Improving Visual-Inertial Navigation Using Stationary Environmental Magnetic Disturbances

David Caruso

université PARIS-SACLAY

NNT: 2018SACLS133

Thèse de doctorat

UNIVERSITÉ PARIS SUD
Comprendre le monde, construire l'avenir

Amélioration des méthodes de navigation vision-inertiel par exploitation des perturbations magnétiques stationnaires de l'environnement

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

École doctorale n°580 : sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: traitement du signal et des images

Thèse présentée et soutenue à Palaiseau, le 1 Juin 2018, par

David Caruso

Composition du Jury :

Pascal Vasseur Professeur, IUT de Rouen (LITIS)	Président
Silvère Bonnabel Professeur, Ecole des Mines de Paris (CAOR)	Rapporteur
José Neira Professeur, Universidad de Zaragoza (GRTR)	Rapporteur
Samia Bouchafa Professeur, Université d'Evry Val de Seine (IBISC)	Examinateur
Michel Dhome Directeur de Recherche, CNRS (Institut Pascal)	Examinateur
Guy Le Besnerais Directeur de Recherche, ONERA (DTIS)	Directeur de thèse
Alexandre Eudes Ingénieur de Recherche, ONERA (DTIS)	Encadrant
David Vissière PDG, Sysnav	Encadrant
Martial Sanfourche Ingénieur de Recherche, ONERA (DTIS)	Encadrant (Invité)

Improving Visual-Inertial Navigation Using Stationary Environmental Magnetic Disturbances

Amélioration des méthodes de navigation vision-inertielles par exploitation des perturbations magnétiques stationnaires de l'environnement

David Caruso

June 1st, 2018

Remerciements

Je souhaite remercier les professeurs José Neira et Silvère Bonnabel pour m'avoir fait l'honneur d'accepter de rapporter cette thèse, ainsi que les professeurs Samia Bouchafa, Pascal Vasseur, et Michel Dhome, examinateurs, pour s'être intéressés à mes travaux et avoir pris la décision de m'accorder le titre de docteur.

Je tiens à remercier mon directeur de thèse Guy Le Besnerais. Très investi, exigent, franc et toujours bienveillant, il a réussi à me pousser à faire une bonne thèse, prodiguant des conseils toujours avisés et efficaces.

Je remercie David Vissière qui, grâce à son enthousiasme légendaire a donné l'impulsion initiale à ces travaux, qui m'a fait confiance jusqu'au bout, même lorsqu'il avait des doutes sur les orientations prises. J'ai finalement eu une grande liberté et autonomie, et ce, malgré les préoccupations d'une PME souvent difficiles à prévoir sur la durée d'un doctorat scientifique.

Je remercie grandement Martial et Alexandre : mes interlocuteurs privilégiés sur les aspects techniques et scientifiques de la thèse, mais aussi sur sa conduite générale. Merci Martial pour m'avoir fait partager ton expérience sur l'odométrie visuelle et les capteurs, ta bonne humeur et ta gentillesse. Merci Alex pour nos discussions techniques qui m'ont permis d'avancer dans ma compréhension tout en redonnant à des détails techniques, souvent arides (sordides?), un sens exploitable et un intérêt qui mérite de les raconter. Sans ton travail de relecture attentif, la qualité des papiers et du manuscrit aurait été grandement réduites.

Je remercie également mes collègues de Sysnav. Charles-Yvan pour m'avoir toujours apporté son aide et ses conseils sur le capteur. Ceux-ci ont été primordiales : sans ses travaux de thèse simultanées aux miens, les résultats auraient été plus ternes. Mathieu Hillion pour avoir toujours une réponse à toute question. Quentin Désile pour l'aide apportées en mécanique et électronique, pour la bonne humeur et les projets fous. Ludivine, pour les rappels réguliers au sujet des tâches que mon cerveau refusait de retenir et pour la logistique efficace lors des missions. Guillaume et Pierre, pour avoir égayer le bureau au troisième étage à coup d'accent marseillais, de "Kwasiment", de Chasse-Patate et de Nerf. Georges Guy, pour être une source de science et d'histoire technologique inépuisable, loquace et passionnante. Marc, Louis et Xavier pour les discussions dans le train Paris-Vernon. J'ai trouvé à Sysnav des ingénieurs excellents et passionnés qui m'ont énormément inspiré dans la façon de résoudre les défis techniques rencontrées, merci à Éric, Jean-Philippe, Pierre-Jean, Adrien, Hendrik, Bertrand, Edouard et les autres.

