les premières itérations permet d'obtenir de meilleurs résultats lorsque les régions suivies ne sont pas suffisamment grandes. La robustesse de la minimisation a été également améliorée par l'utilisation de M-Estimators. Enfin, ce chapitre a montré comment permettre l'utilisation de multiples régions de l'image dans le même suivi à l'aide d'un préconditionnement de la Jacobienne utilisée pour la minimisation. Des optimisations logicielles ont également été effectuées.

Dans les conditions "extrêmes" comme des mouvements brusques provoquant des flous de bougés ou des déplacements dans l'image trop importants, le suivi ne peut naturellement pas fonctionner sans aide extérieure. Le chapitre 5 expliquera comment l'utilisation des capteurs inertiels peuvent aider à compenser les pertes du suivi, voire de les empêcher grâce à la prédiction.

Après le traitement d'image, estimation de la pose, le chaînon manquant pour un fonctionnement automatique est la sélection des zones de l'image à suivre. Suivant l'objectif de l'application et les informations connues *a priori* sur l'environnement, diverses solutions seront présentées dans le chapitre suivant (chapitre 4).

Chapitre 4

Détection et intégration de nouveaux plans

Les chapitres précédents ont montré la faisabilité et l'intérêt d'un système de SLAM qui se sert des plans comme mesure. Un estimateur a été décrit ainsi que la méthode de suivi d'homographies permettant de faire l'interface entre les images et le coeur de l'estimateur. Le suivi – et indirectement l'estimateur – considèrent que les zones suivies dans l'image sont planes. Afin d'assurer un bon fonctionnement de la méthode, il est impératif d'éviter les zones de l'image non planaires. Au lancement de l'application, ou à cause du déplacement de la caméra, les zones suivies peuvent devenir insuffisantes voire inexistantes. Pour permettre à l'estimation de la pose de continuer à fonctionner, il est nécessaire de trouver de nouvelles zones à suivre dans les images. Ce chapitre s'intéresse aux méthodes permettant d'atteindre cet objectif. Suivant le contexte, différentes solutions sont proposées.

La première problématique est d'extraire de l'image les zones planaires. La première section de ce chapitre décrit une méthode permettant de reconnaitre dans l'image des zones qui ressemblent à des éléments d'une base de données. La base de données d'images ne contenant que des régions planes, le fait de reconnaître dans l'image une de ces régions est le gage de sélectionner une zone plane. Cependant, ces bases de données ne sont pas disponibles pour tous les environnements et/ou la vue courante ne peut contenir aucun élément connu. Dans le cas où la reconnaissance n'est pas fonctionnelle, il est nécessaire d'analyser le mouvement de l'image pour détecter les zones planes. La dernière section de ce chapitre démontre qu'il est possible d'effectuer cette analyse tout en continuant d'estimer la pose.

La seconde problématique est d'initialiser dans l'estimateur ces nouvelles régions. Il est en effet nécessaire de savoir quels paramètres initiaux spécifier pour le plan de la nouvelle zone. Si la zone a été reconnue par rapport à une base de données, il est possible qu'une approximation du plan soit fournie. Dans le cas contraire, la deuxième section expliquera les problématiques de l'initialisation sans *a priori* et la solution apportée

par cette thèse.

4.1 Reconnaissance de régions

Le système est supposé disposer d'une source décrivant un ensemble d'images représentant des objets plans existants dans l'environnement, qu'ils proviennent de photographies ou de rendus procéduraux codant une information (comme un marqueur). Considérons une image de la scène qui contient (ou pas) un élément de cette base de données. Cet élément peut-être, pour toutes sortes de raisons, déformé, partiellement caché, éclairé différemment, etc.. À partir de cette image, l'objectif est de déterminer si un élément de la base de données se trouve dans l'image. L'objectif est aussi de déterminer la transformation géométrique entre la représentation de la base de données et la vue actuelle pour pouvoir le localiser. Puisqu'on s'intéresse à des structures planes, il s'agit donc de trouver l'homographie entre les deux projections de l'image.

FIGURE 4.1 – Reconnaissance de tableaux dans le musée des beaux-arts de Rennes

L'objectif est donc bien de reconnaitre un motif précis dans l'image et non une classe d'objets comme dans les problématiques de reconnaissance. De multiples options sont offertes par la littérature, utilisant majoritairement des outils provenant des statistiques, du traitement d'image et de la vision par ordinateur. Deux classes de solutions s'imposent par leur popularité.

4.1.1 Méthodes de reconnaissance par corrélation

Le principe le plus "simple" est de considérer les motifs directement et de chercher dans l'image un motif similaire par comparaison. Considérons une métrique de similarité entre deux motifs, dépendant du type de motif recherché et de la précision voulue. Le motif est considéré comme identique si la métrique renvoie une distance inférieure à une constante.

La comparaison de deux motifs pose la problématique de la déformation du motif éventuellement présent dans l'image. Le motif, s'il est présent dans l'image, peut être déformé suivant toutes les configurations permises par les transformations homographiques. Ce nombre important de solutions doit être testé pour effectuer la reconnaissance. Cette grande dimension de l'espace de recherche rend quasi impossible la

réalisation de la reconnaissance par corrélation sans a priori limitatif sur le motif et son environnement. Une première solution – utilisée en première passe optionnelle pour la reconnaissance de tableaux dans notre application – est de considérer que le motif recherché est entouré d'une région constituée d'un aplat de couleurs et donc dépourvue de gradient. Par une technique de seuillage, on peut retrouver la zone occupée par le motif et, en supposant un *a priori* sur la forme de l'objet recherché, dédistordre ce dernier pour calculer la similarité.

Une seconde solution est de restreindre le type de motif utilisé au lieu de considérer une simplification de l'environnement. L'idée est de modifier le motif de manière à ce qu'il fournisse un indice utilisable quelque soit le point de vue. La reconnaissance de marqueurs (voir figure 4.2) par les logiciels dédiés tels que ARToolKit ou autres fonctionne selon ce principe : le motif est entouré d'une zone de couleur unie. Toujours par seuillage des gradients, la région du motif est détectée en cherchant un contour fermé. Cette méthode est extrêmement rapide et permet de détecter des motifs facilement et de manière très robuste.

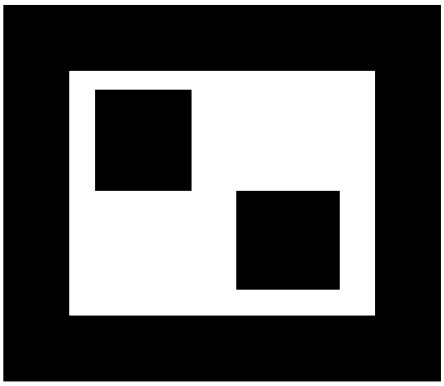


FIGURE 4.2 – Un marqueur détectable par ARToolKit

Ces techniques ont cependant leurs limites incompressibles. La première est qu'elle nécessite des motifs/environnement avec certaines caractéristiques peu fréquentes dans des environnements quelconques. La seconde est que ces techniques sont extrêmement sensibles aux occultations. La moindre occultation provoque une erreur dans l'extraction des contours et rend donc impossible la reconnaissance. Cette méthode est également fortement sensible aux changements locaux du motif. En effet, la mesure de similarité est effectuée sur l'ensemble du motif : une illumination spéculaire par exemple peut provoquer une erreur de reconnaissance.

4.1.2 Méthodes de reconnaissance par primitives

Cette seconde classe de méthodes repose sur l'utilisation d'un faisceau de preuves permettant de reconnaitre notre motif plutôt qu'une comparaison brute et globale.

Chaque élément probant est insuffisant en lui-même, étant trop local et donc peu discriminant. La reconnaissance se base sur la quantité d'éléments probants. Cela permet une plus grande souplesse sur la reconnaissance individuelle et globalement une plus grande tolérance aux perturbations non anticipées. Ces primitives utilisées comme information locale sont le plus souvent des points particuliers (keypoint).

Le premier principe est de sélectionner des points que l'on peut espérer retrouver dans une autre image, donc qui contiennent suffisamment d'informations dans leur voisinage local, et ce malgré des points de vue différents. La sélection des points se fait en fonction de la nature mathématique des gradients avoisinant le point. Le principe de sélection diffère suivant la méthode choisie. Le principe général est de choisir des points dont le voisinage possède suffisamment d'informations tout en privilégiant des régions dont les gradients ne sont pas dans une seule direction principale. Ce dernier critère permet d'éviter les bords, les lignes dans l'image qui contiennent une multitude de points similaires et donc peu discriminables. Pour résumer, la sélection recherche des points saillants et donc faciles à retrouver.

Cette sélection peut se faire, dans le cas de point de Harris, par le calcul du ratio des 2 valeurs propres de la matrice d'auto-corrélation du signal. Dans d'autres méthodes, on utilisera la Hessienne de cette même matrice. Le déterminant ou plus simplement la trace de cette Hessienne est alors considéré comme la métrique de qualité du point dans l'optique d'une reconnaissance après une déformation géométrique. Ces calculs sont effectués sur toute l'image, fournissant une image "réponse" (figure 4.3). Les pics de cette réponse permettent de sélectionner les points.

FIGURE 4.3 – Une image et la réponse de Harris associée

Le second principe est de trouver un moyen de comparer deux points provenant de deux images différentes. Il faut trouver un espace qui permette de décrire le point d'une manière aussi concise et complète que possible le point et ses caractéristiques principales. Cet espace doit être accompagné d'une métrique établissant la distance entre deux éléments dans cet espace. La méthode SIFT (détaillée dans l'encadré 4.5) propose

par exemple de représenter un point par un descripteur dans un espace à 128 dimensions (représentant 16 histogrammes d'orientation des gradients) permettant d'obtenir une représentation valide malgré une rotation dans le plan image ou un changement d'échelle. L'espace est supposé euclidien ; la métrique est donc une simple distance euclidienne. Soit $\mathbf{e}_1$ et $\mathbf{e}_2$ deux descripteurs de points, la distance entre ces deux points est

$$distance(\mathbf{e}_1, \mathbf{e}_2) = \sqrt{\sum_{i=1}^{128} (\mathbf{e}_{1_i} - \mathbf{e}_{2_i})^2} \tag{4.1}$$

4.1.3 Recherche du plus proche voisin

Considérons un espace $\mathbf{C}$ à m dimensions doté d'une fonction permettant de mesurer la distance entre deux éléments définis dans cet espace. Cette fonction de distance permet de classer les éléments les uns vis à vis des autres. Considérons un ensemble $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_n\}$ d'élements appartenant à $\mathbf{C}$. Considérons un élément $\mathbf{g} \in \mathbf{C}, \notin \mathbf{E}$. La recherche du plus proche voisin consiste à trouver l'élément de $\mathbf{E}$ dont la distance avec $\mathbf{g}$ est minimale. Naïvement, le procédé est simpliste puisqu'il consiste à calculer la distance avec chaque élément de $\mathbf{E}$ en conservant l'élément de distance minimale. La complexité est donc en $O(n)$. Qu'advient-il si n est grand ? La recherche du plus proche voisin est alors impossible en temps réel. Il est donc nécessaire de trouver une méthode permettant de réduire cette complexité si l'on souhaite traiter des cas avec une valeur de n importante. La solution proposée nécessite que $\mathbf{E}$ soit connu a priori et que seule la recherche du plus proche voisin soit à effectuer en ligne.

Une solution possible, à la fois simple et adaptée à notre problématique applicative, est la méthode des k-means hiérarchique. C'est une méthode approximative puisqu'elle peut ne pas retourner le vrai plus proche voisin et des raffinements existent dans la littérature, mais les conditions de fonctionnement de notre application nous permettent de nous contenter de ces capacités. En effet, la probabilité qu'un faux plus proche voisin soit renvoyé est suffisamment faible, par rapport au nombre total de points mis en correspondance, pour être éliminé par les méthodes robustes de validation géométrique.

4.1.3.1 La méthode des k-means

La méthode des k-means est la méthode de partitionnement de données la plus usitée et la plus référencée dans la littérature. Elle consiste à regrouper un ensemble de données en k partitions (partition étant utilisé dans sa définition ensembliste). À chaque partition $\mathbf{P}_i$ est associé un centroïde $\mathbf{c}_i$. Les partitions sont créées de manière à minimiser la "variance intra-cluster", c'est à dire la somme

$$S = \sum_{i=0}^{k} \sum_{j=0, \mathbf{e}_j \in \mathbf{P}_i}^{n} distance(\mathbf{c}_i, \mathbf{e}_j)^2 \tag{4.2}$$

Sa construction se fait par un algorithme itératif qui répète les 2 opérations
- pour tout élément de $\mathbf{E}$, l'associer au $\mathbf{P}_i$ dont le $\mathbf{c}_i$ est le plus proche
- recalculer les centroïdes $\mathbf{c}_i$ de tous les $\mathbf{P}_i$

Cette phase d'apprentissage effectuée, le résultat peut-être utilisé pour apparier de nouvelles données à une partition sans modifier ces dernières. Comme toute méthode basée sur une phase d'apprentissage, plus le jeu de données utilisé est important, mieux le partitionnement sera effectué, mieux les centroïdes seront définis.

Cette méthode permet donc de regrouper les éléments d'un ensemble par voisinage. L'inconvénient majeur de cette méthode est qu'il nécessite de spécifier un nombre de partitions k, et un mauvais choix peut entrainer un partitionnement erroné. Dans notre utilisation des k-means, cela n'a pas beaucoup d'importance si ce n'est dans le coût calculatoire puisqu'on ne cherche pas à classifier *stricto sensu* les points mais plutôt à partitionner l'espace de recherche.

4.1.3.2 La méthodes des k-means hiérarchique

Chaque partition créée par la méthode des k-means est de fait elle-même un ensemble. À moins d'être composé d'un seul élément, cet ensemble peut donc à son tour être partitionné par les mêmes moyens. Ainsi, récursivement des partitions de plus en plus petites sont créées jusqu'à atteindre un nombre d'éléments minimum fixé q. Un arbre est créé par cette récursivité, les noeuds de l'arbre contenant les centroïdes de l'ensemble des éléments des branches filles. Les feuilles, les terminaux de l'arbre sont les éléments eux-mêmes. L'illustration 4.4 permet de comprendre cette structure.

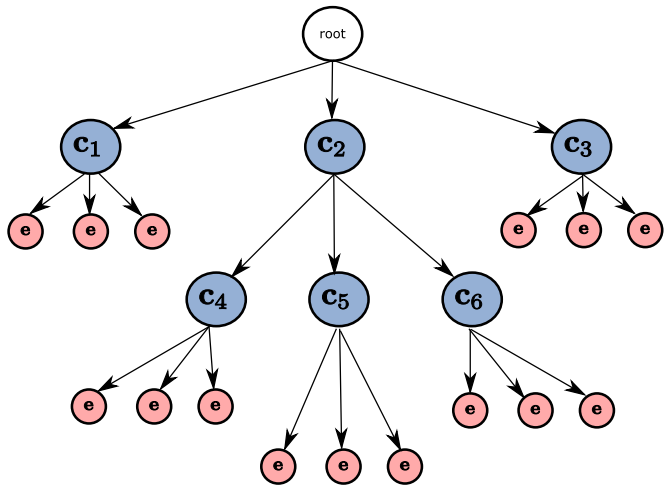


FIGURE 4.4 – Illustration de l'arbre résultant du k-means hiérarchique

La recherche du plus proche voisin consiste alors en la recherche du plus court

chemin dans un arbre, la longueur de l'arc étant défini par la distance de l'élément au noeud de l'arbre. En démarrant à la racine de l'arbre, et pour chaque noeud sélectionné, on cherche le noeud/feuille enfant dont la distance est la plus petite. De cette manière, tous les éléments appartenant à des branches non minimales ne sont pas pris en compte, permettant un gain de temps considérable (Best Bin First).

Comme précisé précédemment, cette solution est approximative, puisqu'elle suppose que si **g** est proche du centroïde, l'élément le plus proche est forcément dans la partition fille. C'est à priori le cas si le jeu de données d'apprentissage est suffisamment important et varié. Nos tests ont d'ailleurs montré que cette solution était tout à fait fonctionnelle. Soit x le nombre d'étages maximum de l'arbre, la complexité est alors au pire de $O(xk)$, bien en deçà de $O(n)$. Dans le meilleur des cas, si les différents partitionnements sont bien équilibrés, alors le nombre d'étages peut être estimé par :

$$k^x = \frac{n}{q} \tag{4.3}$$

$$x = \frac{ln(\frac{n}{q})}{ln(k)} \tag{4.4}$$

4.2 Mise en application

4.2.1 Choix de la méthode

L'application devant être utilisée dans des environnements non contrôlés, et donc où les possibles occlusions font partie des contraintes fortes, la seconde classe de méthodes (basée sur de multiples primitives) est choisie. Cette classe contient un vaste choix de méthodes telles que MSER [Matas 02], Harris [Shi 94], SIFT [Lowe 04], PCA-SIFT [Ke 04], GLOH [Mikolajczyk 05], SURF [Herbert 08], FERNS [Ozuysal 07], etc. Après de multiples tests sur des images prises en environnement réel, il est apparu que SIFT (et sa variante Affine-SIFT) donne des résultats toujours supérieurs dès lors que le point de vue est fortement différent de celui de la base de données. Cette dernière approche a donc été choisie pour décrire le contenu des motifs. La méthode SIFT est décrite dans l'encadré 4.5 et un exemple est donné dans la figure 4.6.

Affine-SIFT donne des résultats meilleurs que SIFT dans les conditions très difficiles – comme une déformation très forte du motif – mais nécessite environ le double de ressources calculatoires. Il pourra être choisi ultérieurement en fonction des capacités du système.

4.2.2 Chargement de la base de donnée

On dispose d'une base de données d'images que l'on suppose présent dans l'environnement et qui représente des zones planaires. Ces images, chargées séquentiellement au démarrage de l'application, sont traitées afin de produire un ensemble **E** important

L'algorithme des Scale Invariant Features Transform
[Lowe 04]

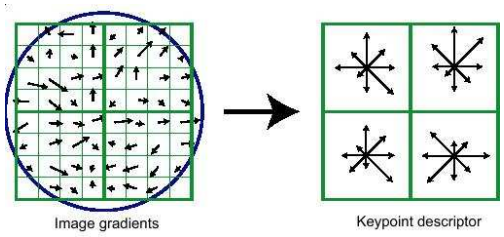
L'algorithme est divisé en quatre étapes :

Détection des extremums dans l'espace d'échelle. Les points clés sont détectés en calculant les extremums de l'espace d'échelle construit à partir du calcul de différence de Gaussiennes (DoG) sur les octaves d'une pyramide Gaussienne.

Localisation des points d'intérêt. Cette méthode de sélection produit trop de points d'intérêt dont certains ne sont pas des indices stables. L'étape de localisation repose sur une approximation de la fonction DoG au voisinage des extremums par un développement de Taylor qui permet d'affiner la localisation des points et de leur attribuer une échelle et un poids relatif à la courbure principale. Ainsi, les points trop proches l'un de l'autre, les points qui ont un trop faible contraste et ceux qui sont situés à un emplacement quelconque sur un contour sont éliminés.

Calcul de l'orientation. Une ou plusieurs orientations relatives à la direction des gradients et à leur amplitude sont associées à chaque point d'intérêt, ce qui assure l'invariance à l'orientation.
L'amplitude et la direction du gradient sont calculées pour chaque pixel au voisinage du point d'intérêt et un histogramme de dimension 36 (soit $36 \times 10^{\circ}$) est construit. Chaque échantillon est pondéré par l'amplitude de son gradient et sa distance au point.
Les pics de l'histogramme correspondent aux orientations dominantes qui sont ajoutées au descripteur. Si plusieurs orientations sont concurrentes, alors un autre point clé, copie du premier est créé et la deuxième orientation lui est affectée.

Construction des descripteurs.



L'amplitude et la direction du gradient sont calculées pour les pixels voisins d'un point d'intérêt. L'amplitude est pondérée par une Gaussienne centrée sur le point. Les résultats sont stockés dans quatre histogrammes d'orientation de dimension 8. Sur la figure ci dessus, la longueur des flèches représente la valeur de l'histogramme pour cette orientation. La figure présente un descripteur de 2×2 histogrammes de 8 dimensions. Les SIFT sont en fait implémentés en utilisant des descripteurs de 4×4 histogrammes de 8 dimensions, soit $4 \times 4 \times 8 = 128$ éléments pour chaque descripteur.

FIGURE 4.5 – Présentation des SIFT

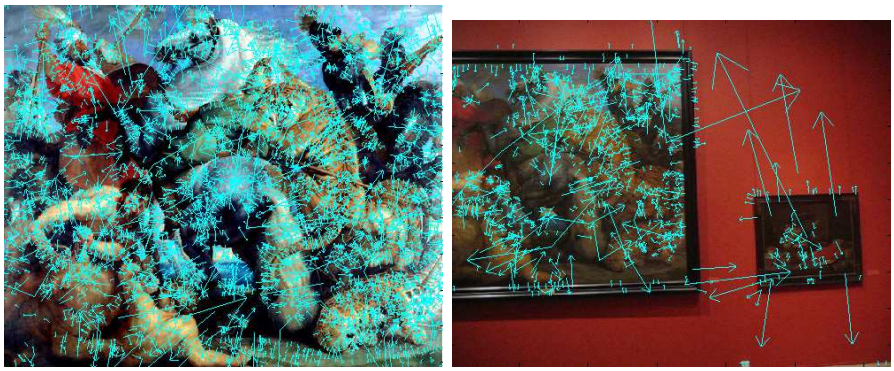


FIGURE 4.6 – Extraction de points SIFT dans l'image de référence et courante

de descripteurs SIFT. Chacun de ces points est relié par un index à l'image à laquelle il appartient et les coordonnées de ce point sont également enregistrées. Une fois l'ensemble obtenu, celui-ci fait l'objet d'un post-processing afin de construire un arbre selon la méthode des k-means hiérarchiques (section 4.1.3.2). Ce traitement est relativement long puisqu'il faut analyser l'ensemble de la base de données. Cependant, cet arbre peut être enregistré dans un fichier par sérialisation et rechargé ultérieurement pour les prochaines exécutions avec la même base de données.

Un jeu d'essai contenant la reproduction numérique de 74 tableaux (illustrés par la figure 4.9) est utilisé dans les tests qui suivent. Ces tableaux sont situés et exposés au musée des beaux-arts de Rennes. Le nombre d'images relativement important permet d'avoir une idée des ressources nécessaires et des résultats en conditions réelles. Cette base de données génère plus de 40000 points répartis suivant l'histogramme de la figure 4.7. Ces chiffres permettent de comprendre la nécessité d'éviter une recherche exhaustive et d'utiliser des méthodes optimisées de recherche dans la base de données.

4.2.3 Recherche de l'image

Une image à analyser génère un autre ensemble de points. Chacun de ces points fait l'objet d'une mise en correspondance dans la base de données de points. Les deux points les plus proches sont extraits de la base de données. Si le premier point est suffisamment proche et à la fois suffisamment éloigné du second point, la mise en correspondance est considérée comme valide.

Cette mise en correspondance génère souvent de nombreux faux positifs. Deux dispositifs complémentaires sont utilisés pour contrer cet effet involontaire. Le premier consiste à ne prendre en compte que les points liés à l'image de la base de données majoritaire (les autres images, éventuellement présentes, de la base de données peuvent

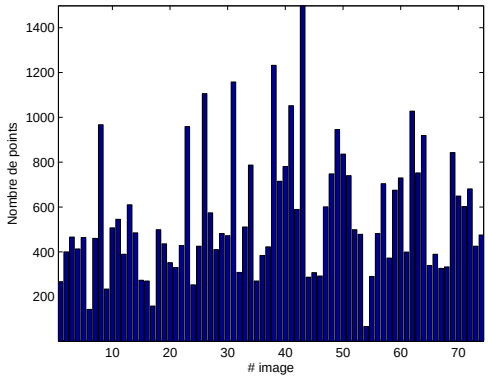


FIGURE 4.7 – Répartition des points dans la base de donnée

être détectées en déterminant l'image majoritaire après la suppression des points de l'image précédemment sélectionnée).

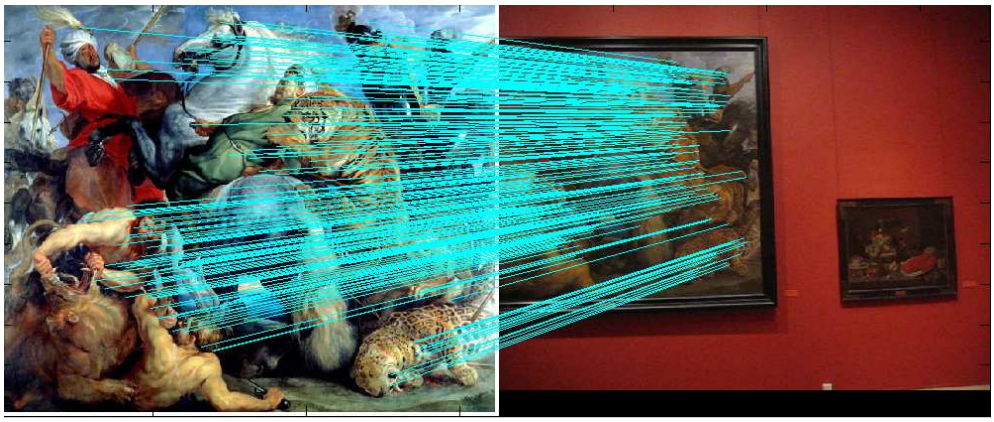


FIGURE 4.8 – Matching de points entre une image de la base de données du musée et une prise de vue

Le deuxième outil utilisé consiste à vérifier la cohérence géométrique de la mise en correspondance. Sachant que l'image est une zone planaire dans la prise de vue courante, une homographie **doit** pouvoir transposer les points $\mathbf{p}_i$ de l'image de référence (capturée par une caméra rattachée au repère $\mathcal{F}_{c1}$) vers les coordonnées dans l'image courante (capturée par une caméra rattachée au repère $\mathcal{F}_{c2}$) :

$$ {}^{c2}\mathbf{p_i} \quad = \quad {}^{c2}\mathbf{H}_{c1}\,{}^{c1}\mathbf{p_i} \tag{4.5} $$

Considérant tout de même que la majorité des points mis en correspondance sont des vrais positifs, une homographie faisant **consensus** est recherchée. Cette homographie

est de toute manière utile pour connaitre la transformation réalisée sur l'image de référence. Ce consensus est trouvé grâce à la méthode RANSAC (RANdom SAmple Consensus, expliqué en annexe). Les points ne se conformant pas au consensus sont considérés comme des "outliers" et éliminés.