Je remercie les personnes que j'ai eu la chance de rencontrer à l'Onera. Même en y étant peu présent, je m'y suis toujours senti très bien accueilli. Calum pour son humour toujours audacieux et pour le soutien mutuel en phase de rédaction, Hélène pour son militantisme et pour une crémaillère mémorable, Maxime F. pour les discussions sur le SLAM et celles non-scientifiques, Maxime B. pour les keynotes plutôt non-scientifiques et pour la feu-Tireuse, Joris et Guillaume pour les énigmes, pour l'animation et pour l'ambiance, Hicham pour faire du Hicham, constamment. Anthelme, Nicolas, Marcella, Maxime D., Isabelle, Oriane, Semy, Pierre, Rodolphe, Alex Boulch, Aurélien Plyer, Frédéric Champagnat et les autres, qui ont tous également participé à cet accueil et avec qui j'ai eu plaisir à échanger.

Je remercie aussi Nadège et Jakob Engel, qui m'ont initié au travail de recherche durant les stages précédents le début de la thèse, respectivement à l'été 2013 au cours de l'année 2014.

J'ai la chance de ne pouvoir travailler efficacement qu'en musique; je remercie à ce titre les Joyeux Urbains, Ibrahim Maalouf, Ratatat, les Cow-Boys Fringants, Orelsan, et une poignée d'autres artistes pour m'avoir accompagné auditivement l'immense partie de la difficile et trop longue phase de rédaction.

Merci à mes parents et ma famille de m'avoir permis d'aller aussi loin dans les études facilement, et qui m'ont toujours encouragé dans tout ce que j'entreprenais. Je remercie également ceux que j'ai eu la chance de rencontrer lors de mon parcours et qui sont devenus mes amis; j'ai été ému de voir certains d'entre eux assister à la soutenance de cette thèse. Je remercie spécialement Marie, qui a été à mes côtés les trois dernières années, qui m'a soutenu dans les moments difficiles sans jamais m'en vouloir, et a rendu ces moments plus faciles à vivre.

Summary

Français

Cette thèse s'intéresse au problème de positionnement (position et orientation) dans un contexte de réalité augmentée et aborde spécifiquement les solutions à base de capteurs embarqués.

Aujourd'hui, les performances atteintes par les VINSs *(système de navigation vision-inertiels) commencent à être compatibles avec les besoins spécifiques de cette application. Néanmoins, ces systèmes de positionnement se basent tous sur des corrections de trajectoire issues des informations visuelles à relativement haute fréquence afin de remédier à la rapide dérive des capteurs inertiels bas-coûts. Cela pose problème lorsque l'environnement visuel n'est pas favorable; par exemple lors de la présence de fumée, d'une illumination de mauvaise qualité, de flou de bougé ou encore lorsque l'environnement est fortement dynamique ou peu texturé.*

Parallèlement, des travaux récents menés par l'entreprise SYSNAV *ont démontré qu'il était possible de réduire la dérive de l'intégration inertielle en exploitant le champ magnétique, grâce à un nouveau type d'*UMI *(unité de mesure inertielle) composée – en plus des accéléromètres et gyromètres traditionnels – d'un réseau de magnétomètres. Celui-ci fournit une mesure du gradient du champ magnétique local à chaque instant qui est exploitée en formulant des hypothèses raisonnables sur le champ. Ce capteur est composé seulement de composants bas-coûts et les algorithmes associés ont un surcoût en calcul faible. Néanmoins, cette méthode de navigation à l'estime est également mise en défaut si les hypothèses sur le champ ne sont pas vérifiées au moins localement autour du capteur, par exemple en environnement extérieur, ou en présence de perturbations instationnaires du champ.*

Nos travaux portent sur le développement d'une solution de navigation à l'estime robuste combinant toutes ces sources d'informations : magnétiques, visuelles et inertielles.

Un premier travail fusionne de façon lâche (loose fusion) des informations de poses issues d'un algorithme d'alignement d'image d'un capteur de profondeur avec celles issues d'un filtre de navigation à l'estime magnéto-inertiel. La logique de l'estimation repose sur la connaissance et la détection des modes de défaillance de chaque estimateur afin d'obtenir une navigation plus robuste.

Un deuxième travail présente une façon statistiquement cohérente d'insérer des termes d'erreurs issues des mesures de gradient magnétiques dans les méthodes d'ajustement de faisceaux classiquement utilisées dans les algorithmes de SLAM *(localisation et cartographie simultanée), déjà étendues par ailleurs aux* VINS*. Nous développons une approche par préintégration des mesures magnétiques, inspirées de techniques proposées dans la littérature pour le traitement des données inertielles.*

Un troisième travail met en œuvre un EKF *(filtre de Kalman étendu) pour la navigation sur les mêmes données. Nous utilisons une structure de filtre inspirée du filtre* MSCKF *proposé par de Mourikis et Roumeliotis en 2007, que nous implémentons dans la formulation en racine carrée du filtre d'information. Comparé à l'approche par ajustement de faisceaux, ce filtre est plus efficient, ce qui laisse entrevoir une implémentation simple sur processeur embarqué.*

Ces méthodes sont toutes testées et validées sur des données issues de capteurs réels, dans des scénarios présentant des difficultés à la fois pour la vision et pour le capteur magnéto-inertiel. Nous démontrons sur ces essais le gain de robustesse issu de la fusion, et nous étudions les performances des différentes combinaisons de capteurs possibles : magnéto-inertielles, vision-inertielles et vision-magnéto-inertielles.

Enfin, un dernier travail concerne certaines propriétés plus fines de la solution basée filtrage. Il est reconnu dans la littérature que ces filtres ne sont pas statistiquement consistants, notamment à cause de l'accumulation des erreurs de linéarisation. Nous explorons et appliquons des solutions à cette problématique d'inconsistance. S'inspirant de travaux récents sur l'utilisation d'erreurs

non-linéaires dans le filtre de Kalman, nous étudions le rôle de la paramétrisation de l'état et du choix de l'erreur filtrée dans le cas particulier de notre problème. Il est montré que, pour certains de ces choix, l'estimateur présente des propriétés intéressantes liées à sa consistance; son comportement s'en trouve amélioré.

English

This thesis addresses the positioning in 6DOF *(position and orientation) issues arising from* AR *(Augmented Reality) applications and focuses on embedded sensors based solutions.*

Nowadays, performance reached by VINSs *(Visual-Inertial Navigation Systems) is starting to be adequate for* AR *applications. Nonetheless, those systems are based on position correction from visual sensors involved at relatively high frequency in order to mitigate the quick drift of low-cost inertial sensors. This is a problem when visual environment is not favorable. As for instance in foggy or smoky environments, in presence of bad or changing illumination, motion blur, or when the scene is highly dynamic.*

In parallel, recent works conducted at the company SYSNAV *have shown that it was feasible to leverage magnetic field to reduce inertial translation drift thanks to a new type of* IMU *(Inertial Measurement Unit), consisting – in addition to the accelerometers and gyrometers – in a network of magnetometers, which allows measuring magnetic field gradient at each instant. This information is exploited with reasonable assumptions about the magnetic field. This system has the advantage of using only low-cost sensors and induces only low computational overhead. However, this dead-reckoning technique fails if the assumed hypotheses are not fulfilled, at least in the vicinity of the sensor. This is the case generally outdoor, or when the magnetic field disturbances are not stationary.*

Our work aims to develop a robust dead-reckoning solution combining information from all these sources: magnetic, visual, and inertial sensors.

A first work does a loose fusion between pose information from a depth image alignment algorithm with those from a magneto-inertial dead-reckoning filter. This system is mainly based on detecting and taking into account the failure modes of each estimator in order to gain robustness.

A second work presents a statistically consistent way to integrate error terms from magnetic gradient values into the bundle adjustment algorithm classically used in SLAM *(Simultaneous Localization and Mapping), which has been already extended to* VINS*. We develop an approach using preintegration of magnetic measurement inspired by techniques proposed in the literature for inertial data handling.*

A third work implements an EKF *(Extended Kalman Filter) for dead-reckoning on the same data. We use a filter structure inspired from the* MSCKF *(Multi-State Constraint Filter), proposed in 2007 by Mourikis and Roumeliotis, that we implement in its information square-root form. Compared to the bundle adjustment approach, the filter is more efficient, which could open the way to an implementation on embedded processor.*

All these methods are tested and validated on real sensors data, on scenarios difficult for vision or magneto-inertial sensors. We show on these trajectories robustness gain from the fusion, and we study the performance of different combinations of sensors: magneto-inertial, vision-inertial and vision-magneto-inertial.

Finally, a last contribution relates to fine grain properties of the filtering solution. It is recognized in the literature that those filters loose their statistical consistency, in particular because linearization errors accumulations. We explore and apply solutions to this issue. Inspired by recent works on the use of non-linear error in the Kalman filter, we study the role of the error state parametrization in the particular case of our problem. It is shown that, with a certain choice of parametrization, the estimator presents interesting properties linked to its consistency; its behavior is found to be improved.

This work was conducted between January 2015 and February 2018. It was motivated and financed by the company SYSNAV*, 57 rue de Montigny, 27200 Vernon, France, and the ANRT within a CIFRE framework. It was supervised scientifically by* SYSNAV *and the Image-vision-apprentissage (*IVA*) team of* Departement traitement de l'information et systèmes *(*DTIS*) of ONERA, Chemin de la Hunière, 91120 Palaiseau.*