4.2.4 Utilisation des résultats

La reconnaissance d'un template est évidemment intéressante pour connaitre l'objet filmé. C'est également la garantie de connaitre une zone planaire utilisable par notre tracker. C'est optionnellement une source d'informations sur les plans de l'environnement puisque les paramètres du plan du mur dans le repère global peut être connu à priori pour chaque tableau. C'est également optionnellement une source pour calculer la pose initiale de la caméra, si on souhaite fixer le repère global sur ce tableau ou que l'on connait la position spatiale de ce dernier.

4.3 Initialisation des plans inconnus

L'estimation de la pose selon la méthode décrite dans ce document repose sur l'utilisation de plans. Ces plans sont utilisés pour estimer la mesure (la matrice d'homographie) et sont donc indispensables au bon fonctionnement de l'estimateur décrit dans le chapitre 2. Dans ce chapitre 2.3, les paramètres des plans étaient supposés approximativement connus à l'avance, à travers l'utilisation de bases de données SIG ou de plans CAO. Connaissant approximativement la précision des bases de données utilisées, les paramètres des plans pouvaient être intégrés dans le vecteur d'état avec une incertitude donnée. Ces informations ne sont malheureusement ni exhaustives ni disponibles partout et de nombreuses régions planaires filmées par la caméra ne sont pas référencées par les bases de données. Sans information sur les paramètres de ces plans, le calcul de pose devient impossible. Il est donc impératif de trouver une solution qui permette d'estimer les paramètres d'un plan filmé sans avoir d'*a priori* sur celui-ci.

Cette section s'attache à expliquer quelles sont les contraintes imposées et la solution apportée dans cette thèse. Cette section ne s'intéresse pas à la recherche de zones planaires dans l'image et la région suivie (dont les paramètres sont à estimer) est considérée être planaire. La sélection des régions intéressantes sera abordée ultérieurement.

4.3.1 Problématique

Il est nécessaire tout d'abord de rappeler que l'estimation – des paramètres de la caméra et de la carte – se fait dans le cadre d'un filtre de Kalman étendu. Les données injectées dans le vecteur d'état doivent suivre une loi de probabilité normale. Impossible donc de considérer une composante de ce vecteur comme suivant une loi uniforme et de pouvoir initialiser aléatoirement un composant de ce vecteur. De la même manière, il n'est pas possible d'utiliser une très grande variance pour simuler une loi uniforme, ceci causant des problèmes d'estimation inévitables notamment à cause de la linéarisation

FIGURE 4.9 – Un aperçu de la base de données utilisée

des modèles de mesure et de déplacement. Il faut impérativement ajouter au vecteur d'état une valeur qui puisse être décrite par une loi Gaussienne.

Une tentation naturelle est d'utiliser une méthode analytique pour estimer les paramètres des plans. Rappelons que ${}^{c1}\mathbf{M}_o$ est la pose de la caméra correspondant à l'image où le plan est observé pour la première fois, et ${}^{c2}\mathbf{M}_o$ la pose de la caméra à l'instant courant. Si la distance entre les deux poses est suffisante, l'homographie mesurée contient les informations sur le mouvement mais également sur le plan mesuré. Les trois composantes ${}^{c2}\mathbf{R}_{c1}$, $\frac{{}^{c2}\mathbf{t}_{c1}}{{}^{c1}d}$ et ${}^{c1}\mathbf{n}$ peuvent être extraites de la matrice par l'utilisation adéquate des valeurs singulières de la matrice d'homographie [Malis 98, Hartley 01]. En utilisant ${}^{c2}\mathbf{M}_o$ et ${}^{c1}\mathbf{M}_o$, les paramètres du plan peuvent être transformés comme souhaité pour obtenir ${}^{o}\mathbf{n}$ et ${}^{o}d$.

Le premier problème de cette solution vient du fait qu'elle n'est pas progressive. Elle nécessite un déplacement suffisant de la caméra pour pouvoir fournir un résultat correct. Cette solution pêche également par son approche déterministe. Elle ignore par construction l'incertitude des paramètres du vecteur d'état et l'ajout du résultat biaise donc la construction de la carte et de l'estimation dans son ensemble. Une erreur par exemple dans l'estimation de ${}^{c1}\mathbf{M}_o$ conduirait à une erreur dans le calcul de ${}^{o}\mathbf{n}$. L'incertitude des paramètres du vecteur d'état, qui permet d'appréhender correctement cette erreur avec l'estimateur, n'est pas propagée aux paramètres de ${}^{o}\mathbf{n}$. Il est donc primordial et indispensable d'apporter une solution qui permette d'intégrer l'incertitude à l'estimation des paramètres du nouveau plan.

Cette problématique d'initialisation est transverse à toutes les implémentations du SLAM par vision monoculaire. C'est la grande problématique qui distingue les recherches sur le SLAM général des recherches sur le SLAM monoculaire où une dimension manque dans la mesure. La littérature a largement traité ce problème dans le cas des points, d'où des solutions adaptées au problème de cette thèse peuvent être extraites.

4.3.2 Ajouts d'éléments et SLAM monoculaire : état de l'art

Soit $\mathbf{X}$, un point 3D défini dans $\mathcal{F}_o$. Soit ${}^{c1}\mathbf{x}$ sa projection dans l'image correspondant au repère $\mathcal{F}_{c1}$. ${}^{c1}\mathbf{x}$ étant une projection, aucune information n'est disponible sur sa profondeur. Seul est défini un rayon infini partant de la caméra et passant par le point projeté. La probabilité que $\mathbf{X}$ soit à une profondeur précise est *a priori* uniforme sur tout le rayon. Les approches présentées dans la littérature cherchent à compenser cette loi uniforme pour obtenir une loi Gaussienne utilisable dans l'EKF. La liste des solutions présentées ci-après n'est pas exhaustive mais contient les propositions majeures qui sont utilisables en temps-réel.

Davison est le premier à fournir une solution d'initialisation dans le cadre du SLAM monoculaire [Davison 03]. Une distance maximum du point est définie, chose accep-

table dans un environnement d'intérieur. Le rayon est alors un segment 3D reliant ce maximum à l'objectif de la caméra. Des hypothèses sur la probabilité que le point soit à une profondeur donnée sont faites uniformément le long de ce rayon. La vraisemblance est calculée en cherchant une ressemblance avec un modèle du point reprojeté. Ces hypothèses sont des particules et l'estimation est faite avec un filtre particulaire à une dimension (illustré par la figure 4.10). La première image donne évidemment une probabilité équivalente pour chaque particule. Avec le mouvement de la caméra, il est hautement probable qu'une particule prenne l'ascendant et provoque alors un pic dans la fonction de densité de probabilité. Cette dernière est approximée à ce moment là par une fonction Gaussienne. Le point étant défini dans les 3 dimensions par une Gaussienne, il peut être ajouté correctement au vecteur d'état et être utilisé comme support.

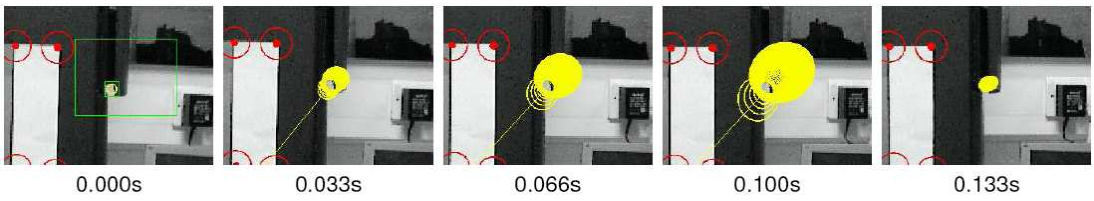


FIGURE 4.10 – Exemple de recherche de la profondeur du point avec le filtre particulaire 1D (Image extraite de [Davison 03])

Hormis la nécessité de définir une profondeur maximale incompatible avec des environnements extérieurs (l'utilisation d'une profondeur maximale importante nécessitant un nombre important de particules et donc un coût calculatoire plus important), cette méthode pose le problème du délai d'utilisation des mesures. En effet, le point est inutilisable le temps que la fonction Gaussienne soit approximée. Cela peut poser des problèmes notamment dans les cas où les points ont une courte durée de vie. À l'opposé, le délai permet de ne pas encombrer le vecteur d'état de points peu fiables, l'augmentation de la taille du vecteur d'état ayant évidemment un coût calculatoire important. Le choix fait dans cette thèse est de privilégier l'information et de chercher à profiter des mesures au plus vite.

Sola propose une méthode [Sola 05] permettant d'intégrer dès la première image le point mesuré dans la carte. Le point est représenté par une attitude et une profondeur dans le repère de la caméra courante. L'auteur remarque tout d'abord que l'incertitude maximum acceptable pour un EKF (à cause de la linéarisation) augmente avec la distance du point, de par le phénomène de projection. D'autre part, tout comme on peut modéliser une fonction complexe par une combinaison de fonctions plus simples, l'article (réutilisant les recherches de [Kwok 04, Alspach 72]) explique que l'on peut approximer grossièrement la loi uniforme par une somme de Gaussiennes (voir figure 4.11).

L'équation de la loi de probabilité est alors défini par la somme :

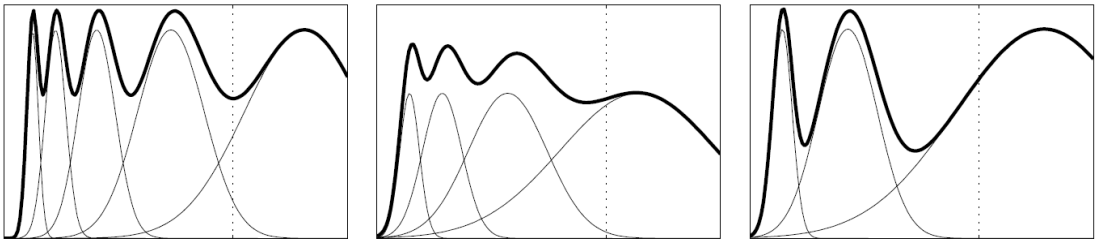


FIGURE 4.11 – Approximations d'une loi normale par une somme de Gaussiennes

$$\sum_{i=1}^{j} \mathcal{N}(\mu_i, \sigma_i) \tag{4.6}$$

où les μ_i et σ_i sont sélectionnés en fonction des bornes minimales et maximales de la profondeur du point de manière à ce que le ratio entre la profondeur et l'incertitude reste constant pour chaque Gaussienne.

Chaque Gaussienne peut être vue comme une estimation différente du même point. Il semble logique de devoir créer autant de cartes qu'il y a de Gaussiennes, ce qui amène à des coûts calculatoires bien trop importants. L'intéret de cet article est de proposer une solution pour insérer l'ensemble des Gaussiennes dans la même carte en utilisant une pondération fonction de la vraisemblance de chaque Gaussienne. Comme pour un filtre particulaire, lorsqu'une Gaussienne a un poids trop faible, elle est éliminée du jeu de solutions. L'objectif étant bien sûr de converger vers une seule Gaussienne et de pouvoir alors représenter le point par une Gaussienne 3D.

Le principe est fortement similaire à celui de la première méthode, mais permet cette fois-ci de ne pas avoir de latence dans l'utilisation des mesures. Reste le problème des bornes imposées, qui peut être problématique principalement pour la borne maximale dans des environnements larges.

Montiel propose [Montiel 06] de considérer un point comme un vecteur partant de la position de la caméra au moment où il a été détecté et allant jusqu'au point réel. Le point peut être ainsi codé par la position ${}^{c1}\mathbf{t}_o$, l'orientation du vecteur (en coordonnées sphériques ϕ et θ) et la norme p de ce vecteur (la distance Euclidienne du point à la position de la première caméra). Le point reste mesurable puisque ce vecteur peut être transformé dans le repère de la caméra courante. Il est représenté par 6 éléments tous mesurables directement à l'aide du vecteur d'état (et donc défini par une Gaussienne) sauf la magnitude p.

L'originalité de cet article est de proposer d'estimer non plus p mais son inverse $d = \frac{1}{p}$. L'article démontre que cela permet de représenter des points à l'infini aussi bien que des points très proches. L'auteur prouve surtout que cette représentation permet

une très faible erreur de linéarisation du modèle de mesure que la profondeur soit infinie ou faible et qu'elle permet donc de coder une incertitude énorme. L'incertitude de d peut donc être représentée par une Gaussienne alors qu'on ne dispose d'aucun *a priori* sur cette dimension.

Un exemple est donné pour une distance bornée entre $[1; +\infty[$. La variance suffisante pour représenter cette intervalle avec 95% de confiance est de seulement 0.25. Ce résultat peut être compris en calculant les bornes de d qui ne sont plus qu'entre $]0; 1]$. Pour un tel intervalle réduit, on peut comprendre qu'une Gaussienne avec une faible variance permette de représenter l'ensemble des possibilités avec une très haute confiance (figure 4.12). L'inverse de la profondeur est initialisée (choisie apparemment heuristiquement) au centre de l'intervalle (soit 0.5 dans l'exemple donné) dans le vecteur d'état.

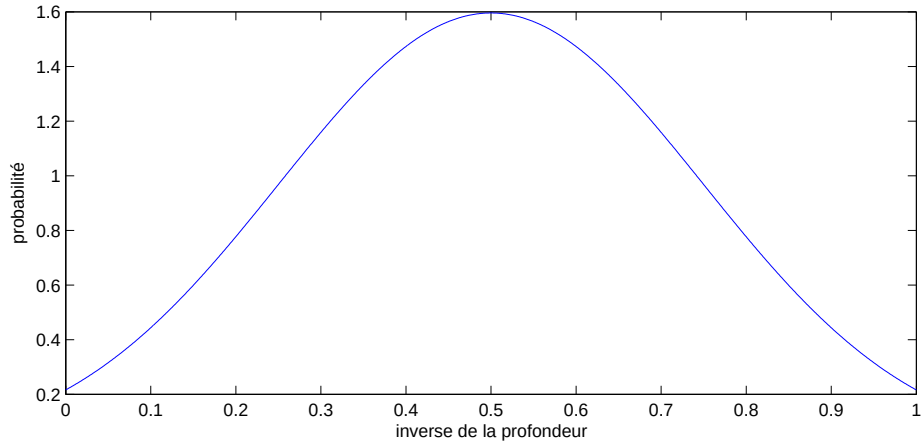


FIGURE 4.12 – La fonction Gaussienne sur l'intervalle de la profondeur inverse [Montiel 06]

Cette solution semble idéale puisqu'elle permet d'éviter tout délai dans l'utilisation des mesures tout en ne contraignant pas la profondeur des points à une distance maximale.

4.3.3 Initialisation des paramètres des plans

Dans un espace Euclidien à 3 dimensions, le plan constitue l'élément géométrique qui sépare cet espace en 2. Sans aucune information *a priori* sur ce plan, ses paramètres peuvent donc être quelconques. En reprenant la notation du plan utilisée dans notre implémentation du SLAM :

$$N = \begin{bmatrix} \phi \\ \theta \\ d \end{bmatrix} \tag{4.7}$$

ϕ et θ (les paramètres d'orientation du plan) sont librement situés dans l'intervalle $[0; 2\pi[$. L'intervalle reste de toute manière trop important, l'équation de mesure n'étant

pas suffisamment linéaire. d est quant à lui librement défini dans $\mathbb{R}$, soit dans l'intervalle $]-\infty;+\infty[$. La seule contrainte sur les paramètres est que le plan était visible dans le repère $\mathcal{F}_{c1}$ de la caméra où il a été observé. Il ne peut donc pas être perpendiculaire au plan image de cette caméra, ni situé en dehors du champ de la caméra. Ces contraintes ne sont pas du tout suffisantes pour représenter sur un petit intervalle les paramètres.

Reprenons l'équation paramétrique classique d'un plan :

$$aX + bY + cZ + d = 0 \tag{4.8}$$

$$\mathbf{n}^{\mathsf{T}}\mathbf{X} + d = 0 \tag{4.9}$$

La normale $\mathbf{n}$ du plan est perpendiculaire aux vecteurs situés sur le plan. Posons deux vecteurs $\mathbf{v1}$ et $\mathbf{v2}$ non parrallèles. La normale est algébriquement déterminée par :

$$\mathbf{n} = \mathbf{v1} \times \mathbf{v2} \tag{4.10}$$

Dans un espace Euclidien, un vecteur peut être représenté par deux points. En imposant que $\mathbf{v1}$ et $\mathbf{v2}$ aient le même point d'origine, ces deux vecteurs peuvent être représentés par 3 points au total. La normale peut donc être exprimée par 3 points non colinéaires. Comme le dernier terme de l'équation (4.8) d est fonction des paramètres de la normale et d'un point du plan, ce terme est également connu à partir de ces 3 points. Le plan est ainsi déterminé entièrement.

Reprenons l'homographie entre deux caméras telle qu'elle est utilisée dans le modèle de mesure classique exprimé dans la section 2.7 :

$${}^{c2}\mathbf{H}_{c1} = {}^{c2}\mathbf{R}_{c1} - \frac{1}{{}^{c1}d}{}^{c2}\mathbf{t}_{c1}{}^{c1}\mathbf{n}^{\mathsf{T}} \tag{4.11}$$

$${}^{c2}\mathbf{H}_{c1} = {}^{c2}\mathbf{R}_{c1} - {}^{c2}\mathbf{t}_{c1}{}^{c1}\mathbf{n'}^{\mathsf{T}} \tag{4.12}$$

où pour simplifier nos futurs calculs

$${}^{c1}\mathbf{n}' = \frac{1}{{}^{c1}d}{}^{c1}\mathbf{n} \tag{4.13}$$

$$||{}^{c1}\mathbf{n}'|| = \frac{1}{{}^{c1}d} \tag{4.14}$$

Posons 3 points définis dans $\mathcal{F}_{c1}$.

$${}^{c1}\mathbf{X}_i = \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^{\mathsf{T}} \forall i = 1..3 \tag{4.15}$$

En réutilisant 4.8, l'équation devient

$$a'X + b'Y + c'Z = -1 \tag{4.16}$$

et donc pour les trois points définis, le système inversible est obtenu directement

$$\begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ X_3 & Y_3 & Z_3 \end{bmatrix} \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \tag{4.17}$$

En projetant les 3 points définis dans $\mathcal{F}_{c1}$ dans le plan $z = 1$:

$$
\begin{aligned}
{}^{c1}\mathbf{x}_i &= \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} && (4.18)\\
&= \begin{bmatrix} \frac{X_i}{Z_i} \\ \frac{Y_i}{Z_i} \\ \frac{Z_i}{Z_i} \end{bmatrix} && (4.19)
\end{aligned}
$$

Le système peut être réécrit en utilisant la projection des points

$$
\begin{bmatrix} Z_1 & 0 & 0 \\ 0 & Z_2 & 0 \\ 0 & 0 & Z_3 \end{bmatrix}
\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}
\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}
= \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \tag{4.20}
$$

${}^{c1}\mathbf{n}'$ est déterminé par la résolution du précédent système :

$$
\begin{aligned}
{}^{c1}\mathbf{n}' &= \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}^{-1}
\begin{bmatrix} Z_1 & 0 & 0 \\ 0 & Z_2 & 0 \\ 0 & 0 & Z_3 \end{bmatrix}^{-1}
\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} && (4.21)\\
&= \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}^{-1}
\begin{bmatrix} Z_1^{-1} & 0 & 0 \\ 0 & Z_2^{-1} & 0 \\ 0 & 0 & Z_3^{-1} \end{bmatrix}
\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} && (4.22)
\end{aligned}
$$

Les points ${}^{c1}\mathbf{x}_i$ sont choisis comme les coordonnées des coins de la région estimée dans l'image du repère $\mathcal{F}_{c1}$ et ne sont pas à estimer [1]. Les seules données à estimer sont les variables Z_1^{-1}, Z_2^{-1} et Z_3^{-1}, respectivement les inverses des profondeurs des points ${}^{c1}\mathbf{X}_1$, ${}^{c1}\mathbf{X}_2$ et ${}^{c1}\mathbf{X}_3$ que l'on regroupera dans un vecteur ${}^{c1}\mathbf{in}$.

Comme dans [Montiel 06], la profondeur peut être estimée dans le SLAM indirectement par son inverse. Les poses ${}^{c2}\mathbf{M}_o$ et ${}^{c1}\mathbf{M}_0$ sont toujours stockées dans le vecteur d'état. Seule la représentation du plan change pendant l'initialisation. Au lieu d'utiliser la représentation (4.7), les 3 variables Z_1^{-1}, Z_2^{-1} et Z_3^{-1} sont stockées et initialisées chacune de la même manière que l'inverse de la profondeur de [Montiel 06].

Le modèle de mesure $\mathbf{z}$ [chapitre slam] est remplacé par le modèle de mesure $\mathbf{z}'$ tant que les paramètres des plans ne peuvent pas être déterminés avec une incertitude Gaussienne.

$$
\mathbf{z}' = {}^{c2}\mathbf{R}_{c1} - {}^{c2}\mathbf{t}_{c1}{}^{c1}\mathbf{n}'^{\mathbf{T}} \tag{4.23}
$$

1. Les points sont ordonnés de manière à apparaitre dans la première image dans le sens trigonométrique afin que la normale soit bien dirigée vers la caméra

La description des Jacobiennes du modèle de mesure $\mathbf{z}'$ – utilisé pendant l'initialisation – est donnée dans l'encadré 4.13. Les points étant représentés dans le repère $\mathcal{F}_{c1}$, le plan résultant n'a pas besoin d'etre transformé dans ce repère, et les Jacobiennes sont donc plus simples que dans le modèle normal. Il est important de noter que la Jacobienne en fonction du plan est constante (elle ne contient pas les termes Z_i^{-1}). Ceci démontre la forte linéarité du modèle en fonction des paramètres de profondeur inverse et donc le bon fonctionnement théorique de la méthode d'initialisation.

Jacobienne de $\mathbf{z}'$ en fonction du plan

$$\frac{\partial \mathbf{H}}{\partial \mathbf{n}'} = \frac{\partial \mathbf{H}}{\partial^{\mathbf{c1}}\mathbf{n}'}\frac{\partial^{\mathbf{c1}}\mathbf{n}'}{\partial^{\mathbf{c1}}\mathbf{in}} \tag{4.24}$$

$$\frac{\partial \mathbf{H}}{\partial^{\mathbf{c1}}\mathbf{n}'} = -(\mathbf{I} \otimes {}^{c2}\mathbf{t}_{c1}) \tag{4.25}$$

$$\frac{\partial^{\mathbf{c1}}\mathbf{n}'}{\partial^{\mathbf{c1}}\mathbf{in}} = -\begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix}^{-1} \tag{4.26}$$

Jacobienne de $\mathbf{z}'$ en fonction de la caméra de référence

$$\frac{\partial \mathbf{H}}{\partial \mathbf{x_{c1}}} = \frac{\partial^{\mathbf{c2}}\mathbf{R_{c1}}}{\partial \mathbf{x_{c1}}} - (({}^{c1}\mathbf{n}' \otimes \mathbf{I})\frac{\partial^{\mathbf{c2}}\mathbf{t_{c1}}}{\partial \mathbf{x_{c1}}} \tag{4.27}$$

$$\tag{4.28}$$

Jacobienne de $\mathbf{z}'$ en fonction de la caméra courante

Cette jacobienne est la même que dans le modèle de mesure de base.

FIGURE 4.13 – Définition des jacobiennes du modèle de mesure pour l'initialisation

4.3.4 Résultats

4.3.4.1 Simulation numérique

Pour valider le fonctionnement de ce modèle d'initialisation des plans, une simulation numérique est utilisée. Les mesures sont calculées à partir du modèle et sont ainsi parfaites. La scène consiste en 3 plans orthogonaux. Par souci de simplicité, les paramètres des plans sont toujours mesurés par la caméra virtuelle, quelle que soit sa position (pas de hors champ ou d'occlusion). La caméra est initialisée pointant vers la "mire", inclinée vers le bas. Elle effectue un mouvement latéral en -x pendant 100 itérations et repart en $+x$ pendant 100 itérations (permettant de vérifier le bon fonctionnement lorsque le modèle de prédiction n'est plus valide). Aucune information *a*

priori n'est donnée à l'estimateur à propos de la scène ou du déplacement de la caméra. Une forte incertitude sur la prédiction et la mesure est utilisée dans les paramètres du SLAM pour ralentir la convergence et permettre une visualisation des résultats.

La figure 4.14 montre la position estimée et réelle de la caméra en X lors de la séquence. Le mouvement est similaire mais on note une différence de facteur d'échelle entre les deux courbes. Cela est dû au fait que l'estimateur n'a aucune idée ni de son déplacement réel ni de la scène réelle, la projection dans l'image rendant impossible la connaissance de l'échelle de la scène. C'est un problème commun à tous les systèmes de SLAM qui n'a pas de solution si ce n'est utiliser des informations *a priori* ou utiliser des sources d'informations autres que la vision (capteurs inertiels par exemple). La figure 4.15 permet de valider malgré ce facteur d'échelle la bonne reconstruction. Soit **n1**,**n2** et **n3** les trois normales des 3 plans. Si la scène est bien reconstruite, les 3 normales doivent être orthogonales, c'est à dire que :

$$(\mathbf{n1} \times \mathbf{n2})\mathbf{n3} \quad = \quad 1 \tag{4.29}$$

La figure 4.15 montre que la convergence vers cette orthogonalité est très rapide. La valeur de départ est de zéro, ce qui correspond bien aux valeurs par défaut (les trois plans étant initialisés avec les mêmes paramètres), qui font que les trois plans sont parallèles. Au final, la scène est correctement modélisée. La figure 4.16 permet de visualiser cette convergence avec une représentation graphique des plans en 3D.

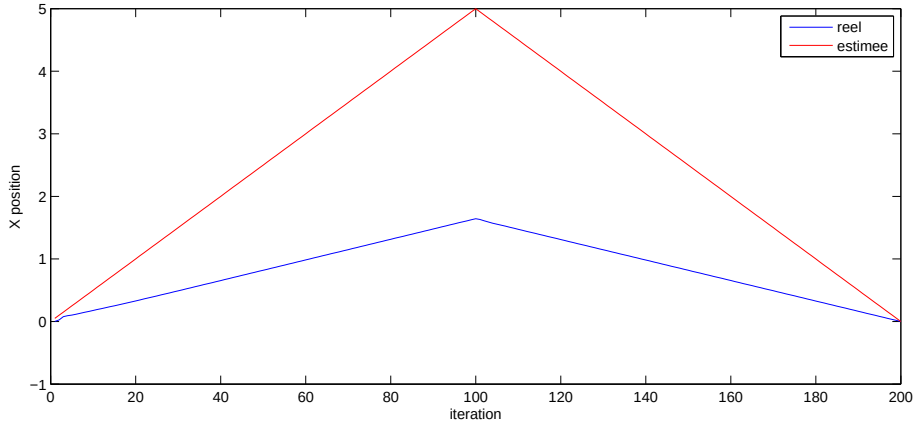


FIGURE 4.14 – Position de la caméra au cours de l'initialisation des plans. La position estimée diffère de la position réelle par un facteur d'échelle impossible à déterminer sans *a priori* sur l'environnement.

4.3.4.2 Séquence d'image

Le deuxième résultat est une séquence dont la scène est similaire à celle simulée précédemment. Une mire à 3 plans, avec des images collées sur chaque plan, est filmée

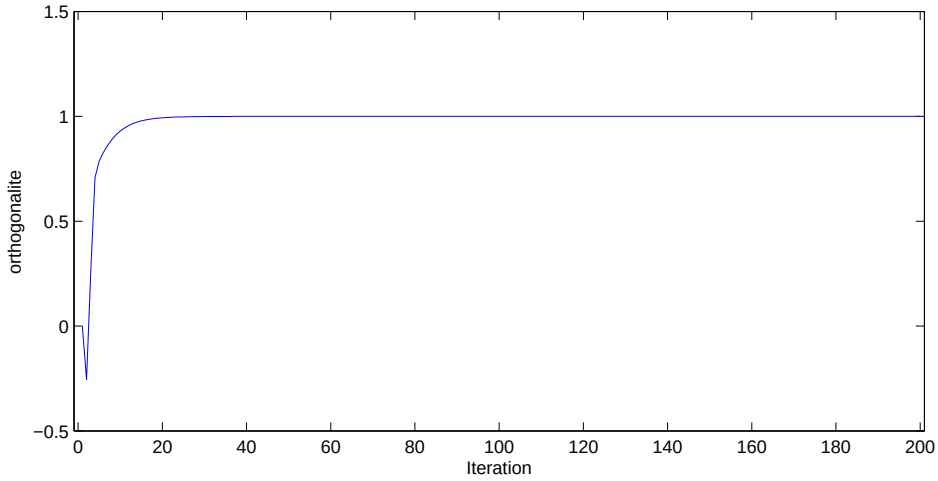


FIGURE 4.15 – Orthogonalité des plans au cours de l'initialisation de ces plans

pendant que la caméra (attachée à un robot "AFMA6" à 6 degrés de liberté) effectue un mouvement de translation et de rotation autour de la mire. Les zones à suivre sont choisies par reconnaissance de région comme présenté dans la section 4.1. Quelques extraits de la séquence avec les zones suivies entourées en rouge sont proposées dans la figure 4.17. La position de la caméra au long de la séquence est décrite par la figure 4.18. La figure 4.19 montre l'orthogonalité des 3 plans comme pour le résultat précédent. La convergence est un peu plus longue à cause d'un faible mouvement de la caméra au début de la séquence. Est également présente sur cette figure une vue 3D des régions suivies selon l'estimation (qui correspond bien à la réalité) à l'itération 244. La figure 4.20 illustre pour un point donné l'évolution de la variance pendant les 50 premières itérations (la suite étant relativement stable). La dernière figure 4.21 représente la profondeur d'un point (parmi les trois d'un plan) le long des 50 premières itérations de la séquence. Moyenne, bornes minimale et maximale permettent de visualiser l'incertitude dans un espace qui a un sens facilement appréhendable. Ces bornes sont calculées pour que leurs distances à la moyenne correspondent à une confiance de 90% connaissant la variance de la variable. Tous ces graphiques permettent de démontrer la qualité de la convergence de la méthode proposée dans une séquence réelle.

4.3.5 Procédure post-initialisation

Lorsque les trois profondeurs inverses des points sont connues avec une incertitude suffisamment faible, le plan peut de nouveau être représenté avec la notation classique présentée dans la section 2.7. La représentation normale (4.7) des paramètres des plans et la représentation basée sur les trois points étant de la même taille (trois scalaires), le changement de représentation ne nécessite pas d'ajout ni de suppression d'éléments dans le vecteur d'état. La fonction (4.22) couplée à la fonction (2.73) permet d'effectuer cette

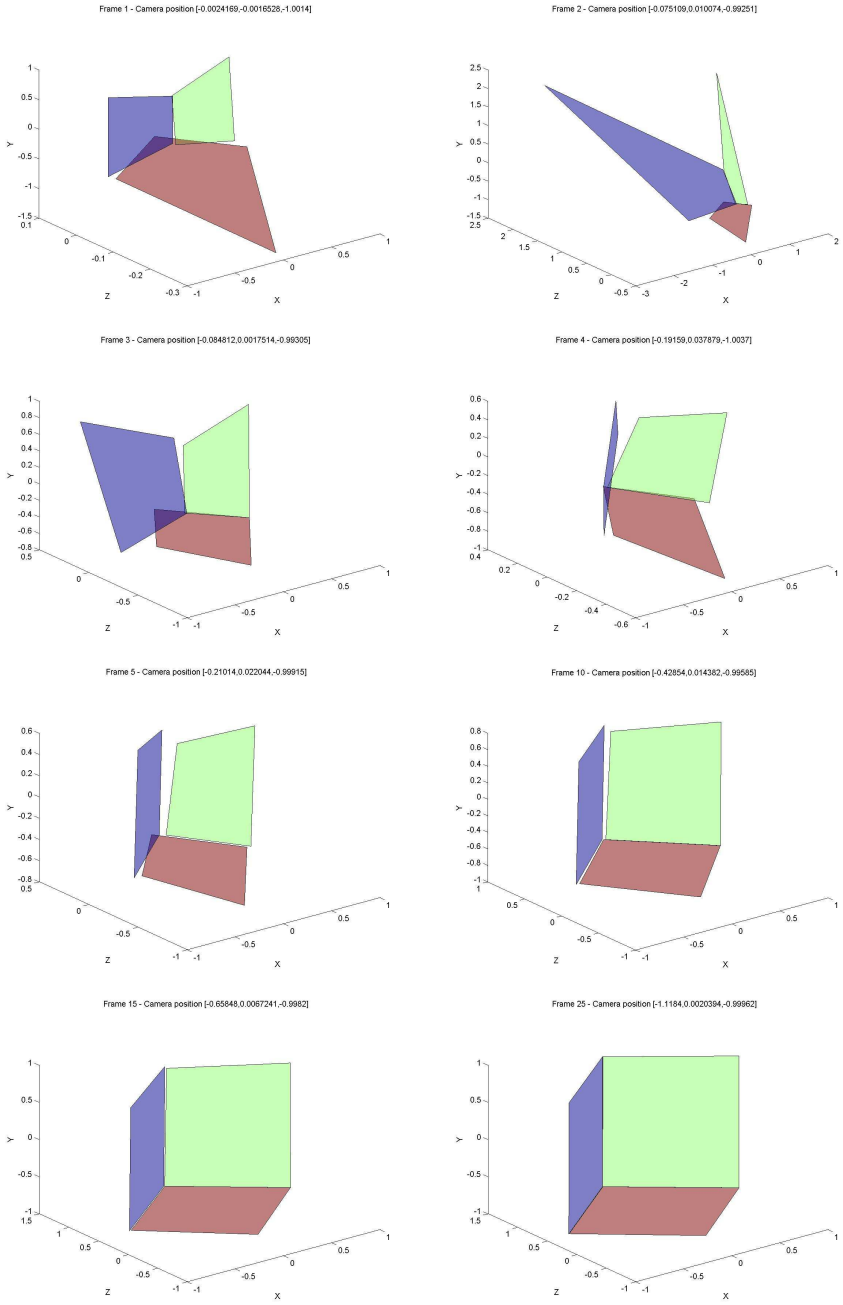


FIGURE 4.16 – Visualisation de la simulation au cours de l'initialisation des plans

conversion. La matrice de corrélation est bien sûr mise à jour en utilisant la Jacobienne correspondante à la composition de ces deux fonctions.

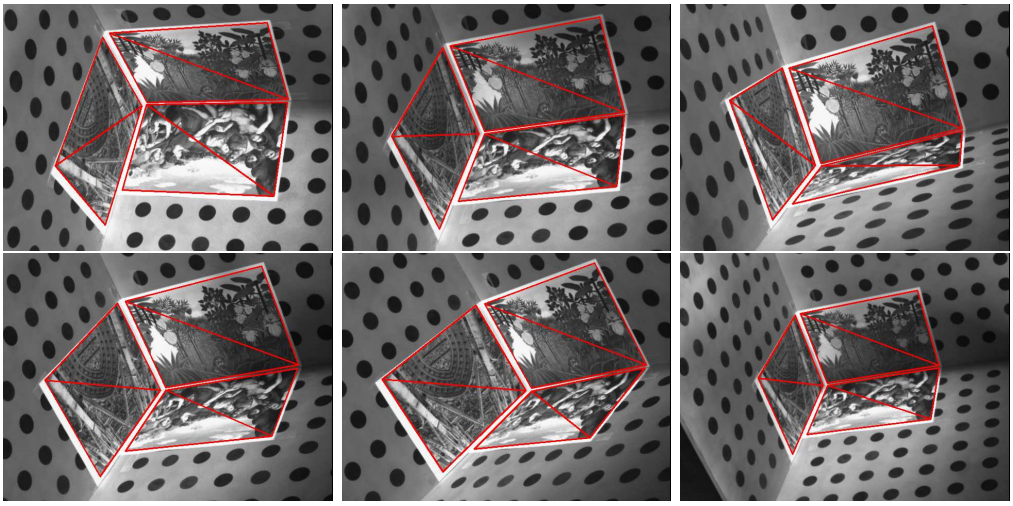


FIGURE 4.17 – Extrait de la séquence d'image utilisée pour le test d'initialisation des plans

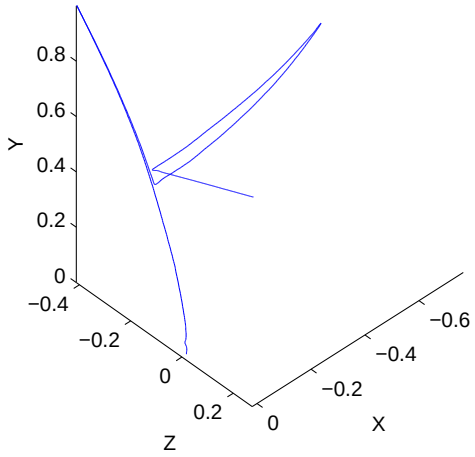


FIGURE 4.18 – Mouvement de la caméra pour le test d'initialisation des plans

Cette transformation est indispensable à la cohérence globale de la carte. La représentation par trois points est "locale" à la zone suivie. Dans l'objectif de pouvoir utiliser pour toutes les zones du même plan le même enregistrement du plan dans le SLAM, il est nécessaire de pouvoir comparer dans un même repère global les résultats

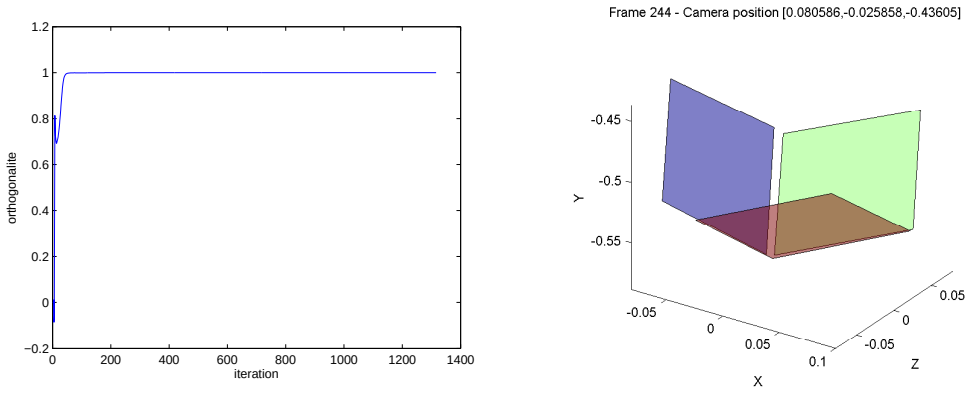


FIGURE 4.19 – Orthogonalité des plans au cours de l'initialisation de ces plans pour la séquence d'images

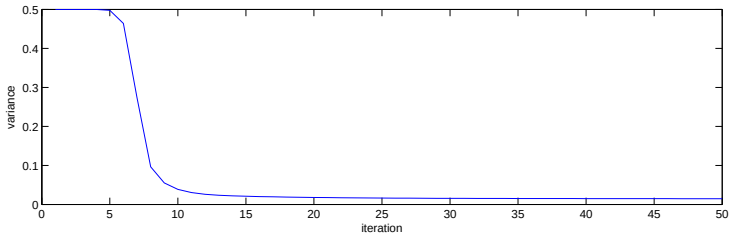


FIGURE 4.20 – Variance de l'inverse de la profondeur d'un des points d'un plan pour les 50 premières itérations de la séquence d'images

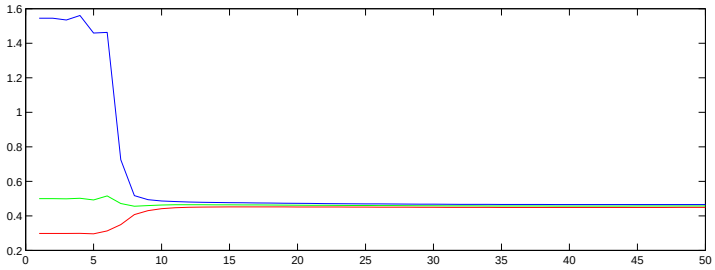


FIGURE 4.21 – Profondeur d'un des points d'un plan pour les 50 premières itérations de la séquence d'image. En vert est affiché l'estimation, en rouge la borne minimale et en bleu la borne maximale pour une confiance de 90%.

de l'initialisation. Cette concentration dans une même variable permet évidemment de profiter des différentes mesures des différentes zones pour avoir une carte cohérente et sans redondance. Un simple test de distance statistique tel que le test du χ^2 permet de déterminer l'appartenance d'une région initialisée à un plan déja existant.

4.3.6 Conclusion sur l'initialisation

Une méthode permettant d'initialiser les plans suivis sans *a priori* vient d'être exposée. Malgré l'absence d'information sur ces plans, cette méthode permet d'enregistrer dans le vecteur d'état d'un EKF les plans dès leur première mesure. Au lieu d'attendre que le plan soit approximativement connu pour l'utiliser dans le SLAM (du fait de la nécessité de l'EKF d'avoir une incertitude Gaussienne) et donc pour connaitre la pose de la caméra, la méthode présentée utilise dès la première mesure les informations fournies par le déplacement dans l'image de la zone suivie. Cette méthode présente une capacité à converger très rapidement vers la solution. Cette méthode d'initialisation, contrairement à d'autres, permet également de considérer des plans extrêmement distants de la caméra. Cet outil est très important dans le cadre de cette thèse, puisqu'il permet de se débarasser des contraintes de connaissances approximatives *a priori* des plans lorsque celles-ci sont indisponibles.

4.4 Extraction des zones planaires

Une scène filmée peut-être extrêmement complexe, comportant de nombreux détails et une géométrie quelconque. La méthode décrite dans cette thèse s'intéresse aux zones planaires de l'environnement.Elle s'intéresse du moins aux zones suffisamment planes pour "paraitre planaires" dans la séquence d'images. Le principal étant que les pixels qui composent ces zones respectent un déplacement homographique. Il est donc nécessaire d'extraire – des images filmées – et de regrouper les régions qui appartiennent à une zone planaire et d'ignorer les autres parties de l'image. La méthode de suivi utilisée(voir chapitre 3) peut renvoyer des mesures erronées si la zone à suivre n'est pas plane (même si souvent l'algorithme détecte rapidement – par des moyens expliqués dans le chapitre 3 – l'incongruité de la zone et l'élimine). Trop de mesures erronées et l'estimation va diverger, d'où l'importance de cette bonne sélection des zones. Notons que cette extraction n'est pas faite en permanence mais lorsque l'algorithme va manquer d'information et qu'il est nécessaire de trouver d'autres sources.

4.4.1 Problématique

D'une manière générale, la problématique est de trouver des zones de l'image suffisamment grandes qui respectent une homographie et qui partagent donc les mêmes paramètres de plan. L'extraction se fait dans le cadre de notre SLAM. Une estimation de la carte et de la pose étant donnée à chaque image, l'algorithme recherché doit profiter de ces informations pour améliorer son fonctionnement. Il doit être capable d'appréhender les configurations dégénérées qui empêchent habituellement d'estimer les paramètres des plans dans l'image telles que les rotations pures et les plans à l'infini. Idéalement, l'algorithme d'extraction doit fournir au SLAM des données le plus rapidement possible, afin d'éviter les latences et les risques de ne plus avoir de mesures visuelles pendant un intervalle de temps. Enfin, l'algorithme se basant sur une séquence d'images et non sur deux images indépendantes (comme c'est le cas dans la majorité

des articles décrivant des méthodes d'extraction de zones planaires), il est important de profiter de la continuité temporelle et de ne pas tout réestimer à chaque image (notamment pour la mise en correspondance des informations visuelles).

4.4.2 État de l'art

L'état de l'art actuel est majoritairement constitué de travaux sur la recherche d'homographies (et donc de plans) utilisant deux images ou plus avec une distance suffisante entre les caméras. Quelques exceptions comme [Kähler 07] essayent de travailler sur des séquences et de détecter les configurations de caméra qui ne permettent pas d'avoir confiance en l'homographie estimée. Une majorité d'articles [Lourakis 02, Bartoli 07] traite de l'extraction des homographies par les méthodes de consensus type RANSAC ou LMedS. [Silveira 06] décrit une solution basée sur une méthode de vote. Ces méthodes semblent fonctionnelles mais ne s'appliquent pas à notre problématique. Elles ne permettent pas une utilisation quasi immédiate des images puisqu'un déplacement minimal de la caméra est requis pour extraire les zones planaires.

4.4.3 Présentation de la méthode d'extraction

Soit $\mathcal{I}$ et $\mathcal{I}'$ deux images prises respectivement dans les repères $\mathcal{F}_{c1}$ et $\mathcal{F}_{c2}$. De $\mathcal{I}$ est extrait un jeu de points caractéristiques $\mathbf{P} = \{\mathbf{p}_i, \forall i = 1 : n\}$ tels les points de Harris. Considérons un maillage $\mathbf{M}$ constitué de triangles $\mathbf{T} = \{\mathbf{t}_i, \forall i = 1 : k\}$ dont chaque triangle a pour coins trois points de $\mathbf{P}$ et dont aucun triangle n'en recouvre un autre. Considérons la reprojection dans $\mathcal{I}'$ de $\mathbf{M}$ et donc de $\mathbf{T}$ et de $\mathbf{P}$. Supposons qu'une homographie soit connue pour chaque triangle qui transfère les points de ce triangle de $\mathcal{I}$ vers $\mathcal{I}'$. Soit un ensemble de triangles $\mathbf{T}'$, sous-ensemble de $\mathbf{T}$ dont les homographies sont similaires et qui sont connectés à au moins un autre triangle de $\mathbf{T}'$ par un côté. Une hypothèse de départ de cette méthode est que tous les points inclus dans l'ensemble des triangles de $\mathbf{T}'$ respectent l'homographie. C'est une hypothèse largement répandue dans la littérature qui peut paraitre abusive puisque seuls les coins des triangles servent de support au calcul de cette homographie. Elle est cependant souvent vérifiée de par les caractéristiques des points choisis qui correspondent aux coins et donc souvent aux changements géométriques (à une variation importante de la normale à la surface).

4.4.3.1 Extraction des points d'intérets

L'ensemble de points $\mathbf{P}$ est généré par la méthode de Harris [Harris 88]. Cette méthode permet de détecter les coins dans les images, plus précisément les points dont la région avoisinante possède de forts gradients dans deux directions différentes. C'est un détecteur de points classiques tel que décrit dans la section 4.1. Dans le cas de la méthode de Harris, le signal de réponse est donné par l'équation :

$$harris(u, v) = det(\mathbf{A}(u, v)) - k * trace(\mathbf{A}(u, v)) \tag{4.30}$$

avec k une valeur typiquement fixée à 0.04 et $\mathbf{A}(u, v)$ est la matrice d'auto-corrélation de la fenêtre autour du point de coordonnées $[u, v]$. Les points donnant une réponse suffisamment forte et suffisamment distants d'un autre point sont sélectionnés pour faire partie de l'ensemble $\mathbf{P}$. Notons qu'un autre détecteur de points peut être potentiellement utilisé. Les points trop près des bords de l'image sont supprimés afin d'assurer une certaine stabilité de l'ensemble $\mathbf{P}$ dans le temps.

4.4.3.2 Maillage

Le maillage des points doit permettre de générer un ensemble de triangle non superposés, dont les points sont significativement non-collinéaires (les trois angles sont suffisamment importants). Ce maillage permet de tisser arbitrairement un lien entre les points de l'image pour faciliter l'extraction des zones planaires, mais n'a pas de signification géométrique stricto sensu. Ces caractéristiques de maillage sont obtenues par l'application d'une méthode classique, la triangulation de Delaunay [Delaunay 34, Devillers 02]. Sans détailler le fonctionnement de cette méthode, le principe de la triangulation de Delaunay est justement de maximiser le plus petit des angles des triangles générés. La figure 4.22 donne une illustration du maillage donné par cette méthode.

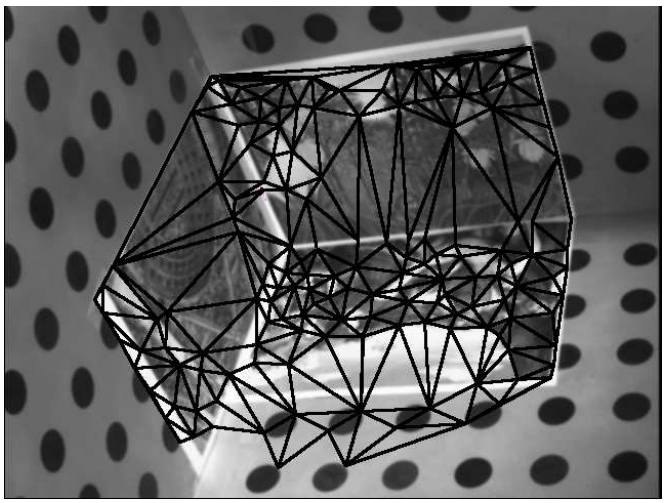


FIGURE 4.22 – Maillage généré par la triangulation de Delaunay sur des points de Harris

L'image peut faire déjà l'objet d'un ou plusieurs suivis de régions planaires initialisées ou non. Afin de ne pas augmenter inutilement le coût calculatoire, les triangles superposés aux régions déja suivies sont supprimés. Cette phase s'effectue après le maillage et non directement sur les points pour s'assurer d'un maillage correct autour de ces zones.

4.4.3.3 Transformation du maillage dans la nouvelle image

L'objectif est d'observer comment les éléments du maillage ont changé par rapport à la première image. Les éléments du maillage doivent être retrouvés dans l'image courante pour reconstituer le maillage. Il s'agit donc, pour simplifier, de retrouver les triangles de $\mathcal{I}$ dans $\mathcal{I}'$. L'extraction des plans se faisant sur une séquence continue d'images et non sur des images dont les prises de vues seraient fort éloignées, il parait plus adéquat d'utiliser les notions de suivi de régions que les méthodes de reconnaissance telle que celle décrite dans la section 4.1.

L'idéal serait de pouvoir utiliser l'intégralité du contenu des triangles pour effectuer le suivi à l'aide de la méthode décrite dans le chapitre 3. Dans l'éventualité où la zone ne serait pas planaire, le suivi basé sur l'estimation d'une homographie décrocherait et permettrait ainsi d'éliminer la zone. Le suivi fournissant directement une homographie utilisable en cas de zone planaire. Ceci pose deux problèmes. Le premier est que suivre ainsi la quasi totalité de l'image nécessite des ressources calculatoires importantes. Le second problème est que la méthode de suivi peut diverger lentement et fournir une solution erronée sans s'en apercevoir.

Dans [Shi 94], l'auteur explique qu'en prenant en compte une petite zone autour d'un point et en supposant un changement de point de vue limité entre deux images, cette zone peut être suivie en considérant une translation de l'image précédente à l'image suivante (au lieu de se référer à une image de référence constante). Cette comparaison effectuée sur une réfèrence variable permet d'utiliser un modèle de mouvement simple mais ne permet pas de vérifier la cohérence à long terme du suivi, d'où un risque de dérive de l'estimation. L'auteur propose une vérification de la cohérence par un modèle affiné 2D sur une image plus éloignée afin de minimiser le risque.

La profondeur des points du maillage est estimée dans le SLAM (voir section 4.4.3.5). À partir d'un certain déplacement, le SLAM est donc capable de donner une estimation de la position dans l'image du point du maillage, ce qui nous permettra d'avoir un deuxième critère de contrôle des dérives.

À chaque image correspond finalement de nouvelles coordonnées pour chaque point de $\mathbf{P}$. Ces nouvelles coordonnées sont la reprojection du même point 3D dans la nouvelle image. En cas de perte de point par le suivi, les triangles correspondants sont supprimés afin de ne conserver que des formes fermées. Notez que le maillage n'est pas recalculé à chaque image. Ansi, même si le maillage change dans l'image, le maillage 3D "virtuel" qui est la source de ces maillages dans les différentes images reste le même.

4.4.3.4 Calcul de l'homographie d'une région triangulaire

Pour faciliter les traitements ultérieurs, les triangles sont considérés séparément dans la phase de calcul de l'homographie ${}^{c2}\mathbf{H}_{c1}$. Chaque triangle de l'ensemble $\mathbf{T}$ va se voir attribuer une homographie ${}^{c2}\mathbf{H}_{c1}$ propre. Un triangle est formé par trois points

dont les coordonnées sont connues dans $\mathcal{I}$ et $\mathcal{I}'$. N'étant pas assuré de la planarité du triangle, les coordonnées des points à l'intérieur du triangle ne peuvent pas être transformées par interpolation d'une image à l'autre. L'homographie n'a donc pour seul support d'estimation que ces trois points. C'est insuffisant, les trois points non colinéaires ne contraignant que six des huit dimensions de l'homographie.