Contents

List of Figures

List of Abbreviations

6DOF	position and orientation.
ADC	Analog to Digitial Converter.
AR	Augmented Reality.
EKF	Extended Kalman Filter.
EKF-SLAM	EKF-SLAM.
ENU	East-North-Up.
GNSS	Global Navigation Satellite System.
GPGPU	General-purpose Processing on Graphics Processing Units.
HDR	High Dynamic Range.
IEKF	Invariant-EKF.
IMU	Inertial Measurement Unit.
INS	Inertial Navigation System.
IRLS	Iteratively Reweighted Least-Square.
KLT	Kanade–Lucas–Tomasi feature tracker.
MAP	Maximum A Posteriori.
MEMS	MicroElectroMechanical System.
MI-DR	Magneto-Inertial Dead-Reckoning.
MI-MSCKF	Magneto-Inertial Multi-State Constraint Filter.
MIMU	Magneto-Inertial Measurement Unit.
MSCKF	Multi-State Constraint Filter.
MVINS	Magneto-Visual-Inertial Navigation System.
NEES	Normalized Estimation Error Squared.
NNLS	Non-Linear Least Squares.
OC-EKF	Observability Constrained EKF.
RANSAC	RANdom SAmple Consensus.
RGBD	Red-Green-Blue-Depth.
ROS	Robotic Operating System.
SLAM	Simultaneous Localization and Mapping.
VINS	Visual-Inertial Navigation System.
VIO	Visual-Inertial Odometry.
VO	Visual Odometry.

Conventions and Notations

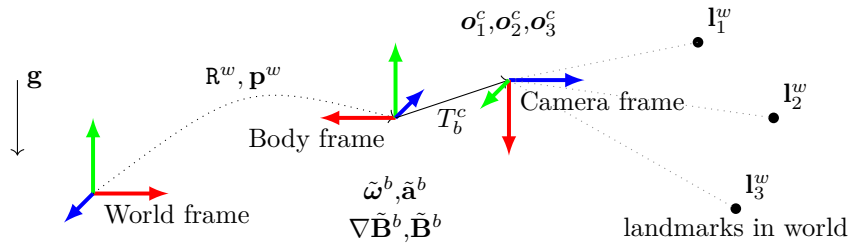


Figure 1: Reference coordinate frames at play in the general problem, with associated typical measurements.

Matrix and manifold elements notations

Bold capital letters $\mathbf{X}$ denote matrices or elements of manifold.
Parenthesis are used to denote the Cartesian product of two elements $a \in \mathcal{A}, b \in \mathcal{B} \mapsto (a, b) \in \mathcal{A} \times \mathcal{B}$
Brackets are for matrices and the concatenation of two compatible matrices.
For a vector $\mathbf{x} = [x_1, x_2, x_3]^\mathsf{T}$, $\mathbf{x}_2$ denotes its second component x_2 and $\mathbf{x}_{2:3}$ is the sub-vector $[x_2, x_3]^\mathsf{T}$.

Matrix vectorization and Kronecker Product

For matrices, we define the vectorization operation $\mathrm{Vec}\,()$ so that:

$$\mathrm{Vec}\left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}\right) = \begin{bmatrix} x_{11}, x_{21}, x_{12}, x_{22} \end{bmatrix}^\mathsf{T}. \tag{1}$$

We also use the Kronecker product specially for noise and error derivations, it is defined as:

$$\mathbf{A} \otimes \mathbf{B} := \begin{bmatrix} \mathbf{A}_{1,1}\mathbf{B} & \mathbf{A}_{1,2}\mathbf{B} & \cdots & \mathbf{A}_{1,p}\mathbf{B} \\ \mathbf{A}_{2,1}\mathbf{B} & \mathbf{A}_{2,2}\mathbf{B} & \cdots & \mathbf{A}_{2,p}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{k,1}\mathbf{B} & \mathbf{A}_{k,2}\mathbf{B} & \cdots & \mathbf{A}_{k,p}\mathbf{B} \end{bmatrix}, , \quad \text{With } \mathbf{A} \in \mathbb{R}^{k\times p} \text{ and } \mathbf{B} \in \mathbb{R}^{m\times p}. \tag{2}$$

This operator is specially handy to express the vectorized version of a product of three matrices:

$$\mathrm{Vec}\,(\mathbf{ABC}) = \left(\mathbf{C}^\mathsf{T} \otimes \mathbf{A}\right) \mathrm{Vec}\,(\mathbf{B}) \tag{3}$$

Which proves particularly useful to differentiate with respect to a matrix coefficients.

Symbol accent

We use a tilde symbol $\tilde{x}$ for measured quantities or generally quantities that can be derived directly or indirectly from sensor reading. This is a rather unconventional choice, that might disturb at first reader coming from a theoretical automatic background, where the tilde often denotes an error variables. More conventionally though, we use a hat $\hat{\mathbf{X}}$ for estimated version of a quantity.

Reference frames convention

The *world* coordinates (i.e. the frame in which the systems navigate) are *always* defined such that the gravity vector writes $\mathbf{g} \simeq [0, 0, -9.81]^\mathsf{T}$. When ambiguous, the reference frame in which a quantity is expressed will be noted in exponent: w stands for the world gravity-aligned reference frame, b for the current body reference frame and c for the camera frame. The Figure 1 summarizes the chosen notations.