Dans [Hartley 01], il est démontré que connaissant la matrice fondamentale ${}^{c2}\mathbf{F}_{c1}$ reliant $\mathcal{I}$ et $\mathcal{I}'$, la matrice d'homographie ${}^{c2}\mathbf{H}_{c1}$ peut être obtenue à partir de la projection de 3 points dans les deux images. La matrice fondamentale contraint les degrés de liberté restants. Cette matrice fondamentale peut-être obtenue de deux manières.

Calcul de la matrice fondamentale La première manière qui vienne à l'esprit est de la calculer à partir des estimations en cours du SLAM. La transformation ${}^{c2}\mathbf{M}_{c1}$ étant connue ainsi que les paramètres intrinsèques de la caméra, ${}^{c2}\mathbf{F}_{c1}$ est aisément calculée. Le problème est que ${}^{c2}\mathbf{F}_{c1}$ sert à la construction de ${}^{c2}\mathbf{H}_{c1}$ qui elle-même sert de mesure au SLAM. Or l'un des principes de l'EKF est qu'il ne doit pas y avoir corrélation entre la mesure et la prédiction. Cette approche est donc inutilisable.

La seconde méthode tient dans la constatation que l'ensemble des points de $\mathbf{P}$ partagent la même matrice fondamentale entre deux vues. L'objectif est donc de trouver la matrice fondamentale qui respecte autant que faire se peut les reprojections des points entre les deux images. De multiples méthodes permettant d'atteindre cet objectif existent, la plus connue étant la méthode des 8 points [LonguetHiggins 81]. Une revue des méthodes et de leurs performances est effectuée dans [Salvi 01]. Ce dernier article recommande la méthode des LMedS (Least Median Square, explications en annexe A) qui calcule chaque possibilité avec l'algorithme des 8 points. Si le consensus est suffisamment important, la matrice fondamentale résultante est alors utilisée pour le calcul de l'homographie des zones triangulaires. La matrice ${}^{c2}\mathbf{F}_{c1}$ obtenue permet de déduire les épipoles ℓ et ℓ' dans les images $\mathcal{I}$ et $\mathcal{I}'$ respectivement :

$$\ell = ker({}^{c2}\mathbf{F}_{c1}) \tag{4.31}$$

$$\ell' = ker({}^{c2}\mathbf{F}_{c1}^{\mathsf{T}}) \tag{4.32}$$

La méthode de calcul de la matrice fondamentale étant basée sur le consensus, tous les points ne respectent pas forcément l'égalité ${}^{c2}\mathbf{p}_i^{\mathsf{T}}{}^{c2}\mathbf{F}_{c1}{}^{c1}\mathbf{p}_i = 0$. Les points ${}^{c2}\mathbf{p}_i$ sont donc transformés pour respecter cette égalité en les projetant orthogonalement sur leur droite épipolaire ${}^{\mathsf{T}c2}\mathbf{F}_{c1}{}^{c1}\mathbf{p_i}$ respective.

Calcul de l'homographie Connaissant ${}^{c2}\mathbf{F}_{c1}$, ℓ et ℓ', la matrice d'homographie peut être estimée [Hartley 01, LopezNicolas 05] par

$${}^{c2}\mathbf{H}_{c1} = [\ell']_{\times}{}^{c2}\mathbf{F}_{c1} - \ell(\mathbf{M}^{-1}b)^{\mathsf{T}} \tag{4.33}$$

où $\mathbf{M}$ est composée pour chaque ligne des coordonnées d'un des 3 ${}^{c1}\mathbf{p_i}$. b est quant à lui calculé pour chaque ligne b_i par :

$$b_i = \frac{({}^{c2}\mathbf{p}_i \times ([\ell']_\times {}^{c2}\mathbf{F}_{c1} {}^{c1}\mathbf{p}_i))^\intercal ({}^{c2}\mathbf{p}_i \times \ell')}{({}^{c2}\mathbf{p}_i \times \ell')^\intercal ({}^{c2}\mathbf{p}_i \times \ell')} \tag{4.34}$$

Dans le cas où le calcul de la matrice fondamentale pose problème, par exemple dans des configurations singulières, une autre solution est décrite dans [LopezNicolas 05]. Si une zone est déja initialisée et suivie par notre algorithme, une homographie dans l'image est connue. L'article propose de remplacer la matrice fondamentale pour compenser les dimensions non contraintes par les 3 points par cette dernière matrice d'homographie.

L'homographie mesurée permet de reprojeter les 3 points à travers une autre homographie et donc un autre plan. Chaque point a donc deux projections dans la deuxième image qui sont reliés par une ligne. L'intersection de ces 3 lignes donne ℓ' qui sert de quatrième point pour calculer l'homographie classiquement (ex. Direct Linear Transform [Hartley 01]).

Chaque triangle possède maintenant sa propre homographie ${}^{c2}\mathbf{H}_{c1}$ reliant sa projection dans $\mathcal{I}$ à sa projection dans $\mathcal{I}'$. Cette homographie peut être mal calculée pour les triangles avec des angles très obtus. Ce type de triangle est donc éliminé afin de ne pas provoquer d'erreurs dans le regroupement des triangles.

4.4.3.5 Utilisation des triangles

L'intérêt du triangle est qu'on peut affirmer, sans aucun doute, que ses trois angles sont coplanaires. Sans même avoir de plans dans la scène, cette méthode permet d'utiliser l'estimateur décrit dans cette thèse avec des mesures d'homographies. Bien sûr, l'objectif est tout de même de trouver des zones planaires pour utiliser la méthode de suivi principale qui est beaucoup plus stable et robuste qu'un suivi de point classique.

Lors de la création du maillage, une copie de la caméra est ajoutée au vecteur d'état pour constituer la caméra de référence $\mathcal{F}_{c1}$. Sont enregistrées en mémoire en tant que constantes les coordonnées dans $\mathcal{I}$ des points constituant au moins un triangle complet (les points de $\mathbf{P}$). Sont enregistrés également les triangles avec les index des points constituant. À chaque point $\mathbf{p_i}$ est attribué une profondeur dont l'inverse est enregistré dans le vecteur d'état. La méthode d'initialisation décrite dans la section 4.3 est en effet utilisée pour appréhender les mesures d'homographie fournies par les triangles.

Chaque triangle est ainsi représenté par trois scalaires, soit la même taille qu'un point dans le SLAM monoculaire classique. De plus, les points peuvent être (et le sont souvent) partagés par plusieurs triangles, ce qui réduit d'autant plus le nombre de paramètres moyen à estimer. Lorsque les trois profondeurs d'un même triangle sont

connus avec suffisamment peu d'incertitude, la représentation sous forme de plans est utilisée. Ce plan est comparé statistiquement à ceux des triangles qui partagent une face avec celui en cours.

Les triangles dont les trois voisins sont différents (angle entre les normales et distance à l'origine supérieurs à un seuil) sont éliminés. Les triangles partageant un même plan (et reliés par une face à au moins un autre triangle de cet ensemble) sont regroupés ensemble dans une même forme géométrique qui servira de zone à suivre pour notre tracker d'homographie. Un seul plan est conservé parmi tous ceux similaires dans le vecteur d'état : celui dont la distance moyenne aux autres plans est minimale est choisi.

4.4.4 Résultats

En utilisant la même séquence de test que dans la section sur l'initialisation des plans 4.3, l'extraction des plans peut être illustrée. La figure 4.23 illustre le maillage créé à partir des points suivis. Après quelques images, la convergence est visible. Pour faciliter la visualisation, la carte finale, à la 300ème itération a été choisie pour montrer le bon fonctionnement. Cette carte est visible selon 3 vues différentes dans la figure 4.24. Cette figure permet de visualiser les 3 plans correctement reconstruits hormis sur certains triangles dont les points sont sur plusieurs plans (visible nettement sur la figure du milieu). Dans ces résultats, les homographies sont calculées sans *a priori* en utilisant la matrice fondamentale, estimée à partir des points, comme support supplémentaire aux triangles. La figure 4.25 démontre que la pose est effectivement calculée en même temps que l'extraction des plans, toujours bien sûr à un facteur d'échelle près.

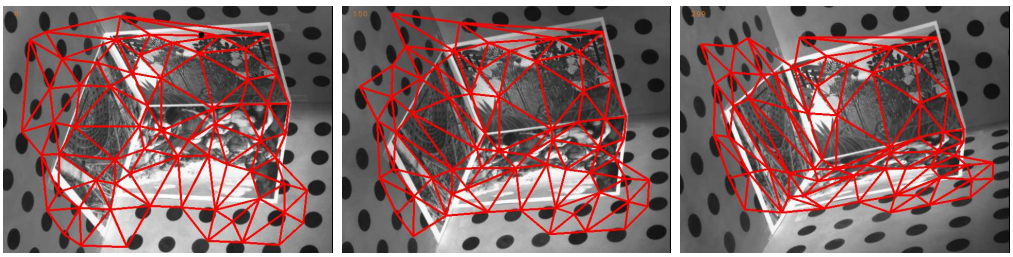


FIGURE 4.23 – Maillage généré par la triangulation de Delaunay sur des points de Harris sur la séquence de test

Les résultats suivants illustrent, sur la même séquence, le regroupement des triangles pour former les zones appartenant à un même plan. La première figure 4.26 montre l'évolution de l'estimation de la profondeur d'un des points d'un triangle. Sont affichées également dans cette figure les bornes minimales et maximales correspondant à l'intervalle de confiance de l'estimation (à 90 % de confiance, soit 2.7σ environ).

Considérons ici que la profondeur d'un point a convergé vers la solution lorsque l'intervalle de confiance à 90% est inférieur à 3 centimètres. Dans la figure 4.27, les triangles bleus sont ceux dont les trois points ont convergé. Les images correspondent aux itérations 14 à 22 de la séquence, ce qui confirme la rapidité de convergence illustrée

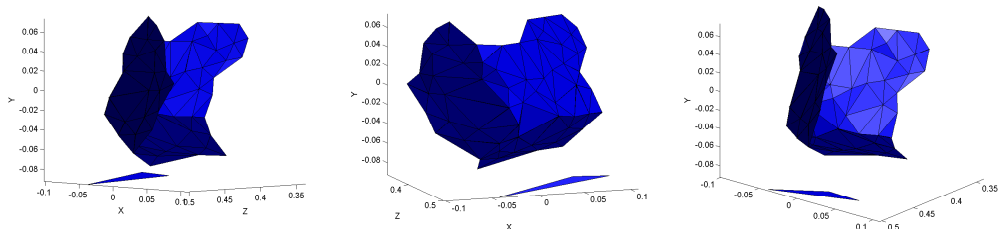


FIGURE 4.24 – Visualisation de la carte estimée à l'itération 300 sans a priori sur la scène. Les homographies sont estimées à l'aide de la méthode basée sur la matrice fondamentale.

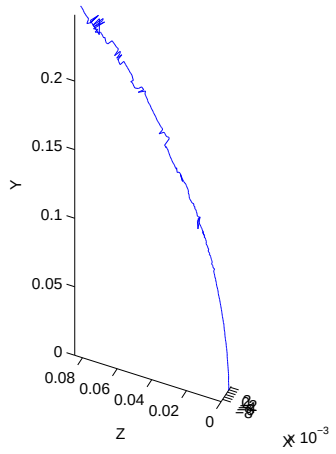


FIGURE 4.25 – Position de la caméra estimée pendant l'extraction des plans

dans la figure 4.26.

Pour clore ces résultats, la figure 4.28 montre la fusion des triangles à l'itération 26. Chaque contour d'une couleur différente représente un plan différent.

4.4.5 Conclusion sur l'extraction des plans

Cette section présentait une méthode permettant d'extraire, à partir d'une séquence d'images, des zones planaires utilisables dans notre SLAM. L'intéret de cette méthode par rapport à l'existant est qu'elle permet une utilisation immédiate des mesures images dans le SLAM sans attendre une translation suffisante pour l'estimation des plans. Basée sur des méthodes éprouvées, son concept est néanmoins original puisque l'extraction des plans se fait parallèlement à l'utilisation des mêmes mesures pour la localisation de la caméra. Cette suppression de la latence permet de prendre en compte des zones qui ne restent pas suffisamment longtemps dans le champ de vision de la caméra mais dont le mouvement apporte tout de même un indice quant aux déplacement du système.

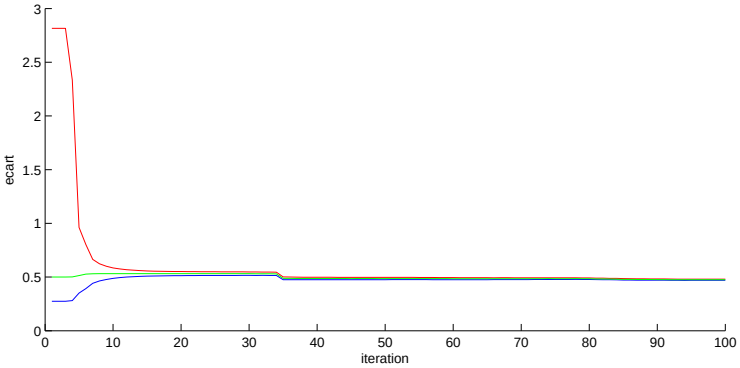


FIGURE 4.26 – Estimation de la profondeur d'un des points d'un triangle pendant l'extraction des plans. En vert est affichée la moyenne de l'estimation, en rouge et bleu les bornes minimales et maximales avec une confiance de 90%.

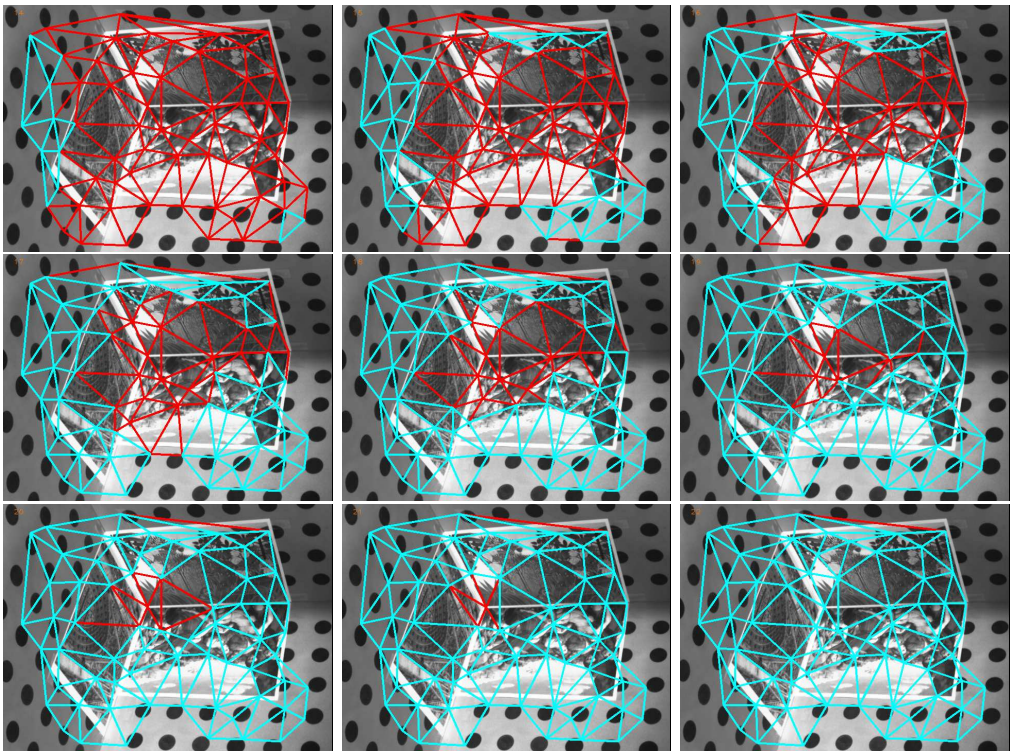


FIGURE 4.27 – Convergence des triangles lors de la séquence. Les triangles bleus sont les triangles dont les 3 profondeurs sont connues avec une incertitude suffisamment réduite.

L'intérêt d'utiliser des plans dans ce cas où aucune information *a priori* n'est connue est double. Le premier intérêt est de pouvoir compléter les plans existants (extraits

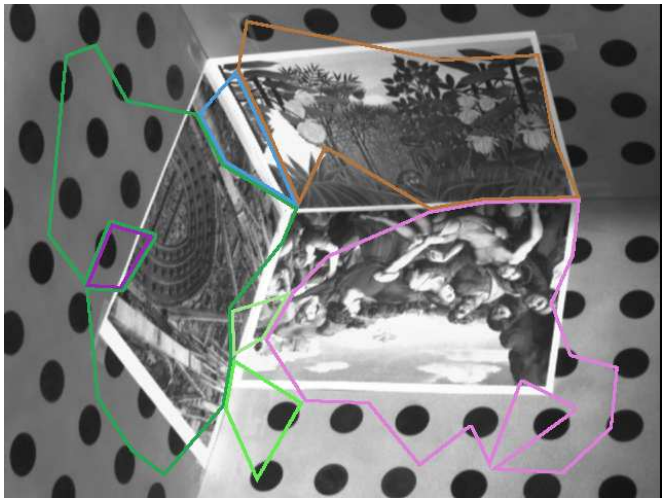


FIGURE 4.28 – Regroupement des triangles par plan à la 26ème itération. Les contours de chaque groupe de triangles sont affichés.

à l'aide de la reconnaissance de régions et/ou d'une base de données de plans) dans les intervalles où les mesures manquent, tout en gardant une cohérence dans la carte et le principe. Le second intérêt tient dans la comparaison vis à vis de l'utilisation de simples points (SLAM monoculaire classique). En effet, l'utilisation de notre méthode rend, cela a été démontré précédemment, la carte plus compacte. De plus, la mesure de l'homographie fournit beaucoup plus d'informations sur la structure de la scène et le mouvement 3D qu'un simple déplacement de points. Alors que les méthodes basées points constituent les liens dans la carte par corrélations lors de la mise à jour du SLAM, cette méthode permet de mesurer le déplacement mutuel de plusieurs points en quelque sorte. La fusion des plans similaires permet de créer une structure encore plus cohérente de la scène en mutualisant l'information des mesures.

4.5 Conclusion sur l'ajout

À travers l'utilisation des 3 méthodes présentées dans ce chapitre, ces travaux ont permis de prouver le potentiel des plans dans le contexte de l'EKF-SLAM. Tout comme les points, les plans peuvent non seulement servir de support de mesure, mais également être utilisés sans *a priori* obligatoires – et sans latence – sur leurs paramètres. Les méthodes présentées ne sont pas trop gourmandes en terme de coût calculatoire pour être utilisées en temps réel et peuvent donc être utilisées dans notre contexte de mobilité. D'autant plus que ces méthodes ne sont pas utilisées à chaque image mais uniquement quand le manque d'information les rendent indispensables.

Chapitre 5

Fusion avec un capteur inertiel

Nous avons exposé dans les chapitres précédents les nombreuses contraintes et problématiques liées à l'utilisation d'une caméra. Nous avons rappelé les modèles mathématiques simplifiés nécessaires pour représenter la caméra, son déplacement et structurer son environnement. Ces contraintes et simplifications peuvent provoquer des approximations dans la pose de la caméra et au pire une divergence dans le calcul dans les cas difficiles. L'exemple le plus simple étant la perte de toute information visuelle à cause d'une occlusion. On souhaite donc pouvoir compléter et compenser les informations du capteur caméra par un système de mesure qui ne possède pas les mêmes propriétés et donc qui se base sur d'autres caractéristiques de son environnement que l'information lumineuse. Les capteurs inertiels sont une solution à ce problème.

Les capteurs inertiels sont des capteurs proprioceptifs : leurs mesures ne dépendent pas directement de leur environnement. Celles-ci sont fournies par le changement d'état du capteur lui-même. On a donc une information supplémentaire décorrélée des mesures caméra qui va nous permettre d'avoir une estimation sans discontinuité du mouvement du système. Parce qu'elle permet de s'assurer de l'indépendance des mesures – et donc de séparer dans le temps les mises à jour de l'état – la non corrélation des mesures va faciliter leur fusion. Cette dernière va contraindre l'estimation dans un espace de recherche plus réduit [1].

Ces capteurs fournissent une mesure du déplacement du système qui pourra être utilisée pour estimer à court terme la nouvelle pose de la caméra. Nous montrerons dans ce chapitre que les améliorations de l'estimation de l'état du système sont nombreuses :

1. l'intégration des données inertielles doit permettre de compenser le manque d'information visuelle liée :
   – à un mouvement brusque (flou de bougé)
   – une occlusion temporaire des objets suivis.
2. au niveau de l'image, l'estimation du déplacement doit permettre de réduire l'incertitude de la pose et donc de la prédiction des mesures. Cela permet donc de faciliter :

1. Les variances de l'état étant diminuées par la multiplication des mesures indépendantes, l'espace des poses possibles est réduit.

- la recherche dans l'image suivante de l'élément suivi.
- l'élimination des mesures images aberrantes.

3. Au niveau de l'estimation statistique de l'état, l'ajout de la mesure directe du déplacement doit permettre :
   - de réduire le problème de visibilité (chapitre slam) du SLAM.
   - de réduire le problème de facteur d'échelle.
   - de réduire les problèmes de linéarisation liés à l'estimation par EKF.

Comme tout capteur, les inertiels possèdent des inconvénients. Nous étudierons pourquoi ces capteurs doivent être utilisés en collaboration avec une autre source d'informations et les solutions que nous avons apportées aux problèmes d'implémentation et de calibrations géométriques et temporelles nécessaires au bon fonctionnement de la fusion multi-capteurs.

5.1 Principes de fonctionnement des IMU

Les capteurs inertiels peuvent être de différents types. Cette thèse s'intéressera uniquement aux gyroscopes et accéléromètres de type MEMS. D'autres types de capteurs inertiels existent mais ceux-ci sont les seuls à la fois abordables d'un point de vue financier et suffisamment compacts disponibles sur le marché. D'une manière générale, MEMS est un acronyme pour *Micro Electro Mechanical Systems*. Cet acronyme désigne les capteurs et actionneurs de taille microscopique. Ils sont souvent moins précis que leurs équivalents mécaniques ou optiques mais sont beaucoup moins chers à fabriquer. Bien que les explications qui suivent soient générales, il est possible que certaines soient uniquement valides pour les capteurs de type MEMS.

5.1.1 L'accéléromètre

Un accéléromètre mesure comme son nom l'indique sa propre accélération sur un axe donné. Schématiquement, un accéléromètre est constitué d'un poids de masse μ et d'un ressort. Sous l'effet de l'inertie, le poids va se déplacer par rapport à sa position initiale si l'objet auquel il est attaché subit une accélération. Le ressort imprime alors une force opposée pour conserver en place le poids. Si on considère le poids comme une masse ponctuelle, on peut appliquer la deuxième loi de Newton :

$$\mu^i \mathbf{a} \quad = \quad \vec{F_r} + \vec{F_o} + \mu \vec{G} \tag{5.1}$$

où ${}^i a$ est l'accélération à proprement parler de l'objet, F_r est la force imprimée par le ressort liée au mouvement du système, F_o est l'ensemble des forces qui bruitent le système comme le frottement et G est la force induite par la force de gravité universelle. On note que si le système ne bouge pas, la force de compensation du ressort est l'opposée de la force de gravité universelle et l'accélération est nulle, ce qui est bien le résultat attendu.

En négligeant les bruits $\vec{F_o}$, cette équation peut être réécrite par :

$$f = \frac{1}{\mu}\vec{F_r} \tag{5.2}$$

$$= {}^{i}\mathbf{a} - {}^{i}\vec{G} \tag{5.3}$$

où f est la force spécifique mesurée par l'accéléromètre (la masse μ étant constante et connue). Cette représentation est bien sûr valide dans le cas où la mesure est parfaite. Dans la réalité, cette mesure est entachée d'une erreur qui varie avec le temps et la température. Cette erreur peut être modélisée comme un bruit gaussien. Ceci peut être modélisé par la somme d'un biais $\mathbf{b}_{acc}$ et un bruit blanc gaussien $\mathbf{n}_{acc}$. Si le facteur d'échelle liée à la conversion du signal électrique en signal numérique est négligé, la mesure peut être modélisée $\mathbf{h}$ comme

$$\mathbf{h}_{acc} = \mathbf{f} + \mathbf{b}_{acc} + \mathbf{n}_{acc} \tag{5.4}$$

La plupart des capteurs inertiels utilisent 3 accéléromètres placés de manière perpendiculaire afin d'avoir une accélération pour chacun. L'orthogonalité du montage n'étant pas parfaite, cela implique une erreur dans la mesure. Toutefois, cette information d'orthogonalité est calibrée en usine et est donc corrigée électroniquement par l'IMU. L'information d'accélération est donc fournie par le capteur inertiel sous forme d'un vecteur de 3 éléments qui indique la direction de l'accélération et la norme sa grandeur (exprimée en ms^{-2}). Ce vecteur contient la force de gravité. Il est nécessaire de retirer cette force de $\mathbf{h}_{acc}$ si l'on souhaite connaitre l'accélération réelle du système.

5.1.2 Le gyroscope

Le gyroscope (en grec, *gyro* signifie 'tour') mesure la vitesse de rotation autour d'un axe d'un objet. Le principe d'un gyroscope MEMS est basé sur celui de la force de Coriolis. Lorsque l'on applique une rotation à un objet en déplacement, une accélération supplémentaire perpendiculaire au mouvement se produit : l'accélération de Coriolis ${}^{cor}\mathbf{a}$. Cette force peut être modélisée par l'équation

$${}^{cor}\mathbf{a} = -2\mu(\boldsymbol{\omega} \times \mathbf{v}) \tag{5.5}$$

où μ est la masse du poids, $\boldsymbol{\omega}$ est la vitesse de rotation et $\mathbf{v}$ la vitesse de déplacement. Si on applique un mouvement vibratoire à un poids et que l'on mesure ce même phénomène vibratoire perpendiculairement, on dispose de toutes les données pour estimer directement la vitesse de rotation sur l'axe. Tout comme l'accéléromètre, 3 gyroscopes sont disposés orthogonalement pour mesurer la vitesse de rotation 3D. Cette vitesse de rotation $\boldsymbol{\omega}$ est exprimée sous la forme $\theta\mathbf{u}$. Ce capteur est également entaché d'une erreur qui peut être modélisée de manière similaire à celle de l'accéléromètre.

$$\mathbf{h}_{gyr} = \boldsymbol{\omega} + \mathbf{b}_{gyr} + \mathbf{n}_{gyr} \tag{5.6}$$

5.2 Description de la plate-forme

Un IMU (Inertial Measurement Unit) est un système regroupant plusieurs capteurs inertiels et disposant de programmes cablés permettant de prétraiter les réponses électriques des capteurs. Dans le cadre de cette thèse, nous avons utilisé la plate forme XSENS MTx (Figure 5.1 - http ://www.xsens.com). Cet IMU contient 3 accéléromètres, 3 gyroscopes et un magnétomètre [2]. Il peut retourner soit les informations corrigées de ces 3 capteurs, soit retourner une information d'orientation absolue (Dans le repère terrestre) estimée par l'électronique de l'IMU à partir de ses capteurs. L'information d'orientation absolue possédant de nombreux défauts (notamment un décalage temporel variable et un dysfonctionnement dans les environnements métalliques), on s'oriente vers l'utilisation directe des informations inertielles. L'interface entre l'IMU et l'ordinateur se fait par une liaison USB 2.0 (Universal Serial Bus).

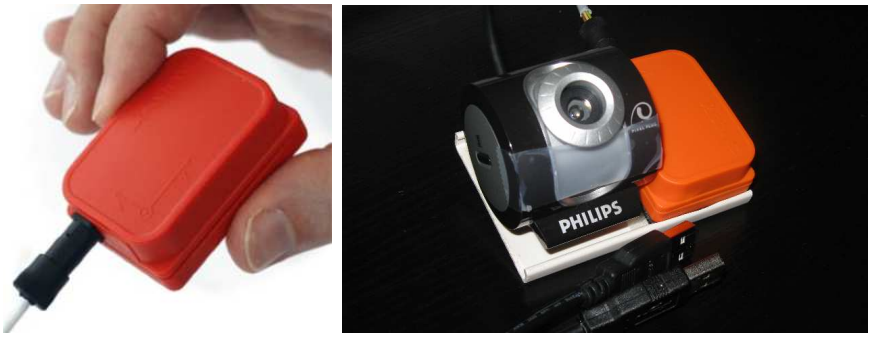


FIGURE 5.1 – Vues du système de capture comprenant une caméra USB et une centrale inertielle

À travers le pilote, le logiciel envoie une demande au capteur. Cette demande est bloquante (le programme attend la réponse avant de passer à l'instruction suivante). Le capteur met à disposition une nouvelle mesure à une fréquence de 100hz sous la forme d'un tableau de données :

count	accx	accy	accz	gyr1	gyr2	gyr3	mag1	mag2	mag3	tmp

où $acc*$ sont les composantes du vecteur d'accélération, $gyr*$ sont les composantes du vecteur de vitesse de rotation, $mag*$ le vecteur indiquant le nord magnétique et *tmp* étant la température du système. *count* est un compteur qui indique l'identifiant de la mesure. Ce compteur n'étant défini que sur 2 octets, il boucle à la 2^{16}ème itération et ne peut donc servir que d'indication relative. Nous l'utilisons simplement pour vérifier que notre processus tourne suffisamment vite pour traiter les données à la fréquence des capteurs inertiels. Les données magnétiques et de température sont ignorées par notre algorithme. La figure 5.2 montre un exemple de sortie fournie par l'interface matlab du driver.

2. Un magnétomètre est un capteur mesurant le nord magnétique. Il renvoie un vecteur de rotation qui définit l'orientation 3D absolue du capteur dans le repère magnétique terrestre.

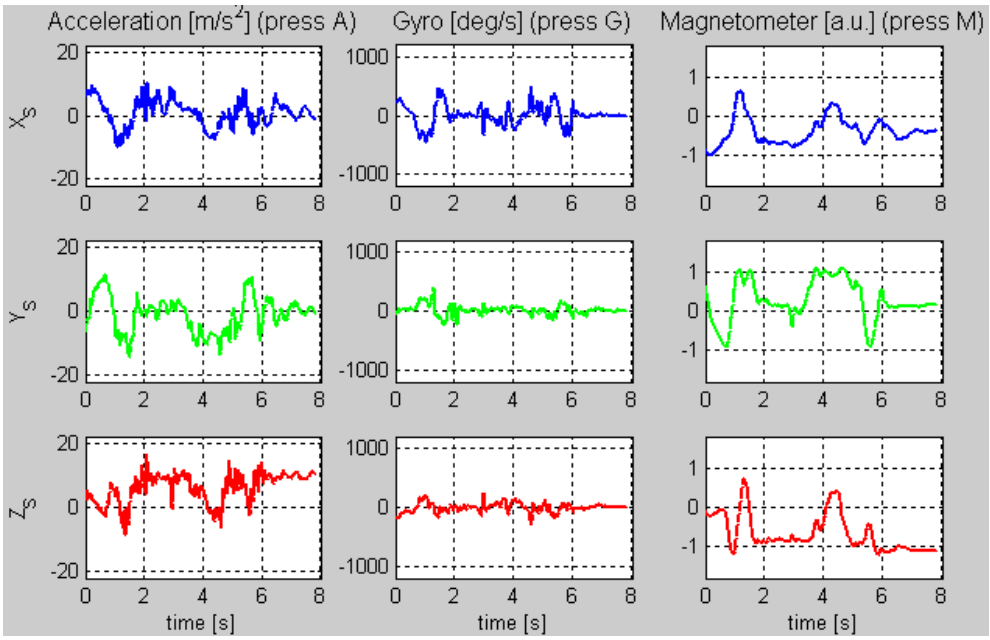


FIGURE 5.2 – Exemple de sortie du capteur obtenue avec l'interface Matlab du kit de développement Xsens

L'incertitude des capteurs inertiels. Pourquoi ne pas utiliser uniquement les capteurs inertiels et vouloir fusionner avec des données images ? La première raison est que leur mesure est biaisée. Cela signifie que même en filtrant le bruit Gaussien on conservera une erreur. Cette erreur étant dépendante d'un certain nombre de paramètres d'état du capteur, elle ne peut être estimée à priori et change en fonction du temps. Pour obtenir une position ${}^{i}\mathbf{t}$ et une orientation ${}^{i}\mathbf{r}$ à partir des données des capteurs inertiels, il est nécessaire d'intégrer leurs mesures dans le temps :

$$ {}^{i}\mathbf{t} = \int\int (\mathbf{h}_{acc} + {}^{i}\mathbf{g})\frac{dt}{2}^{2} \tag{5.7} $$

$$ {}^{i}\mathbf{r} = \int \mathbf{h}_{gyr} dt \tag{5.8} $$

Cette intégration se faisant à 100hz, même un biais faible fait rapidement diverger l'estimation. De plus, on note que dans le cas de la position, nous sommes obligés de rajouter le vecteur gravité défini dans le repère du capteur afin d'obtenir l'accélération propre. L'orientation doit être parfaitement connue afin d'obtenir le vecteur gravité dans le repère voulu. Le vecteur gravité ayant une norme importante ($9.81 ms^{-2}$), même une faible erreur dans l'estimation de l'orientation provoque une erreur non négligeable dans le calcul de ${}^{i}\mathbf{t}$.

Sur une courte période cependant les informations fournies sont suffisantes pour obtenir un déplacement avec une précision correcte. La fusion avec les données image

permet d'estimer le biais et de compenser celui-ci au fur et à mesure de la séquence.

La première expérience effectuée consiste à placer le capteur inertiel sur une table dans aucun mouvement et à analyser les données renvoyées par ce capteur. On estime que les vibrations de l'environnement sont négligeables. L'acquisition a duré 1 minute 30 secondes, soit environ 9000 mesures. Afin de visualiser ces données, nous avons établi un histogramme des mesures. Pour les deux capteurs, l'abscisse correspond à la norme des vecteurs mesurés et l'ordonnée correspond au nombre d'occurences normalisées de manière à ce que l'intégrale de l'histogramme soit égal à 1. Sur la figure 5.3 les points correspondent à cet histogramme. On a ensuite calculé la moyenne et la variance des données mesurées et dessiné la fonction de densité de probabilité normale correspondant à ces moments. Cette fonction de densité de probabilité (nommée pdf pour *Probability Density Function*) apparait en traits continus sur la même figure. Pour simplifier la présentation des résultats, on ne s'intéresse pas à chaque composante du mouvement mais à la norme euclidienne de l'accélération et de la vitesse de rotation.

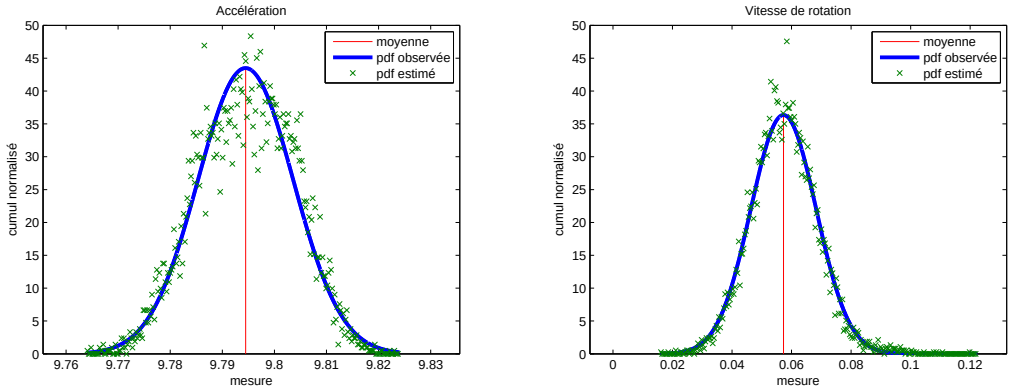


FIGURE 5.3 – Analyse statistique des données capteurs sans mouvement

On peut observer que l'histogramme est très similaire à la pdf estimée, ce qui permet d'avancer que le choix d'une modélisation du bruit par une variable aléatoire gaussienne est correct. Pour l'accélération, la pdf est définie par $\mathcal{N}(9.79, 9.210^{-3})$ et la vitesse de rotation par $\mathcal{N}(5.710^{-2}, 1.110^{-2})$. La moyenne de l'accélération est comme prévu proche de la constante gravitationnelle.

On observe un biais dans les 2 mesures. Le biais peut paraître négligeable mais une fois intégré dans le temps cela donne par exemple pour la vitesse de rotation une erreur d'estimation de 0.5 radians en 10 secondes.

5.3 Fusion Caméra-Capteurs

5.3.1 Problématique

Capteurs inertiels et caméra fournissent des informations différentes, à des instants différents, à une fréquence différente, par des moyens de mesures différents mais permettent pourtant d'obtenir au final une information similaire : Le changement de pose de la caméra. Si on est capable d'intégrer les informations parrallèlement, on dispose en quelque sorte de deux preuves de l'état du système et on améliore donc son estimation. Lorsque l'on utilise ensemble deux mesures différentes dans un même estimateur statistique, on parle alors de fusion. Cette fusion doit bien entendu être faite de manière à respecter les contraintes intrinsèques de l'estimateur.

Une littérature relativement abondante existe sur la fusion vision/inertiel. La plupart de ces articles utilisent une approche similaire à la nôtre dans le cadre d'un filtre de Kalman [Klein 04, You 99, Azuma 99] voire même d'un système de vision SLAM [Gemeiner 07, Pinies 07]. La différence tient dans le cadre applicatif et surtout dans le modèle de caméra utilisé. On trouve également des articles qui profitent de l'estimation du mouvement pour compenser les problèmes images de flou de mouvement [Klein 04]. On citera également pour référence les méthodes analytiques de fusion vision/inertielle [Baldwin 09] qui proposent une solution de fusion sans utiliser d'estimateur statistique.

On sait que les mesures des deux capteurs ne sont pas corrélées et peuvent en conséquence être utilisées séparément en utilisant deux mises à jour différentes sans perdre d'information. La problématique principale de cette fusion tient dans la caractéristique temporelle de l'EKF. En effet, ce filtre est un processus de type Markovien. Le futur dépend uniquement du présent et non du passé. Chaque mise à jour doit être faite sur une mesure postérieure à celle de la mise à jour précédente. Il est donc primordial de synchroniser les deux systèmes de capture pour compenser les différences de délai d'acquisition et s'assurer de la cohérence temporelle. Cette synchronisation sera expliquée dans la section 5.5. Cette nécessité de cohérence est renforcée par le modèle d'évolution à accélération constante utilisé. En effet, entre chaque mesure l'état est prédit en fonction du temps écoulé depuis la dernière mesure. Pour conserver l'efficacité de cette prédiction, il est nécessaire d'avoir une échelle de temps utilisable pour les deux capteurs. On considère par la suite que les deux capteurs sont d'ores et déja synchronisés en attendant une explication dans une section ultérieure.

5.3.2 Modélisation du gyroscope

Rappelons que le modèle de mesure doit être capable de donner une estimation de la mesure à comparer à la mesure réelle pour mettre à jour le vecteur d'état. Le modèle de mesure effectue son estimation à partir des données dont il dispose à priori : variables d'état et constantes. L'équation (5.6) définit la manière dont peut être statistiquement décomposée la mesure provenant du gyroscope. Si on enlève le bruit Gaussien qui est

déterminé par le bruit du modèle de mesure, il nous reste à estimer la mesure du gyroscope selon :

$$\mathbf{h}_{gyr} = {}^{i}\boldsymbol{\omega} + \mathbf{b}_{gyr} \tag{5.9}$$

La vitesse de rotation d'un solide étant la même en tout point de celui-ci, le capteur et la caméra étant solidement liés, la vitesse de rotation de la caméra correspond à celle du capteur inertiel. Les deux n'étant cependant pas représentées dans le même repère, on doit transformer la vitesse de rotation de la caméra dans le repère du capteur :

$$\mathbf{h}_{gyr} = {}^{i}\mathbf{R}_{c}\,{}^{c}\boldsymbol{\omega} + \mathbf{b}_{gyr} \tag{5.10}$$

Nous verrons par la suite comment obtenir ${}^{i}\mathbf{R}_{c}$ et $\mathbf{b}_{gyr}$. On voit que notre état permet d'estimer directement la mesure du gyroscope de manière simple avec un modèle relativement linéaire.

5.3.3 Modélisation de l'accéléromètre

Le modèle de mesure de l'accéléromètre est plus compliqué. On considère toujours le bruit gaussien comme déterminé par le bruit du modèle de mesure. On déduit alors de (5.4) :

$$\begin{aligned}
\mathbf{h}_{acc} &= {}^{i}\mathbf{a} - {}^{i}\mathbf{g} + \mathbf{b}_{acc} && (5.11)\\
&= {}^{i}\mathbf{a} - {}^{i}\mathbf{R}_{c}\,{}^{c}\mathbf{g} + \mathbf{b}_{acc} && (5.12)\\
&= {}^{i}\mathbf{a} - {}^{i}\mathbf{R}_{c}R({}^{c}\mathbf{r}_{o})\,{}^{o}\mathbf{g} + \mathbf{b}_{acc} && (5.13)
\end{aligned}$$

avec ${}^{o}\mathbf{g}$ le vecteur de gravité universelle

$${}^{o}\mathbf{g} = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} \tag{5.14}$$

Les deux systèmes étant reliés rigidement, l'accélération subie par la caméra est également subie par le capteur. Cependant, à cette accélération linéaire s'ajoute l'accélération centripète. Cette dernière composante est liée à la vitesse de rotation de la caméra. En effet, le capteur étant éloigné du centre de rotation instantané, il subit une accélération supplémentaire

$$\mathbf{h}_{acc} = {}^{i}\mathbf{R}_{c}({}^{c}\mathbf{a} + [{}^{c}\boldsymbol{\omega}]_{\times}\,{}^{c}\mathbf{v} - R({}^{c}\mathbf{r}_{o})\,{}^{o}\mathbf{g}) + \mathbf{b}_{acc} \tag{5.15}$$

5.3.4 Modification du vecteur d'état

Les biais $\mathbf{b}_{gyr}$ et $\mathbf{b}_{acc}$ évoluent au cours du temps et ne peuvent pas être estimés hors ligne. Ces inconnues sont ajoutées au vecteur d'état afin d'être estimées en même temps que les autres paramètres. On modifie pour cela le vecteur $\mathbf{x_c}$ qui contient désormais

$$\mathbf{x_c} = \begin{bmatrix} \mathbf{t} & \mathbf{r} & \mathbf{v} & \boldsymbol{\omega} & \mathbf{a} & \mathbf{b}_{gyr} & \mathbf{b}_{acc} \end{bmatrix}^{\mathtt{T}} \tag{5.16}$$

On estime qu'en moyenne les biais sont nuls et peuvent être initialisés trivialement au démarrage du programme. Cette moyenne s'établit sur l'ensemble des exécutions. Leurs variances sont définies de telle manière qu'elles acceptent les changements de biais les plus forts trouvés pendant nos tests à 2σ (aux alentours de 10^{-4}). On peux observer aux vues des modèles de mesures utilisés pour le gyroscope et l'accéléromètre que notre choix de représentation de la caméra est parfaitement adapté à notre problématique puisqu'il contient exactement les informations nécessaires à la prédiction et à l'utilisation correcte de la mesure inertielle (notamment la corrélation entre la vitesse de rotation et la position). Le modèle de prédiction du déplacement de la caméra (chapitre 2.2) devient :

$$f(\mathbf{x_c}, \delta t) = \begin{bmatrix} f_{\mathbf{t}}(\mathbf{x_c}, \delta t) \\ f_{\mathbf{r}}(\mathbf{x_c}, \delta t) \\ f_{\mathbf{v}}(\mathbf{x_c}, \delta t) \\ f_{\boldsymbol{\omega}}(\mathbf{x_c}, \delta t) \\ f_{\mathbf{a}}(\mathbf{x_c}, \delta t) \\ \mathbf{b}_{gyr} \\ \mathbf{b}_{acc} \end{bmatrix} \tag{5.17}$$

5.4 La calibration caméra-capteurs

On a pu voir dans la section précédente que la fusion des données caméra et des données capteur nécessitaient de replacer les données dans le même repère. Cette transformation nécessite donc la connaissance de la matrice ${}^{i}\mathbf{R}_{c}$. On considère que la caméra et le capteur sont solidaires et que ${}^{i}\mathbf{R}_{c}$ est donc invariante. Il est donc possible d'estimer hors ligne cette transformation.

Pour cela, des informations provenant de la caméra et des informations provenant de l'IMU doivent être recalées. L'idée est de comparer le déplacement entre 2 poses mesurées chacune par les deux sources. Afin de rendre le calcul plus robuste, on souhaite pouvoir calculer le déplacement pour chaque couple de poses. Il est donc nécessaire que le capteur fournisse une information absolue. Deux solutions sont possibles : soit utiliser la sortie d'orientation du capteur, soit utiliser l'accéléromètre qui mesure la direction du centre de gravité terrestre dans le cas où il n'y a pas de déplacement. Les tests ont déterminé que cette deuxième solution apportait les résultats les plus précis.

La matrice de rotation ${}^{i}\mathbf{R}_{c}$ peut être comparée à la matrice de rotation entre une caméra et un repère du robot qui la tient. On utilise donc la méthode présentée dans

[Tsai 89] pour calibrer cette rotation (Algorithme 5.4). Les mesures de la gravité étant assez bruitées, on utilisera au moins une dizaine de poses bien distinctes pour obtenir une calibration de qualité.

5.5 La synchronisation caméra-capteurs

Dans les systèmes classiques de fusion de capteurs, la synchronisation est faite matériellement par un système de déclenchement à horloge tierce. Celui-ci envoie à intervalles réguliers et prédéterminés une demande à chaque capteur pour qu'il acquière une nouvelle donnée et appose une information temporelle à ces données. De cette manière, on connait l'intervalle entre chaque mesure et on peut donc les fusionner correctement. Dans le cadre de cette thèse, où l'on souhaite utiliser une caméra intégrée, il est impossible de connecter ce genre de système d'horloge. De plus, on a vu que le capteur inertiel ne fournit pas d'information sur le moment de l'acquisition. La caméra USB, contrairement à une caméra FireWire, ne délivre pas non plus de timestamp. Ce manque d'information temporelle est d'autant plus délicat que la fréquence d'acquisition des données est variable : la fréquence est respectée en moyenne sur un certain nombre de captures, mais pas individuellement (voir fig. 5.5)

A condition d'utiliser un timer dont le grain est inférieur à la fréquence d'acquisition du capteur le plus rapide (en l'occurence 100hz pour l'IMU), on peut mesurer au niveau du système hôte le moment de l'acquisition de la mesure. Cependant, cela ne mesure pas l'instant de la mesure mais l'instant où la donnée arrive sur le système hôte, le délai d'acquisition variant en fonction du capteur, des capacités du système de transfert et de la charge du système hôte. En supposant que ce délai pour chaque capteur ne varie pas ou peu au cours du temps, le problème est donc de déterminer les délais d'acquisition du capteur et de la caméra. Ceci afin de pouvoir corriger les mesures temporelles.

Une solution approximative Il est impossible de mesurer les délais d'acquisition. Cependant, il est possible de mesurer la différence de délai d'acquisition entre le capteur et la caméra. Ceci est suffisant pour notre problème puisqu'on s'intéresse aux intervalles de temps. Une solution pour calculer ce délai grossièrement est de capturer une séquence avec une mire et de n'effectuer qu'un mouvement brusque entre 2 périodes sans mouvement. En calculant la pose à partir de la mire, on cherche la première image où la pose change et on fait de même avec les informations inertielles du capteur. Puisque chaque donnée image et chaque donnée capteur ont une valeur temporelle sur le système hôte, la différence de délai d'acquisition est la différence de temps entre les captures sélectionnées. On ne pourra cependant pas être plus précis que la fréquence de capture la plus basse. Cela permet cependant de réduire l'espace de recherche et de connaitre en nombre d'images la différence de délai d'acquisition.

Une solution plus précise. Comme nous avons pu l'expliquer auparavant, un des avantages de notre modèle de prédiction de mouvement de la caméra est qu'il fournit des informations directement comparables aux données du capteur. Considérons une

Calibration de la rotation

Soit $\mathcal{F}_g$ le repère définissant le centre de gravité terrestre. Considérons n mesures de la gravité ${}^{i}\mathbf{a}_{(x)}$ et n mesures de la rotation de la caméra ${}^{c(x)}\mathbf{R}_o$ où $x = [1..n]$. Considérons la fonction $rvec(a, b)$ qui prend en paramètres 2 vecteurs et qui renvoie la rotation entre ces deux vecteurs ${}^{b}\mathbf{R}_a$.

$$
\begin{aligned}
{}^{g}\mathbf{R}_o &= {}^{g}\mathbf{R}_i {}^{i}\mathbf{R}_c {}^{c}\mathbf{R}_o && (5.18)\\
{}^{i}\mathbf{R}_c &= {}^{g}\mathbf{R}_i^{-1} {}^{g}\mathbf{R}_o {}^{c}\mathbf{R}_o^{-1} && (5.19)
\end{aligned}
$$

On note que ${}^{g}\mathbf{R}_i$ n'est pas défini de manière unique, tout comme ${}^{g}\mathbf{R}_o$. On va donc utiliser le déplacement entre les poses distinctes d et e au lieu des orientations fixes.

$$
\begin{aligned}
{}^{g}\mathbf{R}_o &= {}^{g}\mathbf{R}_{i(d)} {}^{i}\mathbf{R}_c {}^{c(d)}\mathbf{R}_o && (5.20)\\
&= {}^{g}\mathbf{R}_{i(e)} {}^{i}\mathbf{R}_c {}^{c(e)}\mathbf{R}_o && (5.21)\\
{}^{g}\mathbf{R}_{i(e)}^{-1} {}^{g}\mathbf{R}_{i(d)} {}^{i}\mathbf{R}_c &= {}^{i}\mathbf{R}_c {}^{c(e)}\mathbf{R}_o {}^{c(d)}\mathbf{R}_o^{-1} && (5.22)\\
rvec({}^{i}\mathbf{a}_{(d)}, {}^{i}\mathbf{a}_{(e)}) {}^{i}\mathbf{R}_c &= {}^{i}\mathbf{R}_c {}^{c(e)}\mathbf{R}_{c(d)} && (5.23)
\end{aligned}
$$

On se trouve donc dans le cas d'un système $\mathbf{AX} = \mathbf{XB}$ à résoudre qui peut être simplifié grâce aux propriétés des matrices de rotation [Tsai 89]. Soit θ_i, θ_c et θ_{ic} les angles de rotation respectifs de $rvec({}^{i}\mathbf{a}_{(d)}, {}^{i}\mathbf{a}_{(e)})$, ${}^{c(e)}\mathbf{R}_{c(d)}$ et de ${}^{i}\mathbf{R}_c$. Soit $\mathbf{v}_i$ et $\mathbf{v}_c$ les axes de rotation respectifs de $rvec({}^{i}\mathbf{a}_{(d)}, {}^{i}\mathbf{a}_{(e)})$ et de ${}^{c(e)}\mathbf{R}_{c(d)}$. On a alors

$$
\begin{aligned}
P(\theta, \mathbf{v}) &= 2sin(\frac{\theta}{2})\mathbf{v} && (5.24)\\
Q(\theta) &= 2tan\frac{\theta}{2} && (5.25)\\
[P(\theta_i, \mathbf{v}_i) + P(\theta_c, \mathbf{v}_c)]_{\times} Q(\theta_{ic}) &= P(\theta_c, \mathbf{v}_c) - P(\theta_i, \mathbf{v}_i) && (5.26)
\end{aligned}
$$

Cette dernière équation est de la forme $\mathbf{Ax} = \mathbf{b}$. On peut donc résoudre ce système par une méthode linéaire classique (Au moins 2 couples de poses, en pratique beaucoup plus, sont nécessaires pour pouvoir inverser la matrice).