Rotation parametrization and SO(3) Group notation

For rotations, we use the convention that $\mathtt{R}^w$ transforms a vector from body frame to world frame by left multiplying it. For the sake of clarity of the developments, we represent the attitude of the sensor as a rotation matrix belonging to the matrix Special Orthogonal Group. This group is denoted SO(3) and its associated Lie algebra $\mathfrak{so}(3)$ — the set of skew symmetric matrices. Any element of $\mathfrak{so}(3)$ can be identified with a vector of $\mathbb{R}^3$: $[x]_\times \in \mathfrak{so}(3)$ with $x \in \mathbb{R}^3$ and vex so that $vex([x]_\times) = x$. exp and log are the standard exponential map and logarithm on matrices. We will for conciseness often use "vectorized" versions of exp and log in the case of SO(3):

$$\mathrm{Exp}_{\mathrm{SO}(3)}(\boldsymbol{\delta\theta}) : \begin{array}{ccc} \mathbb{R}^3 & \to & \mathrm{SO}(3) \\ \boldsymbol{\delta\theta} & \mapsto & \exp([\boldsymbol{\delta\theta}]_\times) \end{array} \tag{4}$$

and $\mathrm{Log}_{\mathrm{SO}(3)}(\mathtt{R}) : \mathrm{SO}(3) \to \mathbb{R}^3$ the inverse function. With these conventions,

$$\mathrm{Log}_{\mathrm{SO}(3)}\left(\mathrm{Exp}_{\mathrm{SO}(3)}(x)\right) = x \tag{5}$$

Other Matrix Lie Group Notations

For other matrix Lie group, that would be introduced f9or instance in Chapter 7. We will employ notation similar to the above. A (matrix) element of the Lie algebra $\mathfrak{m}$ will be noted $m^\wedge = \log(\mathbf{M}) \in \mathfrak{m}, \mathbf{M} \in \mathcal{M}$ (this is the equivalent to $[\theta]_\times$ on $\mathfrak{so}(3)$). And can be identified to an element $\mathbf{m} \in \mathbb{R}^n$. We write the vectorized version of exp and log maps as $\mathrm{Exp}_{\mathcal{M}}(\mathbf{m})$ and $\mathrm{Log}_{\mathcal{M}}(\mathbf{M})$

Spaces

$\mathbb{R}^n$	Euclidean space of dimension n
$\mathbb{R}^{n\times m}$	Matrices of n rows and m columns
$\boldsymbol{\mathcal{O}}(n)$	the orthogonal matrix group of size n
SO(3)	the special orthogonal matrix group, rotation matrix
$\mathfrak{so}(3)$	its Lie algebra
SE(3)	the special euclidean matrix group, rigid transform matrix

Constants

$\mathbf{0}_{n\times m}$	zero matrix of size $n \times m$
$\mathbf{I}_n$	the identity matrix of size $n \times n$ or corresponding linear application.
$\mathcal{N}(\mathbf{u}, \boldsymbol{\Sigma})$	the Gaussian distribution of mean $\mathbf{u}$ and covariance $\boldsymbol{\Sigma}$

Operators

$\mathbf{X} \boxplus \delta\boldsymbol{X}$	retraction operator from tangent space to manifold
$\mathbf{A} \otimes \mathbf{B}$	the Kronecker product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$
$\dot{\boldsymbol{X}}$	the total time derivative of vector or matrix $\boldsymbol{X}$
∂_{A}	the partial derivative operator $\frac{\partial}{\partial \mathrm{Vec(A)}}$ with respect to the coefficients of Vec (A)
$\|\mathbf{x}\|_{\boldsymbol{\Sigma}}$	the Mahalanobis norm of invertible covariance $\boldsymbol{\Sigma}$: $\|\mathbf{x}\|_{\boldsymbol{\Sigma}} = \mathbf{x}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{x}$.
$\boldsymbol{\Sigma}^{\frac{1}{2}}$	With $\boldsymbol{\Sigma}$ a matrix: a square root matrix of $\boldsymbol{\Sigma}$
$[\mathbf{x}]_\times$	for $\mathbf{x} \in \mathbb{R}^3$ matrix of application with $\mathbf{y}$ $\mathbf{y} \mapsto \mathbf{x} \times \mathbf{y}$ (cross-product)
$e^x, \exp(x)$	depending on the context, will denote the scalar exponential or the matrix exponential, or the lie group exponential.
$\mathrm{Exp}_{\mathcal{M}}(\mathbf{x})$	For $\mathcal{M}$ a matrix Lie group, the "vectorized" matrix lie group exponential
$\mathrm{Log}_{\mathcal{M}}(\mathbf{X})$	For $\mathcal{M}$ a matrix Lie group, the "vectorized" lie group logarithm.