FIGURE 5.4 – Calibration de ${}^{i}\mathbf{R}_c$

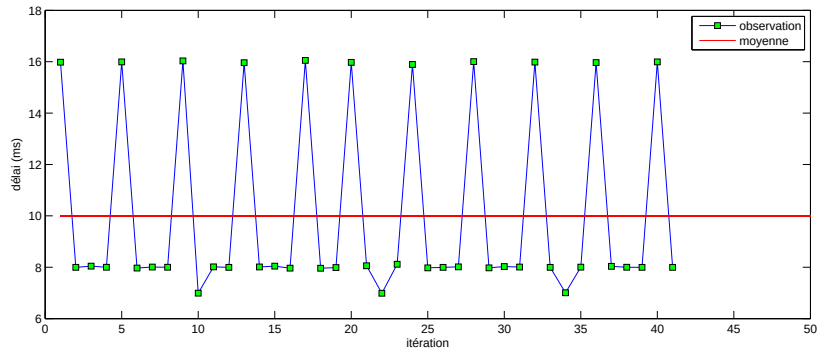


FIGURE 5.5 – Fréquence d'acquisition des données du MTx

séquence dont la pose peut être calculée précisemment avec notre filtre. Une séquence d'images avec l'instant d'acquisition associé et une suite de mesures capteurs également associées à un timestamp sont donc disponibles. À un léger biais près, la norme de la vitesse de rotation est la même pour la caméra que pour le capteur (la rotation relative ne changeant pas la norme). La figure 5.6 montre la similitude des vitesses de rotation (les hautes fréquences en moins évidemment pour la vitesse de rotation de la caméra puisqu'elle résulte d'un processus de filtrage) à un décalage temporel près.

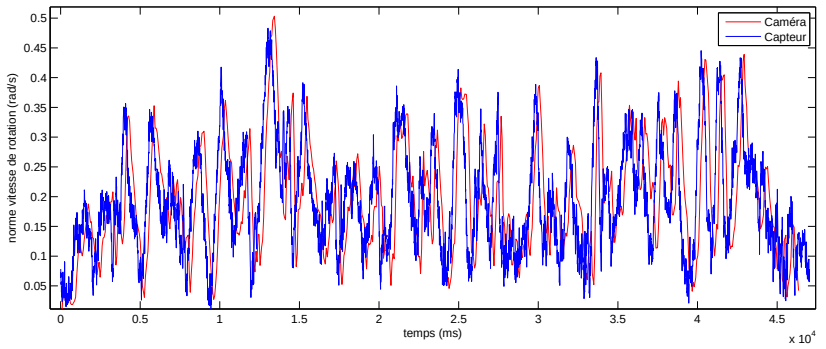


FIGURE 5.6 – Décalage des données caméra/capteurs sur une séquence

Cette équivalence est utilisée pour trouver la différence de délai d'acquisition. Le capteur inertiel est capable d'acquérir ses mesures beaucoup plus rapidement que la caméra. On cherche donc un δt à ajouter au temps d'acquisition du capteur inertiel qui minimise :

$$\sum (\mathbf{h}_{gyr}(t) - {}^{i}\boldsymbol{\omega}(t+\delta t))^2 \tag{5.27}$$

En interpolant linéairement ${}^i\boldsymbol{\omega}(t+\delta t)$ puisqu'il s'agit d'une fonction discrète. Pour notre plate-forme expérimentale, on trouve grâce à cette méthode un δt de 132ms (soit plus de 2 images et plus de 13 mesures capteurs) qui semble permettre un bon fonctionnement de la fusion dans nos tests. Il reste cependant des problèmes lors des changements d'illumination brutaux. Ces changements provoquent un ralentissement sensible de l'acquisition vidéo dû aux traitements d'images incontournables effectués par les pilotes de la caméra. Dans ces cas rares, la synchronisation n'a pas de solution.

5.6 Implémentation

Une implémentation correcte pour ce type de problématique nécessite de :
- pouvoir fusionner des données à des fréquences distinctes
- minimiser le nombre de mesures non traitées par manque de temps
- prendre en compte le décalage des délais d'acquisition entre les 2 capteurs

L'idée de notre implémentation est d'utiliser un thread différent pour chaque source de mesure. Un thread avec peu de calculs mais avec une grande fréquence de répétition pour les capteurs inertiels et un thread plus gourmand en calculs mais à plus basse fréquence. Il faut cependant prendre garde à ne pas utiliser le filtre simultanément dans les deux threads si l'on souhaite conserver une cohérence des données. Lorsque l'un des threads utilise le filtre, il met en place un drapeau qu'il enlèvera lorsqu'il aura fini son traitement. L'autre thread voyant ce drapeau saura qu'il ne peut pas demander d'opération au filtre et suivant le capteur attendra ou non la fin du blocage.

Cette méthode peut fonctionner du fait que chaque thread passe la majorité de son temps alloué à faire autre chose qu'utiliser le filtre. Le thread capteur est bloqué sur la réception des données et le thread caméra passe la majorité du temps à faire du traitement d'image et effectuer d'autres calculs non liés au filtrage.

On conserve dans une structure queue l'historique des mesures effectuées avec le capteur inertiel. À chaque étape de mise à jour du filtre utilisant les mesures inertielles, on prend en compte la mesure dans la queue dont le délai avec le temps actuel est le plus proche du délai d'acquisition entre la caméra et le capteur.

5.7 Démonstrations des avantages

5.7.1 Robustesse à des mouvements brusques et à une occlusion temporaire

La figure 5.7 montre une séquence pour laquelle un mouvement brusque a été effectué. Le mouvement trop rapide par rapport au temps d'ouverture du capteur caméra génère un flou de bougé qui provoque une perte du suivi basé image. Les capteurs inertiels doivent donc permettre de compenser la perte d'information visuelle pendant un laps de temps réduit. Ceci permet d'être capable d'estimer la position de la région suivie dans l'image au moment où celle-ci se stabilise.

Ce schéma de fonctionnement peut être utilisé pour les occlusions temporaires dans

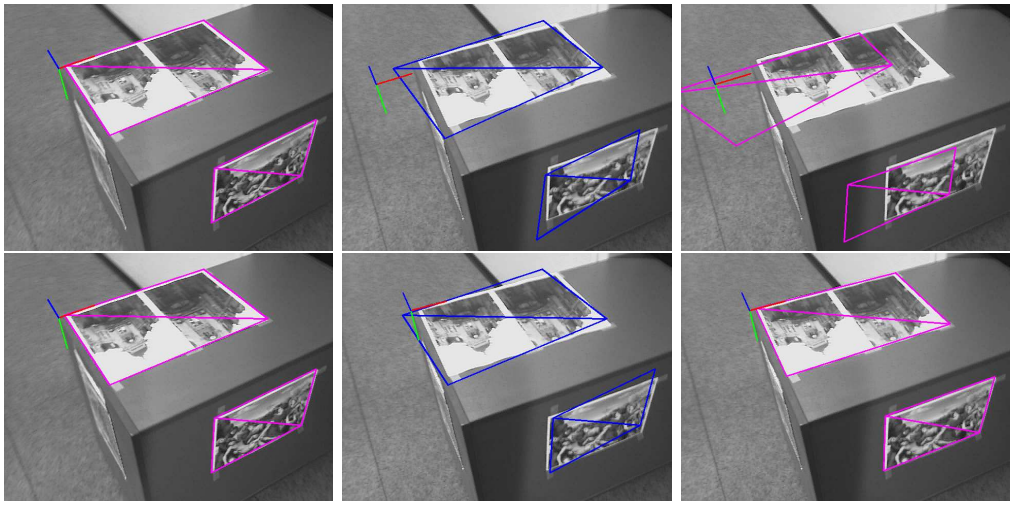


FIGURE 5.7 – Exemple de robustesse à des mouvements brusques. La colonne de gauche montre l'image acquise avant le mouvement brusque. La pose et le suivi fonctionnent encore. La colonne du milieu montre l'image après le mouvement et la prédiction de la mesure. La colonne de droite montre l'image après le mouvement et après la tentative de suivi des zones. Dans la ligne du haut, la prédiction n'est pas aidée par les capteurs inertiels, contrairement à la ligne du bas.

l'image. La figure 5.8 montre une séquence où les zones suivies ont été occultées synthétiquement et où les capteurs ont permis de compenser cette perte d'information visuelle.

5.7.2 Réduction de l'incertitude

Le fait d'utiliser une autre source d'information – *a fortiori* avec une fréquence d'acquisition rapide – permet à l'estimation d'avoir une incertitude plus réduite. En effet, l'étape de mise à jour d'un filtre de Kalman ne peut pas structurellement augmenter l'incertitude. La figure 5.9 montre pour une séquence la différence d'incertitude sur la pose entre une estimation uniquement basée vision et une autre qui fusionne les données inertielles.

Par corrélation, l'incertitude diminue à la fois pour la carte et pour la localisation. On peut donc en déduire que l'incertitude de reprojection d'un élément de la carte va être également réduite. La zone dans l'image où peut se trouver statistiquement un élément de la carte va être plus petite (Figure 5.10). Cela implique que la recherche de cet élément dans une nouvelle image va être facilitée mais surtout que l'on va pouvoir plus efficacement éliminer les mesures aberrantes (liées à une erreur du système de suivi image) qui n'auraient pas été détectées par le système de suivi.

Cette réduction des diverses incertitudes permet également d'améliorer la qualité du filtre de kalman en réduisant les problèmes de linéarisation. En effet, les incertitudes étant plus petites l'approximation linéaire de la mise à jour de la matrice de covariance est de meilleure qualité tout comme celle de la prédiction de par l'utilisation de délais

FIGURE 5.8 – Exemple de robustesse à une occlusion temporaire. La première et dernière image représente les images avant et après l'occlusion. Cette perte des mesures visuelles, provoquée par un flou et assimilable à une occlusion, dure plus de 3 secondes. Les images intermédiaires montrent que malgré la perte visuelle, la pose continue à être estimée approximativement grâce à l'utilisation des capteurs inertiels.

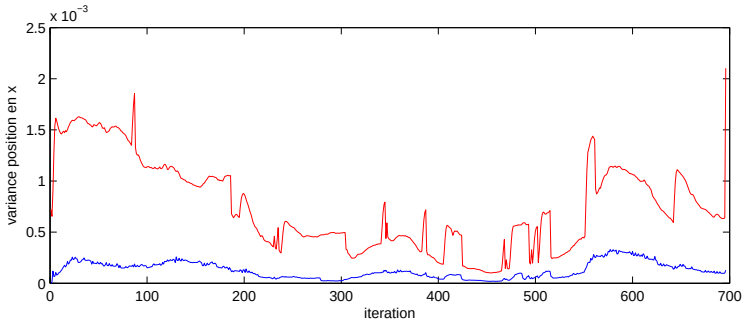


FIGURE 5.9 – Comparaison des incertitudes pour une même séquence avec et sans capteurs inertiels. La variance de la translation en x de la caméra est comparée. En rouge est affichée l'évolution de la variance sans capteurs inertiels. En bleu est affichée la même variance avec les capteurs inertiels. L'évolution globale reste similaire, mais l'incertitude avec les capteurs inertiels est plus faible.

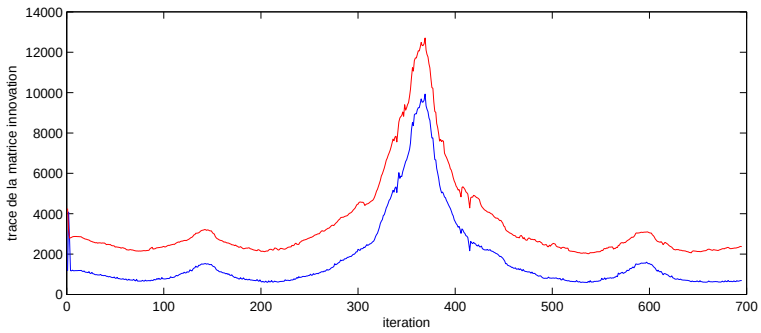


FIGURE 5.10 – Comparaison des incertitudes dans l'image pour une même séquence (séquence [chap slam]) avec et sans capteurs inertiels. La trace de la matrice d'innovation avant toute mesure image est comparée. En rouge est affichée l'évolution de la trace sans capteurs inertiels. En bleu est affichée la même trace avec les capteurs inertiels. L'évolution globale reste similaire, mais la trace avec les capteurs inertiels est plus faible.

plus courts.

Finalement, un autre avantage non négligeable des capteurs inertiels réside dans la mesure directe du mouvement. Le chapitre 2 a montré que la mise à jour de l'état du système à partir des mesures basées vision provoquent deux problèmes. Le premier est une dérive de l'estimation lorsque l'on ne fait que mesurer de nouveaux éléments. Le second problème, similaire et lié aux mesure visions, est que l'estimation n'est contrainte que par des mesures en deux dimensions. Jamais aucune information sur la profondeur réelle n'est fournie. Si on ne fournit pas à l'estimation de métrique sur les dimensions (par ex. en fournissant des plans connus à priori), la construction de la carte et la localisation se font à un facteur d'échelle près. Le capteur, à travers ses mesures directes du déplacement, permet d'avoir implicitement une idée du facteur d'échelle. La figure 5.11 montre la différence d'estimation du facteur d'échelle sur une scène simple avec et sans capteurs.

5.8 Conclusion sur l'utilisation des capteurs

Ce chapitre a démontré que les capteurs inertiels pouvaient être utilisés (dans des conditions normales) en complément et simultanément à la vision. Des méthodes pour la fusion, la synchronisation et la calibration des capteurs inertiels ont été expliquées. Elles permettent une utilisation cohérente des deux sources d'information malgré les difficultés liées aux caractéristiques matérielles de ceux-ci. En plus des avantages "évidents" des capteurs inertiels comme la compensation temporaire du manque d'information visuelle, ce chapitre a pu démontrer les avantages pour les méthodes de SLAM en général.

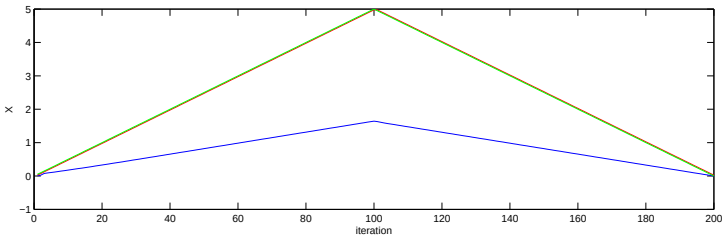


FIGURE 5.11 – Réduction de l'incertitude sur le facteur d'échelle avec les capteurs inertiels. La séquence est une simulation numérique avec uniquement une translation en X. La courbe verte représente la vérité terrain. La courbe bleue représente l'estimation de la position en X sans capteurs inertiels. La courbe rouge représente l'estimation de la position en X avec capteurs inertiels. L'incertitude du facteur d'échelle du à la vision est compensée par les capteurs inertiels. Le léger décalage entre la vérité terrain et l'estimation avec capteurs est due au fait qu'on considere l'accelération mesurée comme l'accelération actuelle alors qu'elle est en fait l'accelération des 10 ms précédentes.

Chapitre 6

Résultats de l'intégration

Ce chapitre a pour objectif de présenter des résultats synthétisant l'ensemble des méthodes décrites dans cette thèse en condition d'utilisation normale. Deux séquences sont exposées, chacune différant par le type de méthode permettant de sélectionner les régions à suivre.

6.1 Conditions d'utilisation

Ces résultats ont été obtenus à partir d'une caméra USB à faible coût (inférieur à 40 euros) et d'un capteur inertiel. Les données de la caméra sont fortement bruités à cause de la qualité médiocre du capteur. Des artefacts d'entrelacement et de flou de bougé sont visibles dès les plus petits mouvements. La caméra effectue des modifications de sa plage d'intensité lumineuse en fonction de la scène observée afin d'obtenir un maximum de contraste, ce qui provoque en retour des changements d'intensité lumineuse des régions de l'image. Cette caméra a été néanmoins choisie pour la faible distortion radiale des images qu'elle renvoie.

Bien que ces séquences soient enregistrées et traitées hors ligne, il a pu être vérifié que le temps de calcul pour chaque trame était inférieur à l'intervalle d'acquisition (60 ms) pour ces images de résolution 640 pixels de large et 480 pixels de haut. Les données des capteurs sont traitées dans un *thread* parallèle à une fréquence de 100 hertz.

Etaient connus par avance les paramètres intrinsèques $\mathbf{K}$ de la caméra, la transformation ${}^{i}\mathbf{R}_{c}$ reliant le repère la caméra $\mathcal{F}_c$ et le repère du capteur inertiel $\mathcal{F}_i$ et une base de données de photographies de la scène le cas échéant. La mise au point est bien sûr manuelle et fixe pendant l'ensemble des séquences. Dans ces séquences, l'écart type (pour le modèle de prédiction du déplacement) de l'accélération était de $0,15$ et celui de la vitesse de rotation était de $0,1$. L'écart type de la position d'un point – utilisé pour le calcul de l'incertitude de l'homographie – est fixé à 2. L'écart type des biais des capteurs inertiels est fixé à $0,015$. L'écart type des accélérations mesurées par le capteur est de $0,15$ et celui des vitesses de rotation est de $0,1$. Une mesure est rejetée

si la différence avec son estimation ne respecte pas le test du χ^2 pour un pourcentage de confiance fixé ici à 95%.

Pour terminer, les séquences sont capturées avec la caméra en main, ce qui provoque des soubresauts dans les mouvements dans les images.

6.2 Une séquence basée sur la reconnaissance des régions à suivre

Cette séquence est filmée dans un couloir. L'objectif de cette séquence est d'augementer la scène avec un poster virtuel collé contre un mur de la scène.

La caméra se déplaçant sans conserver la même cible, des éléments à suivre apparaissent et disparaissent en fonction du point de vue de la caméra. Lorsque de nouvelles régions de la scène apparaissent, une reconnaissance de région (voir section 4.1) est effectuée à l'aide d'une base de données de photographies pour trouver de nouvelles zones planaires de l'image à suivre. Lorsque des régions réapparaissent une nouvelle fois après un déplacement de la caméra, l'estimateur de pose va détecter qu'une ancienne région suivie est dans le champ de vision probable. Le suivi est alors effectué de nouveau sur cette zone en initialisant l'homographie à l'aide de la prédiction de la mesure fournie par le SLAM. Ceci permet de considérer la mesure sans avoir à rajouter de nouveaux éléments au vecteur d'état de l'estimateur et donc d'améliorer le SLAM en validant la localisation et la carte calculées depuis la disparition de l'élément ("Loop-Closing").

Afin de pouvoir augmenter la scène avec une information géo-localisée, la pose initiale de la caméra est calculée par une méthode externe [Dementhon 95]. Pour cela, le premier plan est considéré comme connu parfaitement et avec une normale à (0,0,1). Une objet rectangulaire spécifique de la base de données, connu pour être sur ce plan, est recherché. Les 4 coins de l'objet dans l'image fournissent la mesure image et la taille de l'objet étant connu, les points 3D correspondants aux 4 coins peuvent être estimés. Le repère global est alors ici le coin haut gauche de l'objet. Les objets 3D servant à augmenter la scène devront être positionnés par rapport à ce repère.

Tous les paramètres des autres plans sont inconnus *a priori*. Ils sont estimés par la méthode exposée dans la section 4.3. La reconnaissance de région n'est ici utilisée que pour déterminer les zones à suivre et non pour déterminer le plan auquel la région appartient. Une sélection extraite de la séquence d'images est illustrée dans la figure 6.1

Une modélisation 3D de la caméra et des régions suivies a été effectuée pour visualiser la scène et est présentée dans la figure 6.2

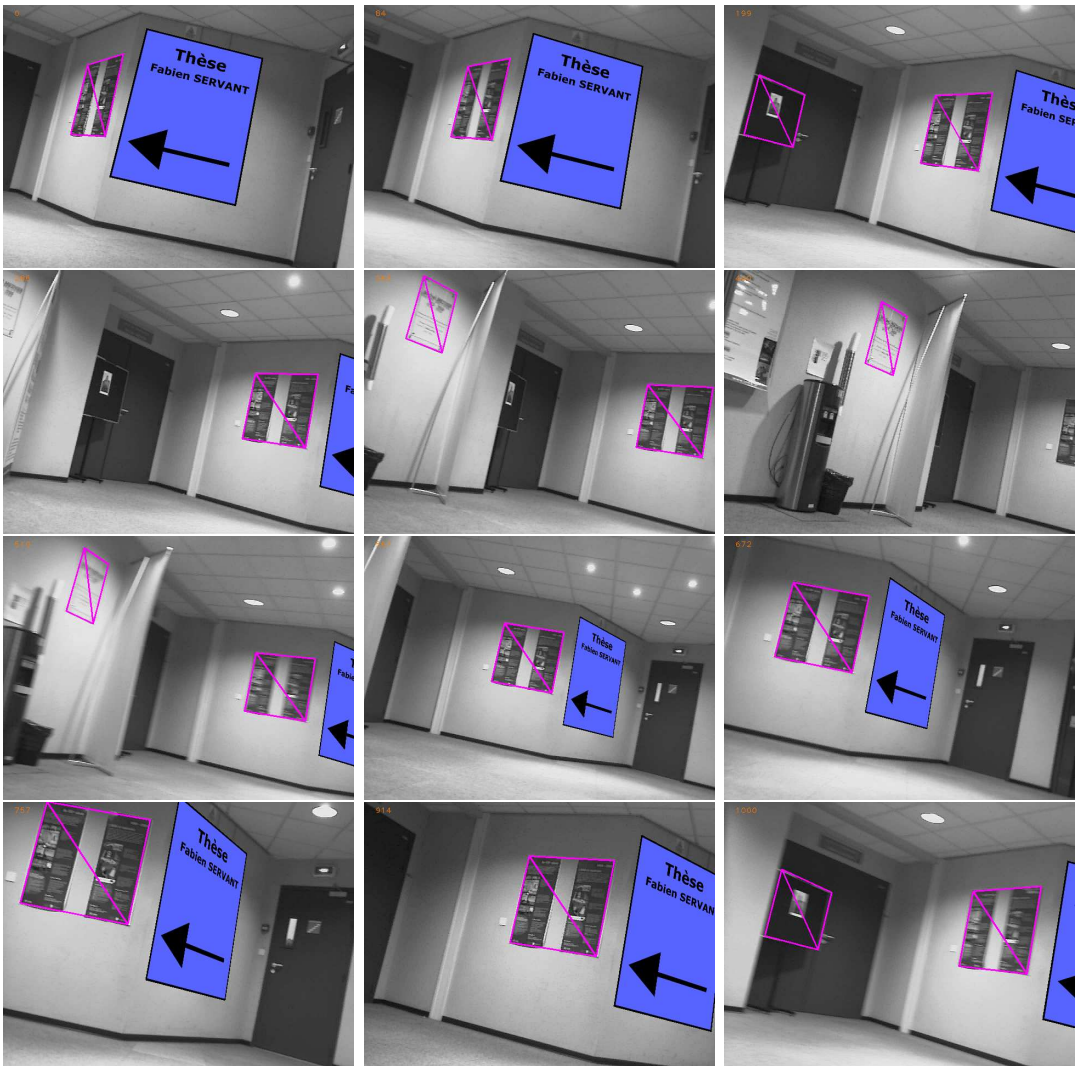


FIGURE 6.1 – Sélection d'images clés de la première séquence de résultats utilisant la reconnaissance de régions

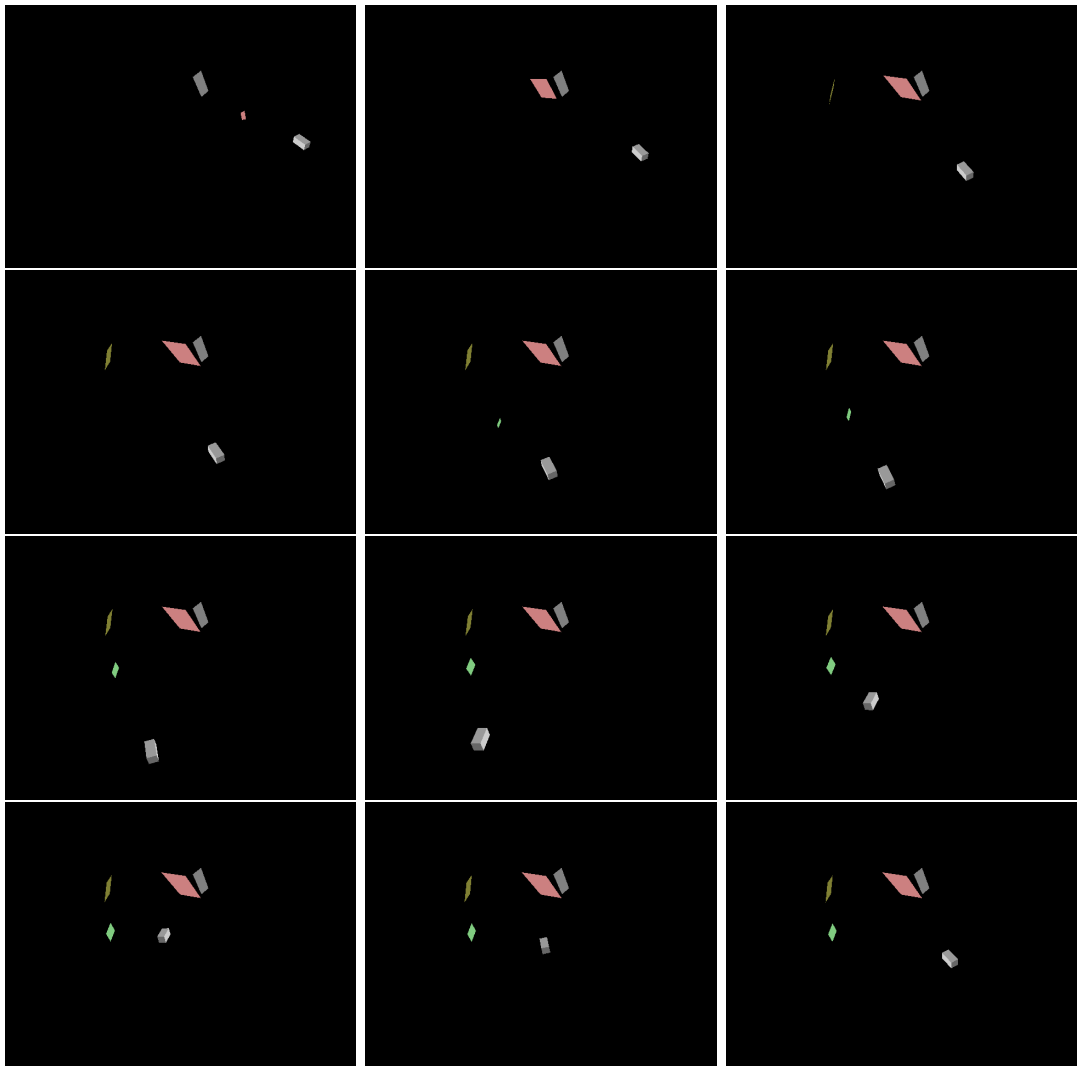


FIGURE 6.2 – Modélisation 3D de la caméra et de la carte pour une sélection d'images clés de la première séquence de résultats utilisant la reconnaissance de régions

6.3 Seconde séquence utilisant l'extraction automatique des plans

La seconde séquence illustre le comportement de notre méthode lorsqu'aucune information n'est disponible sur la scène. La pose initiale de la caméra est fixée comme étant le repère global (0 pour tous les paramètres). La méthode de localisation et d'extraction des plans simultanées (voir section 4.4) est utilisée pendant le début de la séquence. Les résultats de cette expérience sont illustrés dans la figure 6.3. Une experience de réalité augmentée, avec un modèle 3D, utilisant cette séquence est illustrée dans la figure 6.4.

En rouge sont affichés les triangles dont les profondeurs des points ne sont pas suffisamment connus. Pour chaque point de la scène est estimé l'intervalle de profondeur possible suivant un pourcentage de confiance de 95%. Si cet intervalle est inférieur à 1 cm, le point est considéré suffisamment connu pour l'extraction et il est affiché dans la séquence par un cercle blanc. Comme expliqué dans le chapitre dédié à cette problématique, lorsque les 3 points d'un triangle sont connus suffisamment, ce dernier est comparé aux autres triangles dans le même état pour être regroupé avec ceux qui lui sont coplanaires. Cette vérification se fait ici en vérifiant si (statistiquement) le produit vectoriel des deux normales est supérieur à 0.98 et si le ratio des deux distances du plan à l'origine est proche de 1 avec un écart maximal de 2 cm. Les triangles regroupés sont affichés dans une couleur définie pour chaque groupe autre que le rouge.

Lorsqu'un groupe contient plus de 3 triangles,les points correspondant aux triangles ne sont plus suivis. L'image dans laquelle les points ont été extraits est utilisée pour initialiser le suivi d'homographie. La région formée par le groupe de triangles est utilisée pour le suivi. L'homographie entre l'image d'origine et l'image actuelle prédite par le SLAM sert alors d'initialisation au suivi (nécessaire à cause de la distance possible entre les points de vue des deux caméras). Les zones prises en charge par le suivi d'homographies et intégrées dans la carte sont affichées en vert. Il est important de noter que certaines zones non planaires ont été extraites dans cette séquence. Grâce à la distance entre les deux vues et à la vérification de la validité de la mesure par le SLAM, ces régions sont cependant immédiatement écartées. Les dernières images montrent également (comme dans la première séquence) que le SLAM est capable de retrouver une région disparue pendant un certain temps.

N'ayant aucune connaissance sur la scène ni sur un quelconque repère absolu, il est impossible d'augmenter ce type de scène avec une information géolocalisée. Ce type de méthode est habituellement utilisée en conjonction avec une initialisation externe de la pose de la caméra. Cette séquence a cependant été tournée sans aucune information préalable pour montrer cette possibilité. Afin de visualiser tout de même le bon fonctionnement de l'estimation de pose, un point situé sur le bureau est sélectionné. Ce point et la normale de la région à laquelle il appartient sont utilisés pour fabriquer un repère virtuel. Ce repère est affiché en bleu dans l'image. Bien évidemment, la normale

étant mal connue dans les premières itérations, le repère est mal affiché sur les premières images. On note cependant que la pose est affichée correctement très rapidement, ce qui dénote une convergence très rapide de l'initialisation des plans. La sixième image (correspondant à la trame 61, soit 4 secondes) de la séquence démontre que les plans ont été extraits alors que le mouvement depuis la pose initiale n'est pas très important. Il faut noter le raffinement de l'affichage du repère avant même que les plans ne soient extraits, ce qui indique que la localisation s'effectue correctement pendant l'extraction.

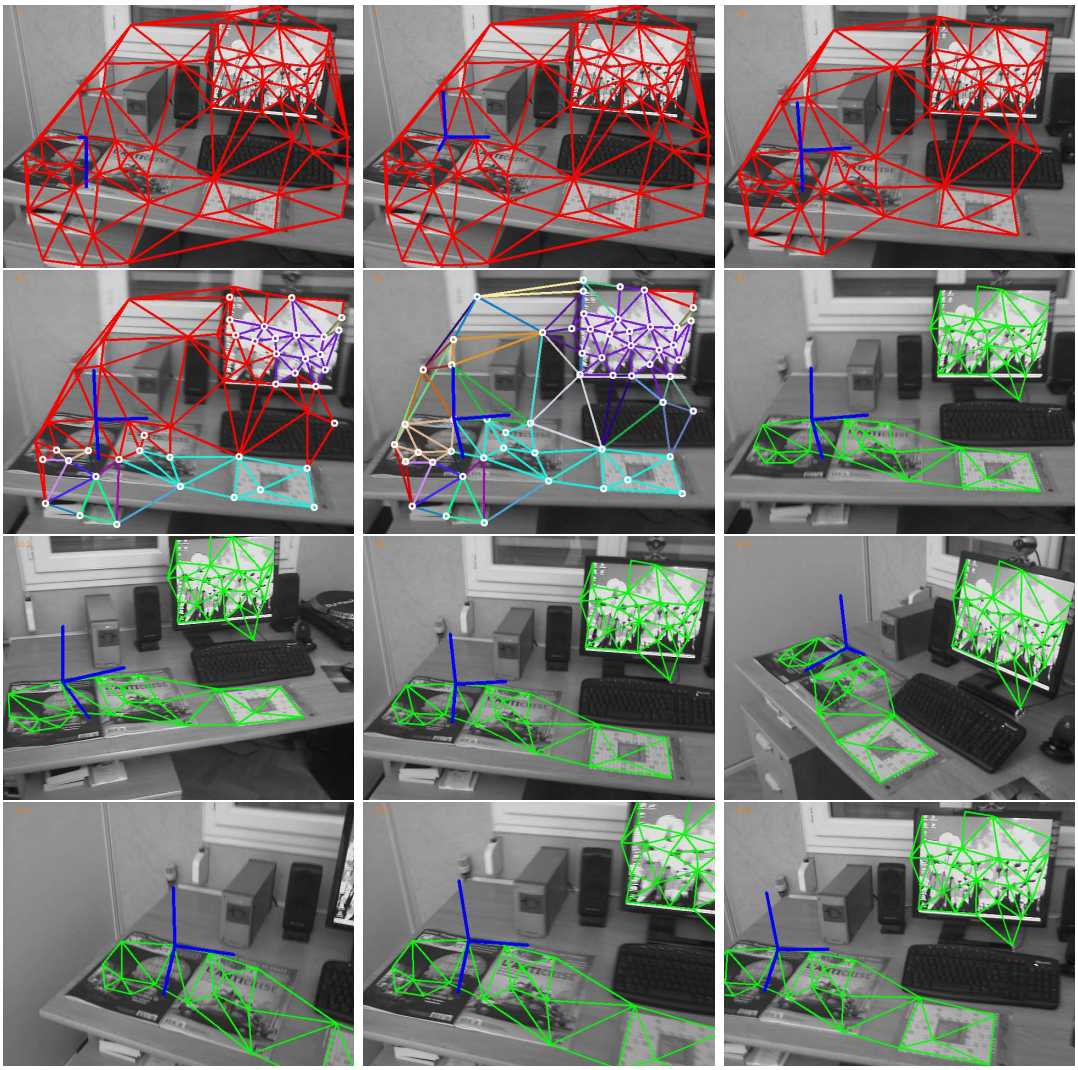


FIGURE 6.3 – Sélection d'images clés de la seconde séquence de résultats utilisant l'extraction automatique de zones planaires

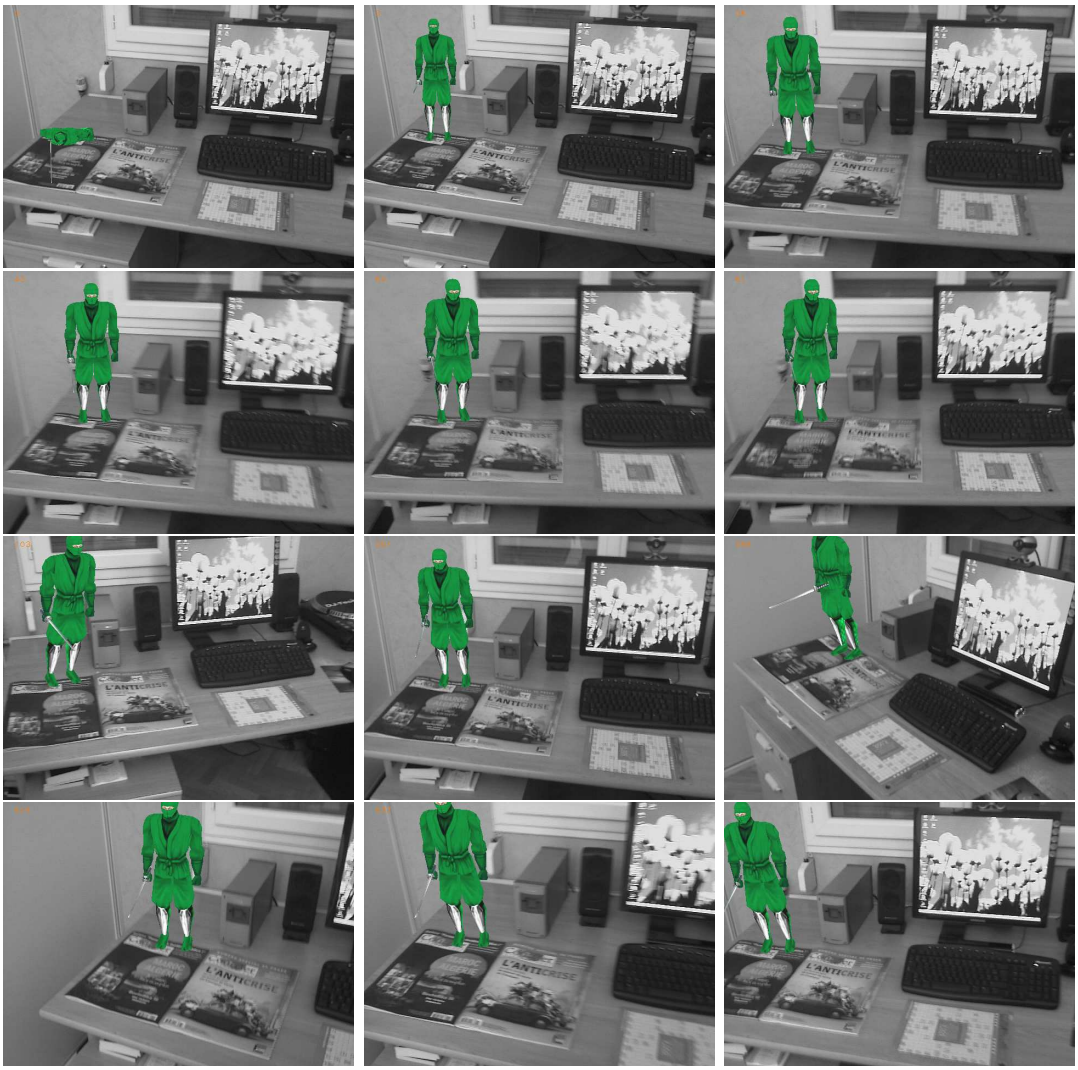


FIGURE 6.4 – Sélection d'images clés, augmentées par un modèle 3D, de la seconde séquence de résultats utilisant l'extraction automatique de zones planaires

Chapitre 7

Conclusion

Cette thèse a permis de montrer dans son ensemble la faisabilité d'un système d'estimation de pose de caméra – basé sur les plans – respectant des contraintes industrielles données en apportant une solution originale et complète. Ce pipeline complet, la robustesse, la rapidité et la prise en compte des propriétés de l'environnement et des possibilités matérielles à disposition sont autant d'atouts de la méthode proposée.

7.1 Un pipeline complet

Le pipeline commence par la sélection des zones à suivre par reconnaissance de templates ou par extraction des zones planaires. Deux méthodes ayant le même but mais dont le choix dépend des connaissances *a priori* de l'environnement permettent d'optimiser la phase d'initialisation. La reconnaissance est souvent plus robuste mais limitée aux régions présentes dans la base de données. L'extraction des plans s'appliquant quant à elle à n'importe quel type de scène, le choix de la méthode dépend de l'application finale. Ces deux méthodes ont fait l'objet de démonstrations montrant leur fonctionnement sur des séquences réelles.

Ce pipeline est complété par une méthode d'initialisation des paramètres des plans permettant d'introduire dans notre estimateur des régions planaires lorsque aucune information sur les paramètres du plan – auquel cette zone appartient – n'est connue par avance. Ceci permet d'utiliser des plans et des zones sans *a priori* sur l'environnement. Cette méthode d'initialisation, utilisant le déplacement homographique, permet d'utiliser les images sans attendre que les paramètres des plans soient estimés correctement, et donc ne crée pas de latence inutile comme dans la majorité des méthodes existantes dans la littérature. Ceci permet d'utiliser des régions n'apparaissant pas nécessairement longtemps dans la séquence d'images. Ce mémoire démontre la rapidité de convergence de cette méthode d'initialisation et sa robustesse aux bruits et autres manques de mesure.

Lorsque les zones sont initialisées et considérées planes, l'interface entre les images et l'estimateur est le suivi de régions planaires par estimation d'une transformation homographique. Cet outil permet d'estimer le déplacement dans l'image des zones sélectionnées. C'est ce déplacement, cette homographie, qui sera utilisée comme mesure par notre estimateur statistique. C'est en quelque sorte l'interface entre la géométrie abstraite de notre estimation et la réalité retranscrite par l'image capturée. Encore une fois, ce document a permis de démontrer sa capacité de convergence et l'adaptation à notre problématique générale.

Coeur du pipeline et de la méthode, l'estimateur statistique. Basé sur les notions de SLAM monoculaire et de filtre de Kalman étendu, cet estimateur permet l'estimation de la pose de la caméra au fur et à mesure du temps mais également les paramètres des plans de l'environnement afin d'assurer une cohérence globale. Recevant en entrée des homographies plus ou moins bruitées, cet estimateur est capable de fonctionner comme cela a déja été dit avec des plans inconnus mais également avec des plans dont les paramètres sont connus de manière incertaine et sont contenus dans des bases de données. Utiliser des plans inconnus permet une plus grande liberté d'utilisation. A l'opposé, utiliser des plans connus permet de réduire le problème de facteur d'échelle et de faciliter l'estimation absolue de la pose. Des démonstrations en simulations et en situations réelles ont permis de montrer son fonctionnement.

Derniere étage de ce pipeline, la prise en compte des mesures provenant d'une centrale inertielle permet d'avoir une vérité terrain afin de faciliter l'estimation. Ces mesures inertielles permettent de réduire les incertitudes liées à la mesure image. Celle-ci ne permet pas tout le temps de contraindre l'estimation et provoque notamment une incapacité à estimer autrement qu'à un facteur d'échelle près. La mesure inertielle, par sa mesure "physique" du déplacement, permet de compenser partiellement les informations non mesurables par une caméra. Elle permet surtout de compenser pendant quelques instants les moments où – à cause d'un manque de texture ou d'un déplacement trop rapide – la mesure image vient à manquer.

Tous ces étages du pipeline permettent à l'application de fonctionner de manière autonome sans intervention de l'utilisateur tout en profitant des bases de données d'images et de paramètres des plans éventuellement disponibles.

7.2 Robustesse

L'estimateur statistique permet de propager les incertitudes dans le temps et dans l'espace. Les éventuelles erreurs de mesure sont ainsi dans la mesure du possible compensées par la connaissance *a priori* et *a posteriori* de l'environnement liée aux autres mesures. Cet estimateur permet d'avoir une prédiction de la mesure permettant de faciliter le suivi dans l'image aussi bien pendant l'initialisation que pendant le suivi de zone. En effet, la prédiction permet d'une part d'initialiser le suivi à chaque image en

cas de déplacement important et d'autre part d'éliminer les mesures trop éloignées de la prédiction et donc *a priori* erronées.

L'algorithme de suivi de zones a fait l'objet d'attentions particulières pour garantir une certaine robustesse. Cet algorithme est basé sur une méthode existante capable de suivre un déplacement modélisé par des homographies efficacement. Des adaptations ont été effectuées pour faciliter la convergence malgré la présence de bruits dans l'image ou de changements de conditions d'utilisation notamment au niveau de l'illumination.

Un effort a également été fait pour diminuer la surface de la région suivie nécessaire au bon fonctionnement du suivi. De par la complexité du modèle de déplacement et la faible qualité des images, l'algorithme de suivi nécessite un certain nombre de pixels pour fonctionner correctement. Plus grande est la surface nécessaire, plus difficile est la sélection de zones convenables. Les améliorations apportées permettent d'utiliser des surfaces suffisamment petites pour de nombreuses scènes.

7.3 Rapidité

Les algorithmes de suivi et d'initialisation sont très rapides, grâce notamment à une rapidité de convergence nécessitant très peu d'itérations et une possibilité de parallélisation non négligeable.

Mise à part les optimisations logicielles nécessaires effectuées, la méthode proposée permet de réduire le coût calculatoire par rapport aux méthodes de SLAM monoculaire classiques. Malgré un modèle de mesure plus complexe qu'à l'accoutumée, le coût calculatoire est plus faible sur la méthode présentée dans cette thèse qu'avec un modèle basé points. En effet, le coût calculatoire en $O(n^2)$ est principalement lié à la taille du vecteur d'état. Dans un modèle classique basé points, la carte est constituée de trois scalaires par point. Avec la méthode proposée, la carte est constituée de six paramètres par caméra de référence et de trois scalaires par triangle. Dans le pire des cas, la carte est donc de même taille. Dans la réalité, parce qu'il n'y a pas une caméra de référence par plan et que les triangles sont fusionnés suivant leur appartenance à un plan, la carte est beaucoup plus petite et la complexité réduite.

7.4 Prise en compte des contraintes de l'environnement

Le pipeline présenté dans cette thèse permet de profiter de l'abondance de zones planaires dans les environnements construits par l'homme. Extraction, suivi et estimateur sont axés sur cette caractéristique de l'environnement. Cependant, grâce à la méthode d'extraction des zones planaires, le suivi peut fonctionner pendant quelques temps sans aucune zone planaire dans l'environnement. À partir du moment où peuvent être extraits des triangles avec leurs trois angles suffisamment ouverts, la pose peut être estimée à l'aide des plans "virtuels" construits par ces triangles. Ceci permet un fonctionnement continu malgré l'inexistance de zones planaires pendant certaines périodes.

Bien sûr, les zones planaires seront préférées, permettant l'utilisation de méthodes de tracking plus performantes puisque respectant un modèle de déplacement sans simplifications importantes.

7.5 Prise en compte des contraintes matérielles

La première contrainte matérielle étant la capacité à fonctionner avec des ressources calculatoires, le lecteur se dirigera sur le paragraphe ci-dessus, consacré à la rapidité, pour plus de détails. Les implémentations optimales n'ayant pas encore été effectuées sur les plate-formes cibles,il est impossible d'avancer de chiffres précis. Cependant les premiers tests notamment du tracking et de l'estimateur en simulation sur ces plate-formes permettent d'espérer un fonctionnement en temps réel sur les UMPC fournis.

La seconde contrainte est liée aux périphériques utilisés. Cette thèse expliquait comment les capteurs inertiels fournis pouvaient être utilisés pour faciliter le processus. La caméra quant à elle est de qualité moyenne, type "webcam" et fournit une image sans contraste et bruitée. La robustesse du pipeline permet de compenser la plupart du temps ces défauts matériels incontournables.

Une autre complication liée aux périphériques utilisés tient dans la connectivité utilisée. Ces périphériques utilisent en effet une liaison USB qui va provoquer des délais dans l'acquisition des données. Cette prise en compte des délais est nécessaire pour la fusion des informations visuelles et inertielles. Elle fonctionne en condition normale. Dans certains cas cependant (par exemple lors d'une baisse soudaine de luminosité) les délais peuvent subitement s'allonger à cause des traitements d'image effectués par les pilotes de la caméra. Dans ces cas exceptionnels le fonctionnement correct de l'estimation et de la fusion n'est plus garanti.

7.6 Perspectives

7.6.1 Perspectives d'amélioration du SLAM

Un travail intéressant et non effectué serait de méler le SLAM monoculaire basé points avec notre méthode. Parce que la structure globale est similaire, cette fusion devrait être possible sans rencontrer de problème d'incompatibilité. Ceci permettrait de prendre le relais de la méthode proposée lors des périodes longues où aucune zone planaire n'est visible. Il reste à déterminer cependant des heuristiques pour basculer entre une méthode ou une autre, la modification ayant un certain coût calculatoire.

Aucune recherche n'a été effectuée sur les problèmes de fermeture de boucle. Comment reconnaitre après une longue séquence un zone de l'image déja suivie auparavant ? La zone étant plane, il y a peu de risque qu'elle soit sensiblement déformée. Actuellement, lorsque l'estimateur pense qu'il y a normalement dans le champ de vision de la

caméra une ancienne zone potentiellement présente, il tente un suivi en réinitialisant l'homographie du tracker avec les valeurs prédites. Ceci fonctionne si l'incertitude est suffisamment faible et que la distance dans l'image de la zone prédite et de la zone réelle n'est pas trop grande. Des algorithmes permettant de rendre plus robuste cette association des mesures à des éléments de la carte non mesurés depuis longtemps existent dans la littérature [Surmann 04, Newman 06] et pourraient peut-être être appliqués à la méthode proposée.

Toujours concernant le SLAM, notre méthode considère actuellement la carte de l'environnement comme un tout. L'ensemble des éléments rencontrés pendant la séquence d'utilisation est utilisé pour la mise à jour de l'état, y compris des éléments très éloignés. Ceci conduit à une augmentation permanente du vecteur d'état et donc à un accroissement du coût calculatoire. Dans de longues séquences, la carte peut devenir très grande et rendre impossible le calcul de la mise à jour du filtre dans des temps raisonnables. Des solutions permettant de gérer de grandes cartes existent dans la littérature (voir section 2.3.3). Elles proposent, entre autre, de diviser la carte en sous-cartes reliées entres elles et de n'estimer qu'une partie de la carte à la fois. Ces méthodes sont actuellement appliquées à des cartes de points. *A priori*, rien n'indique que l'application de ces méthodes à une carte de plans comme utilisée dans ce document poserait de problèmes majeurs. Le choix de la méthode optimale pour les plans et sa mise en place restent cependant à effectuer.

7.6.2 Perspectives d'amélioration du suivi de régions

D'autres perspectives de recherche intéressantes concernent le suivi de régions qui constitue une partie importante pour le bon fonctionnement de la méthode présentée dans cette thèse.

Notre méthode nécessite l'ajout régulier de régions à suivre. En effet, l'utilisateur se déplaçant, l'environnement observé change avec le temps. Ce document à montré comment sélectionner des régions intéressantes dans la séquence d'image. Cependant aucune recherche approfondie n'a été faite sur le choix des moments judicieux pour effectuer cette sélection. Pour le moment, la recherche est faite dans les zones de l'image ne contenant pas de région suivie quand le nombre de pixels suivis est inférieur à un seuil. Il serait intéressant de trouver une méthode permettant de déterminer intelligemment quand rechercher de nouvelles zones à suivre et quelles parties de l'image analyser. Une méthode préemptive basée sur l'analyse des déplacements précédents de la caméra pourrait être imaginée pour combler cette lacune.

Lors de l'extraction actuelle des plans dans une séquence d'images, les points sont extraits par une méthode classique non spécifique à la recherche de plans. Un traitement d'image plus poussé permettrait peut-être de trouver des zones qui ont plus de chances d'être planaires. D'autre part, le suivi de point utilisé lors de l'extraction des plans est une méthode générique qui n'utilise la prédiction fournie par le SLAM que pour prédire

la position du point dans l'image. Une possibilité serait de transformer la région autour du point en la considérant localement planaire comme proposé dans [Montiel 06] pour améliorer la qualité de cette méthode de suivi.

7.6.3 Perspectives d'améliorations logicielles

Une perspective industrielle importante serait d'implémenter de manière optimale l'ensemble du pipeline sur les plates-formes cibles pour vérifier le bon fonctionnement en temps réel. Ceci nécessitera surement quelques modifications dans le pipeline d'execution. Pour le moment, et par manque de temps, il reste à optimiser l'estimateur statistique, notamment la partie d'insertion. Des solutions algorithmiques ont déja été trouvées. Dans un objectif d'industrialisation du procédé ou de certaines de ses composantes, il semble intéressant d'étudier la faisabilité du portage des algorithmes sur des puces dédiées (type FPGA). Ce portage semble être particulièrement efficace pour les étapes de traitement d'image.

Annexe A

Estimation robuste

A.1 Problématique

Soit $\mathbf{Q} = \{\mathbf{q}_i, \forall i = 1 : n\}$ un ensemble d'objets mathématiques. Soit f un modèle permettant de transformer les éléments de $\mathbf{Q}$ suivant les paramètres donnés par $\mathbf{x}$. Le résultat de cette transformation est un autre ensemble $\mathbf{Z} = \{\mathbf{z}_i, \forall i = 1 : n\}$. Les propriétés de $\mathbf{Q}$ et $\mathbf{Z}$ dépendent de la problématique d'estimation. La relation entre $\mathbf{Q}$ et $\mathbf{Z}$ est :

$$\mathbf{z}_i = f(\mathbf{x}, \mathbf{q}_i) \tag{A.1}$$

Soit $\hat{\mathbf{x}}$ une estimation de $\mathbf{x}$ et $\mathbf{Z}$ la mesure. L'objectif de l'estimation est de trouver les paramètres $\hat{\mathbf{x}}$ qui minimisent l'erreur $||\mathbf{e}||$:

$$\mathbf{e}_i = \mathbf{z}_i - f(\hat{\mathbf{x}}, \mathbf{q}_i) \tag{A.2}$$

La méthode permettant d'obtenir $\hat{\mathbf{x}}$ compte tenu de l'erreur dépend de f.