Symbols

$\mathbf{g}$	gravity vector
$\mathtt{R}$	Rotation matrix, element of SO(3)
$\mathbf{B}$	Magnetic field
$\nabla\mathbf{B}$	Magnetic field gradient as a 3×3 matrix
$\mathbf{v}$	Speed
$\mathbf{p}$	Position
$\mathbf{l}$	landmark position parameters (generally $[x, y, z]$ 3D position)
$\boldsymbol{o}$	pixel coordinate in an image
π	projection function of a camera as a function from $\mathbb{R}^3 \to \mathbb{R}^2$
$\boldsymbol{\xi}$	Pose (orientation and position) of an object.
$\boldsymbol{\xi}^{a\leftarrow b}$	Pose (orientation and position) of object b in object a frame
π^{-1}	"retroprojection" function of a camera $\mathbb{R}^2 \to \mathcal{S}^2$ or $\mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}^3$.

Table 1: List of reserved symbol

Introduction

This thesis deals with navigation, i.e., the capability to know the position of a mobile object while it moves relative to a reference frame. Navigation is an old issue and was initially closely related to maritime journeys. The difficulty arising from the navigation at sea was mainly because, for human sense, the sea does not provide any visible distinctive landmark. Scientists all around the world have thus used their creativity to find technical solutions to ease maritime navigation. Along the centuries, more an more sophisticated tools were developed to extend navigation capability to aerial, terrestrial, submarine, and space navigation and improve accuracy and reliability.

Two class of methods are used altogether in navigation. The first class is dead-reckoning, that estimate the displacement relative to a previous position at a particular time. The second is landmark-based navigation, which estimates the position of the object relative to known landmarks.

The most spectacular landmark-based system is undoubtedly the satellite positioning systems, launched in the second half of the 20th century. These systems provide a meter-accurate position all around the globe. This class of methods is called GNSS (Global Navigation Satellite System).

However, the requirements for navigation and positioning differs vastly from applications to another, and GNSS, despite its global nature, does not answer every application need.

One of these applications that will keep on growing in the next few years is *spatially aware computer applications*, whose a hard instance is the AR (Augmented Reality). AR requires a very high precision localization and orientation knowledge relative to its close, generally indoor, environment.

AR's ultimate goal is to seamlessly integrate, into the *real* world perceived, additional – computer-generated – information. To do so, AR technology tries to cheat on the subject senses to provide him with information in the most natural way for a human. Two senses are preferred for this purpose: sight, and sound. They are nowadays combined in most advanced AR experiences. AR enables new ways to interact with sophisticated computer systems in numerous fields: urban turn-by-turn navigation, surgery, facilities maintenance, it also enables new possibilities in arts, fashion, interior design, tourism, e-commerce, or gaming. This technology might even become, in the future, the main human-machine interface, superseding the standard (touch)-screen, mouse, and keyboard.

Compared to the traditional interface paradigm AR aims to simulate the *spatialization* of information in a way human will naturally understand. This means reprojecting object image as if they were entirely part of the 3D world and generating a sound signal which human ears can localize spatially. Being able to create such signal involves several technologies:

- one generating the information that the eyes and ears will understand (a headset with AR glasses and earphone)
- one modeling the real world sufficiently to spatialize *in a relevant way* the information (a mapping system)
- finally one allowing to sense the movement of the subject and position it in the environment (a localization system).

The work presented in this thesis focused exclusively on the latter.

Such a localization system must have the following properties:

- *Full 6*DOF *pose.* The spatialization needed for 3D virtual object reprojection requires knowing the position and orientation of the eye of the subject.
- *High-frequency.* A high enough frequency position refresh rate is required; otherwise, the user feels uncomfortable scattered movements. In practice, the framerate requirement is higher than video or animation movies, because of the more intrusive nature of AR.
- *Small-latency.* A large latency will cause the subject to be sick quickly. Human senses can already perceive few milliseconds of delay.

Figure 0.1: An quadcopter drone equipped with camera and inertial sensors for navigation purpose

- *Smoothness of estimation.* The localization system should give a smooth positioning estimate for avoiding gross artifacts on the reprojected image.

Moreover, for general public adoption there are also severe constraints on the technical solution:

- *Infrastructureless.* This provides easy installation of the system and does not require complex deployment.
- *Battery powered.* This provides higher movement freedom.
- *Light and small system.* In order to be carried by a human. Ideally being integrated into lightweight glasses.
- *Low price.* It should be affordable enough for mass market. This mainly constrains the usable sensor technology.
- *Robustness to dynamic and environment.* In order to cover the range of movement and location where the user would want to use such a system.

Interestingly, these properties and constraints are shared with another application: position estimation in mobile robotic. This is particularly valid for autonomous drone – as the one depicted on Figure 0.1: each point above also applies. In practice, the algorithm and systems employed in both applications (augmented reality and mobile drone) are very similar.

State-of-the-art localization systems for AR headset, principles and limitations

During the three years of the thesis, some headsets were released for mass-market by industrial companies. Some examples are given in Figure 0.2. The localization systems employed provide 6DOF at high frequency, small latency, and relatively accurate localization. The novelty of the headsets of the last years is their focus on infra-structureless position estimation.

Internally, these systems position tracking is based on the following principle: an IMU signal (acceleration and rotation speed) is integrated to predict a pose knowing the previous one, while visual measurements are used to correct this prediction, that otherwise would quickly drift. The details on how the image processing is done can vary greatly, and we will go through some example along the thesis.

Depending on technical details these processes are called VIO (Visual-Inertial Odometry) or SLAM (Simultaneous Localization and Mapping), and the system is broadly called VINS (Visual-Inertial Navigation System). More precisely, VIO names the process of reducing the drift of the IMU with short-term visual information. This is a pure *dead-reckoning* process where position errors will accumulate with time. In contrast, SLAM often refers to the simultaneous construction and storage of a map of the environment in computer memory. This map can then be reused by the tracking component to reduce to correct the drift of the VIO.

These tracking components are not perfect, however. They mainly rely on the quality of visual data, which highly depends on the environment and can not be guaranteed beforehand. These systems thus employ an advanced failure detection logic. They notify the user of these failures,

(a)

(b)

(c)

(d)

Figure 0.2: Example of products released during this doctoral work with AR capable localization without infrastructure. (**a**) High-end smartphone with precise 6DOF (position and orientation) localization capabilities and plane detection for augmented reality through the screen. (**b**) Virtual-reality headset and controllers. (**c**)-(**d**) Augmented reality headset with see-through screen. All these systems are equipped with cameras and IMU (Inertial Measurement Unit) for the 6DOF positional tracking. As noticed immediately, nowadays, headset systems are still bulky and invasive which limits their social acceptation.

which degrades the interaction. Typical failure cases occur when the environment is difficult to understand for the visual processing subsystem or when it does not provide enough information, for instance because of the absence of light, the presence of smoke, in a highly dynamic scene, etc. Besides, these systems are somewhat bulky, and battery duration is still an unsolved problem.

The novelty of our work: exploiting opportunistic magnetic field disturbances to complete the visual-inertial information

In this thesis, we propose a new sensor that could complete the VIO and solve some of its issues. In this document – by analogy to the IMU abbreviation – we will name this sensor a MIMU (Magneto-Inertial Measurement Unit).

The MIMU contains – in addition to gyrometers and accelerometers – a magnetometers array and has the capability to leverage the local disturbance of magnetic field to improve positional tracking. A technique that we call in this thesis MI-DR (Magneto-Inertial Dead-Reckoning).

We will explore how the addition of the MIMU can be used to improve the VINSs that are used in a AR context. In particular, we will focus on the improvement of the VIO, the dead-reckoning component of these systems.

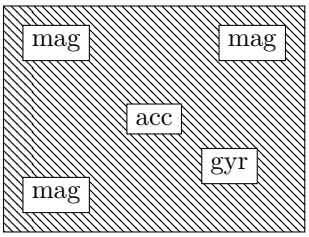


Figure 0.3: Diagram of a 2D MIMU sensor. Accelerometer (acc) and gyrometers (gyr) are completed by several magnetometers (mag). The relative positions of the sensors are precisely known.

General Contributions and Thesis Outline

The subject is investigated broadly, from the point of view of sensor choices, estimator choices, and theoretical aspects of the fusion estimator.

We study, describe, implement and evaluate dead-reckoning systems (hardware and software) that fuse information from a MIMU with information from visual sensors. By combining the MI-DR and VIO techniques, we show that we are able to improve the robustness of the dead-reckoning in scenario challenging for *both* MI-DR and VIO. We present results of two different sensor choices (depth sensor + MIMU and monocular camera + MIMU) and several sensor fusion algorithms. We attempt to derive the estimators from consistent mathematic framework and error modeling of each sensor, and we evaluate them on real data, captured with a prototype mounted specifically for this work. Finally, on the theoretical part, we also investigate fine estimator properties that are linked with the notion of consistency of the estimation, observability, and invariance.

This document is separated in two parts.

The first part presents general notions and a first attempt to fuse visual information with MIMU sensor. It goes from Chapter 1 to Chapter 3:

- The Chapter 1 comes back briefly on the core technology that provides dead-reckoning capabilities: inertial navigation. We present the formulation of inertial navigation problems through its historical use in high-end inertial navigation and describes how these equations can be leveraged with low-cost sensors.

- The Chapter 2 presents the MIMU and the MI-DR technologies. These technologies significantly improve low-cost inertial navigation. MIMU is at the center of this work: investigating the practicality and usefulness of such a technology compared/alongside to VIO algorithm is the main issue addressed in the thesis and, among the rich literature on VINS, its use is the main feature that distinguishes our work.
- Finally, the Chapter 3 describes a first loosely coupled approach of the fusion problem. We investigate fusion possibilities between an active depth sensor and the MIMU. This chapter mainly relates the work done during the first half of the doctoral work.