Cette estimation des paramètres nécessite que la mesure représente réellement l'application de f sur $\mathbf{Q}$ pour que les paramètres $\hat{\mathbf{x}}$ puissent être estimés correctement. Il peut arriver cependant que les mesures soit erronées à cause de bruit de mesure ou parce que l'objet mesuré n'est en fait pas un élément de $\mathbf{Q}$. Il est alors impératif de prendre en compte le mieux possible ces "outliers". Une première méthode nommée M-estimateurs a été présentée dans le chapitre sur le suivi basé image (voir section 3.5.3). Deux autres méthodes sont également utilisées dans cette thèse : La méthode RANSAC (RANdom SAmple Consensus) et la méthode LMedS (Least Median Square).

A.2 Les méthodes de minimisation robuste par consensus

Le principe de ces deux méthodes est similaire et est basé sur le principe du consensus. Dans des conditions optimales, l'estimation des paramètres $\hat{\mathbf{x}}$ nécessite un nombre

k de mesures. Dans les faits, le cardinal de $\mathbf{Z}$ est bien souvent largement supérieur à k, pour compenser le bruit des mesures et minimiser le risque que les mesures ne contraignent pas tous les degrés de liberté.

1. Prenons un sous-ensemble $\mathbf{S}$ de $\mathbf{Z}$ dont le cardinal est k et choisi aléatoirement. L'estimation de $\hat{\mathbf{x}}$ est alors effectuée sur $\mathbf{S}$ et non plus sur $\mathbf{Z}$. Si $\mathbf{S}$ est dégénéré (Il ne permet pas l'estimation de $\hat{\mathbf{x}}$), un autre sous-ensemble est choisi aléatoirement.

2. Pour chaque élément de $\mathbf{Z}$, l'erreur $\mathbf{e}_i$ est calculée en fonction des paramètres estimés à l'étape précédente. Des éléments sont sélectionnés (cette sélection dépendant de la méthode) et sont regroupés dans un sous-ensemble $\mathbf{A} = \{\mathbf{a}_i, \forall i = 1 : c\}$.

3. Les étapes 1 et 2 sont répétées N fois. L'ensemble $\mathbf{A}$ dont les propriétés respectent au mieux un critère de qualité est sélectionné. Le terme de "inlier" est utilisé pour décrire les éléments de $\mathbf{A}$ en opposition à "outliers" pour les autres éléments de $\mathbf{Z}$. On dit que les paramètres $\hat{\mathbf{x}}$ correspondant au sous-ensemble $\mathbf{A}$ sélectionné font *consensus*.

4. L'estimation des paramètres $\hat{\mathbf{x}}$ finaux est effectué sur l'ensemble $\mathbf{A}$.

Idéalement, N doit être le nombre de combinaisons de k éléments de $\mathbf{Z}$. Dans la pratique, N serait alors trop grand et occasionnerait un coût de calcul trop important. N est alors choisi statistiquement de manière à minimiser la probabilité P de ne pas générer aléatoirement un sous-ensemble (parmi les N) comprenant uniquement des "inliers". Soit p une constante déterminant le pourcentage d'"outliers" qui semble probable et qui est fixé manuellement. P est déterminé par :

$$P(p, |\mathbf{S}|, N) = (1 - (1 - p)^{|\mathbf{S}|})^N \tag{A.3}$$

$P(p, |\mathbf{S}|, N)$ étant une constante fixée manuellement (qui revient à exprimer la probabilité acceptable que l'algorithme ne fonctionne pas), N peut être calculé par :

$$N = \frac{log(1 - P(p, |\mathbf{S}|, N))}{log((1 - p)^{|\mathbf{S}|})} \tag{A.4}$$

La différence entre les deux méthodes est simplement dans la sélection des éléments de $\mathbf{A}$ et du critère de qualité de $\mathbf{A}$.

A.2.1 RANSAC

RANSAC nécessite qu'un seuil ϵ, correspondant à la valeur maximale que peut prendre l'erreur d'un élément de $\mathbf{Z}$ pour qu'il soit classé comme "inlier" et donc qu'il fasse partie de $\mathbf{A}$, soit fixé manuellement. Ceci nécessite donc de connaitre pour chaque

utilisation de cette méthode les caractéristiques des mesures et les précisions dans les mesures requises par l'estimateur. Le critère de qualité utilisé dans l'étape 3 est le cardinal du sous-ensemble **A** courant.

A.2.2 LMedS

Dans le cas de la méthode LMedS, le sous-ensemble **A** contient l'intégralité de **Z** trié suivant l'erreur $\mathbf{e}_i$ associée à chaque $\mathbf{z}_i$. Le critère de qualité utilisé dans l'étape 3 est la médiane du sous-ensemble **A** courant : Le sous-ensemble **A** dont la médiane est la plus petite est sélectionné. Aucun seuil n'est ici fixé, cependant le choix de la médiane implique qu'au moins 50% des éléments de **Z** soient des inliers.

Annexe B

Dérivations des matrices

Les notions expliquées dans cette section sont détaillées dans [Brewer 78].

La dérivée $\frac{\partial \mathbf{x}}{\partial \mathbf{y}}$ d'un vecteur $\mathbf{x}$ en fonction d'un autre vecteur $\mathbf{y}$ (nommée jacobienne) est une matrice dont chaque terme $\mathbf{J_{ab}}$ consiste en la dérivée du scalaire $\mathbf{x_a}$ en fonction du scalaire $\mathbf{y_b}$. Lorsque l'élément à dériver n'est plus un vecteur mais une matrice $\mathbf{X}$, le résultat devient une structure délicate à manipuler algébriquement. À condition d'utiliser cette méthode dans chaque dérivée d'une application, la structure peut être simplifiée en dérivant $\frac{\partial(\mathbf{X^s})}{\partial(\mathbf{Y^s})}$. Le résultat redevient une matrice puisque $(\mathbf{X}^s)$ et $(\mathbf{Y}^s)$ sont des vecteurs. Notons que ce principe s'étend à tout type de matrice $\mathbf{X}$ ou $\mathbf{Y}$, y compris un simple vecteur ou un scalaire.

Multiplications Considérons le terme $\mathbf{AB}$, la multiplication des deux matrices $\mathbf{A}$ et $\mathbf{B}$. $\frac{\partial \mathbf{AB}}{\partial \mathbf{C}}$ peut être calculée par $\frac{\partial(\mathbf{AB^s})}{\partial(\mathbf{C^s})}$. Ceci nécessite le calcul de $\mathbf{AB}$ avant la dérivation. Il est possible de garder séparés les deux éléments par souci de lisibilité et de simplifications ultérieures. l'article [Brewer 78] défini que

$$(\mathbf{ADB}^s) = (\mathbf{B}^\mathsf{T} \otimes \mathbf{A})(\mathbf{D}^s) \tag{B.1}$$

et donc

$$(\mathbf{IAB}^s) = (\mathbf{B}^\mathsf{T} \otimes \mathbf{I})(\mathbf{A}^s) \tag{B.2}$$
$$(\mathbf{ABI}^s) = (\mathbf{I} \otimes \mathbf{A})(\mathbf{B}^s) \tag{B.3}$$

alors

$$\frac{\partial \mathbf{AB}}{\partial \mathbf{C}} = (\mathbf{B}^\mathsf{T} \otimes \mathbf{I})\frac{\partial \mathbf{A}}{\partial \mathbf{C}} + (\mathbf{I} \otimes \mathbf{A})\frac{\partial \mathbf{B}}{\partial \mathbf{C}} \tag{B.4}$$

Inversion La dérivation de l'inverse d'une matrice est également utilisée dans cette thèse et peut être calculée par :

$$\frac{\partial \mathbf{A^{-1}}}{\partial \mathbf{C}} = \frac{\partial \mathbf{A^{-1}}}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial \mathbf{C}} \tag{B.5}$$

$$= (-\mathbf{A^{-T}} \otimes \mathbf{A}^{-1}) \frac{\partial \mathbf{A}}{\partial \mathbf{C}} \tag{B.6}$$

Bibliographie

[Alspach 72] D. Alspach, H. Sorenson. – Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4) :439–448, Aug 1972.

[Azuma 99] R. Azuma, B. Hoff, III Neely, H., R. Sarfaty. – A motion-stabilized outdoor augmented reality system. *Virtual Reality, 1999. Proceedings., IEEE*, pp. 252–259, Mar 1999.

[Bailey 06] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, Eduardo Nebot. – Consistency of the ekf-slam algorithm. *2006 IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pp. 3562–3568, Oct. 2006.

[Baldwin 09] Mahony Baldwin, Trumpf. – A nonlinear observer for 6 dof pose estimation from inertial and bearing measurements. *IEEE Intl. Conf. on Robotics and Automation*, 2009.

[BarShalom 93] Y. Bar-Shalom, X.-R. Li. – *Estimation and Tracking, Principles, Techniques, and Software.* – Artech House, Boston, 1993.

[Bartoli 07] A. Bartoli. – A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105 :2007, 2007.

[Benhimane 07] S. Benhimane, E. Malis. – Homography-based 2d visual tracking and servoing. *Int. Journal of Computer Vision*, 26(7) :661–676, juillet 2007.

[Bourquardez 04] O. Bourquardez. – *Asservissement visuel et localisation d'un hélicoptère miniature.* – Rapport de DEA, DEA photonique, image et cybernétique, 2004.

[Brewer 78] J.W. Brewer. – Kronecker products and matrix calculus in system theory. *IEEE Trans. on Circuits and Systems*, 25 :772–781, 1978.

[Brown 71] D.C. Brown. – Close-range camera calibration. *Photogrammetric Engineering*, 4(2) :127–140, mars 1971.

[Chong 99] K.S. Chong, L. Kleeman. – Feature-based mapping in real large scale environments using an ultrasonic array. *IJRR*, 18(1) :3–19, January 1999.

[Comport 06] A.I. Comport, E. Marchand, M. Pressigout, F. Chaumette. – Real-time markerless tracking for augmented reality : the virtual visual servoing framework. *IEEE Trans. on Visualization and Computer Graphics*, 12(4) :615–628, juillet 2006.

[Csorba 97] M. Csorba. – *Simultaneous Localization and Map Building.* – PhD. Thesis, University of Oxford, 1997.

[Davison 02] A.J. Davison, D.W. Murray. – Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :865–880, Jul 2002.

[Davison 03] A.J. Davison. – Real-time simultaneous localisation and mapping with a single camera. – *IEEE Int. Conf. on Computer Vision*, pp. 1403–1410, 2003.

[Davison 07] A. Davison, I. Reid, N. Molton, O. Stasse. – Monoslam : Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6) :1052–1067, 2007.

[Delaunay 34] B Delaunay. – Sur la sphère vide. a la memoire de georges voronoi. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskih i Estestvennyh Nauk*, (7) :793–800, 1934.

[Dementhon 95] D. Dementhon, L. Davis. – Model-based object pose in 25 lines of codes. *Int. J. of Computer Vision*, 15(1-2) :123–141, 1995.

[Devillers 02] O. Devillers. – The Delaunay Hierarchy. *International Journal of Foundations of Computer Science*, 13 :163–180, 2002.

[Dhome 89] M. Dhome, M. Richetin, J.-T. Lapresté, G. Rives. – Determination of the attitude of 3D objects from a single perspective view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12) :1265–1278, décembre 1989.

[Dhome 90] M. Dhome, J.-T. Lapresté, G. Rives, M. Richetin. – Determination of the attitude of modelled objects of revolution in monocular perspective vision. – *European Conf. on Computer Vision, ECCV'90*, vol. 427 of *Lecture Notes in Computer Science*, pp. 475–485, Antibes, avril 1990.

[Drummond 02] T. Drummond, R. Cipolla. – Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7) :932–946, juillet 2002.

[DurrantWhyte 06] Hugh Durrant-Whyte, Tim Bailey. – Simultaneous localization and mapping : Part i. *IEEE Robotics & Automation Magazine*, 13(2) :99–108, 2006.

[Estrada 05] C. Estrada, J. Neira, J.D. Tardos. – Hierarchical slam : Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4) :588–596, Aug. 2005.

[Eustice 05] R.M. Eustice, H. Singh, J.J. Leonard. – Exactly sparse delayed-state filters. *, 2005. ICRA 2005. IEEE Intl Conf on Robotics and Automation*, pp. 2417–2424, April 2005.

[Eustice 06] R.M. Eustice, H. Singh, J.J. Leonard. – Exactly sparse delayed-state filters for view-based slam. *Robotics, IEEE Transactions on*, 22(6) :1100–1114, Dec. 2006.

[F. 06] Dellaert F., Kaess M. – Square Root SAM : Simultaneous Localization and Mapping via Square Root Information Smoothing. *The Intl Journal of Robotics Research*, 25(12) :1181–1203, 2006.

[Faugeras 87] O.D Faugeras, G. Toscani. – Camera calibration for 3D computer vision. – *Proc Int. Workshop on Machine Vision and Machine Intelligence*, pp. 240–247, Tokyo, février 1987.

[Fischler 81] N. Fischler, R.C. Bolles. – Random sample consensus : A paradigm for model fitting with application to image analysis and automated cartography. *Communication of the ACM*, 24(6) :381–395, juin 1981.

[Gee 07] Andrew P. Gee, Denis Chekhlov, Walterio Mayol, Andrew Calway. – Discovering planes and collapsing the state space in visual slam. *British Machine Vision Conference*, September 2007.

[Gemeiner 07] Peter Gemeiner, Peter Einramhof, Markus Vincze. – Simultaneous motion and structure estimation by fusion of inertial and vision data. *Int. J. Rob. Res.*, 26(6) :591–605, 2007.

[Guivant 01] J.E. Guivant, E.M. Nebot. – Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3) :242–257, Jun 2001.

[Hager 98] G. Hager, P. Belhumeur. – Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10) :1025–1039, octobre 1998.

[Harris 88] C. Harris, M. Stephens. – A combined corner and edge detector. – *Alvey Conference*, pp. 147–151, Manchester, 1988.

[Hartley 01] R. Hartley, A. Zisserman. – *Multiple View Geometry in Computer Vision*. – Cambridge University Press, 2001.

[H.DurrantWhyte 96] D.Rye H. Durrant-Whyte, E.Nebot. – Localisation of automatic guided vehicles. *the 7th international symposium on Robotics research*, pp. 613–625, 1996.

[Herbert 08] Bay Herbert, Ess Andreas, Tuytelaars Tinne, Van Gool Luc. – Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3) :346–359, 2008.

[Horaud 89] R. Horaud, B. Conio, O. Leboulleux, B. Lacolle. – An analytic solution for the perspective 4-points problem. *Computer Vision, Graphics and Image Processing*, 47(1) :33–44, juillet 1989.

[Isard 98] M. Isard, A. Blake. – Condensation – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1) :5–28, janvier 1998.

[Julier 01] S.J. Julier, J.K. Uhlmann. – A counter example to the theory of simultaneous localization and map building. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 4 :4238–4243 vol.4, 2001.

[Jung 04] I.K. Jung. – *Simultaneous localization and mapping in 3D environments with stereovision.* – PhD. Thesis, Toulouse : Institut National Polytechnique, 2004.

[Ke 04] Y. Ke, R. Sukthankar. – Pca-sift : A more distinctive representation for local image descriptor. – *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 506–513, Washington, DC, juillet 2004.

[Kähler 07] Olaf Kähler, Joachim Denzler. – Detecting coplanar feature points in handheld image sequences. *Conference on Computer Vision Theory and Applications*, pp. 447–452, 2007.

[Klein 04] Georg Klein, Tom Drummond. – Tightly integrated sensor fusion for robust visual tracking. *Image and Vision Computing*, 22(10) :769–776, September 2004.

[Knight 01] J. Knight, A. Davison, I. Reid. – Towards constant time slam using postponement. *International Conference on Intelligent Robots and Systems*, 2001.

[Koller 97] D. Koller, G. Klinger, E. Rosse, D. Breen, R. Whitaker, M. Tuceryan. – Real-time vision-based camera tracking for augmented reality applications. – *Int. Symp. on Virtual Reality Software and Technology, VRST'97*, pp. 87–94, Lausanne, Switzerland, septembre 1997.

[Kwok 04] N.M. Kwok, G. Dissanayake. – An efficient multiple hypothesis filter for bearing-only slam. *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 1 :736–741 vol.1, Sept.-2 Oct. 2004.

[Liu 90] Y. Liu, T.S. Huang, O.D. Faugeras. – Determination of camera location from 2D to 3D line and point correspondences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1) :28–37, janvier 1990.

[LonguetHiggins 81] H.C. Longuet-Higgins. – A computer algorithm for reconstructing a scene from two projections. *Nature*, 293 :133–135, septembre 1981.

[LopezNicolas 05] G. Lopez Nicolas, J.J. Guerrero, O.A. Pellejero, C. Sagues. – Computing homographies from three lines or points in an image pair. *CIAP'05*, pp. 446–453, 2005.

[Lourakis 02] M I. A. Lourakis, A. A. Argyros, S. C. Orphanoudakis. – Detecting planes in an uncalibrated image pair. *BMVC*, 2002.

[Lowe 87] D.G. Lowe. – Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3) :355–394, mars 1987.

[Lowe 91] D.G. Lowe. – Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5) :441–450, mai 1991.

[Lowe 04] D. Lowe. – Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2) :91–110, 2004.

[Malis 98] E. Malis. – *Contributions à la modélisation et à la commande en asservissement visuel*. – PhD. Thesis, Universite de Rennes 1, Télécommunications et Traitement du Signal, novembre 1998.

[Marchand 02a] E. Marchand, F. Chaumette. – Virtual visual servoing : a framework for real-time augmented reality. – *EUROGRAPHICS'02 Conf. Proceeding*, vol. 21(3) of *Computer Graphics Forum*, pp. 289–298, Saarebrücken, Germany, septembre 2002.

[Marchand 02b] E. Marchand, N. Courty. – Controlling a camera in a virtual environment : Visual servoing in computer animation. *The Visual Computer Journal*, 18(1) :1–19, février 2002.

[Martin 02] F. Martin, R. Horaud. – Multiple camera tracking of rigid objects. *Int. Journal of Robotics Research*, 21(2) :97–113, février 2002. – (INRIA RR-4268, september 2001).

[Matas 02] J. Matas, O. Chum, M. Urban, T. Pajdla. – Robust wide baseline stereo from maximally stable extremal regions. *British Machine Vision Conference*, pp. 384–393, 2002.

[Mei 07] C. Mei, P. Rives. – Cartographie et localisation simultanée avec un capteur de vision. *Journées Nationales de la Recherche en Robotique*, October 2007.

[Mikolajczyk 05] Krystian Mikolajczyk, Cordelia Schmid. – A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10) :1615–1630, 2005.

[Molton 04] N. D. Molton, A. J. Davison, I. D. Reid. – Locally planar patch features for real-time structure from motion. *British Machine Vision Conference*, septembre 2004. – (To appear).

[Montemerlo 03a] M. Montemerlo. – *FastSLAM : A factored solution to the simultaneous localization and mapping problem*. – PhD. Thesis, Robotics Institute, Carnegie Mellon University, juillet 2003.

[Montemerlo 03b] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. – FastSLAM 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *Intl Joint Conf on Artificial Intelligence*, 2003.

[Montiel 06] J. Montiel, J. Civera, A. Davison. – Unified inverse depth parametrization for monocular slam. – *Robotics : Science and Systems*, Philadelphia, USA, August 2006.

[Newman 00] P. Newman. – *On the structure and solution of the simultaneous localisation and map building problem.* – PhD. Thesis, University of Oxford, 2000.

[Newman 06] P. Newman, D. Cole, K. Ho. – Outdoor slam using visual appearance and laser ranging. *IEEE Intl Conf on Robotics and Automation*, pp. 1180–1187, 2006.

[Oberkampf 96] D. Oberkampf, D.F. Dementhon, L.S. Davis. – Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3) :495–511, mai 1996.

[Ozuysal 07] M. Ozuysal, P. Fua, V. Lepetit. – Fast keypoint recognition in ten lines of code. – *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2007.

[Phong 95] T.-Q. Phong, R. Horaud, A. Yassine, P.-D. Tao. – Object pose from a 2D to 3D point and line correspondance. *Int. Journal of Computer Vision*, 15(3) :225–243, juillet 1995.

[Pietzsch 08] Tobias Pietzsch. – Planar features for visual slam. *German conf. on Advances in Artificial Intelligence*, pp. 119–126, 2008.

[Pinies 07] P. Pinies, T. Lupton, S. Sukkarieh, J.D. Tardos. – Inertial aiding of inverse depth slam using a monocular camera. *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2797–2802, April 2007.

[Pinies 08] P. Pinies, J.D. Tardos. – Large-scale slam building conditionally independent local maps : Application to monocular vision. *IEEE Transactions on Robotics*, 24(5) :1094–1106, Oct. 2008.

[Rekleitis 03] Ioannis Rekleitis, Gregory Dudek, Evangelos Milios. – Probabilistic cooperative localization and mapping in practice. *IEEE Intl. Conf. on Robotics and Automation*, pp. 1907–1912, Sept. 2003.

[Safaee-Rad 92] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, K.C. Smith. – Three dimentional location estimation of circular features for machine vision. *IEEE Trans. on Robotics and Automation*, 8(2) :624–639, octobre 1992.

[Salvi 01] J. Salvi, X. Armangue, J. Pages. – A survey addressing the fundamental matrix estimation problem. *Intl Conf. on Image Processing*, 2 :209–212 vol.2, Oct 2001.

[Se 02] S. Se, D. Lowe, J. Little. – Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Intl. Journal of Robotics Research*, 21 :735–758, 2002.

[Shi 94] J. Shi, C. Tomasi. – Good features to track. – *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pp. 593–600, Seattle, Washington, juin 1994.

[Silveira 06] Geraldo Silveira, Ezio Malis, Patrick Rives. – Real-time robust detection of planar regions in a pair of images. *Intl. Conf. Intelligent Robots and Systems*, pp. 49–54, Oct. 2006.

[Silveira 08] G. Silveira, E. Malis, P. Rives. – An efficient direct approach to visual slam. *IEEE Transactions on Robotics*, 24(5) :969–979, Oct. 2008.

[Simon 98] G. Simon, M.-O. Berger. – A two-stage robust statistical method for temporal registration from features of various type. – *Int. Conf. on Computer Vision, ICCV'98*, pp. 261–266, Bombay, India, Janvier 1998.

[Simon 02] G. Simon, M.-O. Berger. – Pose estimation for planar structures. *IEEE Computer Graphics and Applications*, 22(6) :46–53, novembre 2002.

[Smith 91] R. Smith, M. Self, P. Cheeseman. – A stochastic map for uncertain spatial relationships. *Autonomous Mobile Robots : Perception, Mapping, and Navigation (Vol. 1)*, éd. par S. S. Iyengar, A. Elfes, pp. 323–330. – Los Alamitos, CA, IEEE Computer Society Press, 1991.

[Smith 06] P. Smith, I. Reid, A. Davison. – Real-time monocular slam with straight lines. *British Machine Vision Conference*, 1 :17–26, September 2006.

[Sola 05] J. Sola, A. Monin, M. Devy, T. Lemaire. – Undelayed initialization in bearing only slam. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2499–2504, Aug. 2005.

[Surmann 04] Hartmut Surmann, Andreas Nüchter, Kai Lingemann, Joachim Hertzberg. – 6d slam a preliminary report on closing the loop in six dimensions. *IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.

[Tardos 02] Juan D. Tardos, Jose Neira, Paul M. Newman, John J. Leonard. – Robust mapping and localization in indoor environments using sonar data. *Int. J. Robotics Research*, 21 :311–330, 2002.

[Thrun 98] Sebastian Thrun, Wolfram Burgard, Dieter Fox. – A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3) :29–53, 1998.

[Tsai 86] R.Y. Tsai. – An efficient and accurate camera calibration technique for 3D machine vision. – *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'86*, pp. 364–374, Miami, Floride, juin 1986.

[Tsai 89] R. Tsai, R. Lenz. – A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. on Robotics and Automation*, 5(3) :345–358, juin 1989.

[Williams 01] S. Williams. – *Efficient Solutions to Autonomous Mapping and Navigation Problems.* – PhD. Thesis, University of Sydney, 2001.

[You 99] S. You, U. Neumann, R. Azuma. – Hybrid inertial and vision tracking for augmented reality registration. *IEEE Virtual Reality*, pp. 260–267, Mar 1999.

[Zhang 92] Z. Zhang, O. Faugeras. – *Three D-Dynamic Scene Analysis : A Stereo Based Approach.* – Springer-Verlag, Secaucus, NJ, USA, 1992.

Table des figures

Résumé

Le travail présenté dans ce manuscrit se situe dans le domaine de la réalité augmentée. Dans le contexte de la réalité augmentée, il est nécessaire de calculer la position relative entre la caméra et la scène en temps réel. Cette thèse présente une méthode complète de calcul de pose reposant sur l'utilisation de structures planes présentes dans les environnements urbains intérieurs et extérieurs. Le suivi de pose est effectué à l'aide d'une caméra bas coût et d'un capteur inertiel.

Notre approche consiste à tirer parti des plans pour faciliter le calcul de la pose. Les homographies obtenues par un algorithme de suivi dans l'image de régions planes, présenté dans cette thèse, servent de mesures à notre méthode de localisation et cartographie simultanées. Cette méthode de SLAM permet d'assurer un suivi de la pose robuste tout en permettant une reconstruction de la scène 3D et fonctionnant sur le long terme en propageant les incertitudes des mesures. Des travaux sur la sélection des régions à suivre et sur l'initialisation des paramètres des plans, correspondant à ces régions, sont également présentés. Des expériences en simulation et sur des séquences d'images réelles montrent la validité de notre approche.

Mots-clefs : Réalité augmentée, localisation et cartographie simultanées

Abstract

Our work deals with computer vision. The problem of augmented reality implies a real time estimation of the relive position between camera and scene. This thesis presents a complete method of pose tracking that works with planar structures which are abundant in indoor and outdoor urban environments. The pose tracking is done using a low cost camera and an inertial sensor.

Our approach is to use the planes to make the pose estimation easier. Homographies computed by an image tracking algorithm presented in this document are used as measurements for our Simultaneous Localization And Mapping method. This SLAM method permits a long term and robust pose tracking by propagating the measurements uncertainties. Works about selection of regions to track and their corresponding plane parameters initialization are also described in this thesis. Numerical and image based experiments shows the validity of our approach.

Key words : Augmented reality, simultaneous localization and mapping