The second part of the thesis focus on a second sensor choice that is closer to the state-of-the-art in VINS. We investigate 6DOF dead-reckoning by *tight* fusion between a monocular camera and a MIMU. We study in this context two different estimation paradigms: optimization and filtering.

- Chapter 4 is a digest review of the related work about monocular VINS, SLAM and VIO. It is an an organized subjective selection of recent and interesting work in the field.
- Chapter 5 presents a mathematically sound way to integrate the MI-DR ideas into the bundle adjustment problem: an energy minimization formulation often used to solve for the positioning problem in robotic and SLAM community. We demonstrate the usefulness of this energy in an application to incrementally solve the optimization problem. We show the benefits of the fusion on real data.
- Chapter 6 presents an alternative way to fuse the same information through a less computationally intensive filtering approach. We show that the improvement of robustness observed in the previous chapter is still achieved within this framework.
- Finally, Chapter 7 focuses on theoretical aspects of the filtering approach and study its consistency property. This is done with the help of invariance theory from the theoretical automatic community. It is the most technical chapter of the document.

Publications relating the doctoral work

Parts of this thesis were also presented in the following publications:

- Caruso, D., Sanfourche, M., Le Besnerais, G., and Vissiere, D. (2016). Infrastructureless Indoor Navigation With an Hybrid Magneto-inertial and Depth Sensor System. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Alcalà de Henares

- Caruso, D., Eudes, A., Sanfourche, M., Vissiere, D., and Le Besnerais, G. (2017a). An Inverse Square-root Filter for Robust Indoor/Outdoor Magneto-visual-inertial Odometry. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo

- Caruso, D., Eudes, A., Sanfourche, M., Vissiere, D., and Le Besnerais, G. (2017b). Robust Indoor/Outdoor Navigation through Magneto-visual-inertial Optimization-based Estimation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver

- Caruso, D., Eudes, A., Sanfourche, M., Vissière, D., and Le Besnerais, G. (2017c). A Robust Indoor/Outdoor Navigation Filter Fusing Data from Vision and Magneto-Inertial Measurement Unit. *Sensors*, 17(12):2795

These contributions are all presented in Chapters 1 to 6, with more details than in the original publications. In particular, the Chapter 5 goes well beyond the conference paper [Caruso et al., 2017b]. The Chapter 7 is a late and new contribution on the subject that has not been published at the time of writing.

Part I

Antipasti: Magneto-inertial Dead-Reckoning and a First Fusion Approach with an Active Visual Sensors

Table of Contents

Chapter 1

General Notions About Inertial Sensors and Navigation

This chapter reviews briefly high end navigation solution in Section 1.1, then describes the inertial sensor model we will use in Section 1.2 and focuses in Section 1.3 on the mechanization equations of strapdown case, which is the only sensor configuration that can be implemented with the low-cost sensors we are interested in. Equations of Section 1.2 together with the error model briefly presented in Section 1.3 will be very often referred to in subsequent chapters.

1.1 Strapdown Inertial Navigation

1.1.1 The Navigation Problem and its Application Solved by INS

The problem of inertial navigation can actually be summarized in a very simple way: knowing at each time t either the *attitude* or the *rotational velocity* $\boldsymbol{\omega}(t)$, the *specific acceleration*[1] $\mathbf{a}(t)$, and the local value of the *gravitational field* $\mathbf{g}(\mathbf{p}(t))$, an INS (Inertial Navigation System) aims to estimate the evolution of the position, orientation and speed of an object it is attached to, with respect to another reference frame. In order to reach this goal, it proceeds through a pure integration of the kinematic movement equation. This integration process is called *dead-reckoning*.

The main advantage of these systems is that they do not rely on external references once the initialization point has been given to the INS, which makes them highly robust to external disturbances or sabotage, especially compared to architecture based localization systems, such as the GNSSs. For this reason, INS are extensively used in products relying critically on their position estimate such that aircraft, boat, submarines, missile, space-ship, satellites. Being critical for these military and industrial applications, INS have been extensively developed after the beginning of the second half of the 20th century.

These efforts made the field a very mature engineering domain which is largely understood and mastered by these industrial and academic experts. For instance, the complete mathematical model of earth navigation and how to discretize it in a computer program to do the estimation is described extensively in [Savage, 2000][2]. On the hardware side, accelerometers products are mainly based on the measurement of the movement a seismic mass (the measurement processes *per se* being very diverse) while gyroscopes are either mechanical (based on the conservation of angular momentum of a spinning mass), optical (based on Sagnac effect [Sagnac, 1914]), or vibrating mass (based on Coriolis effect).

In the vast majority of applications, however, the need for localization is not expressed in an inertial frame but in an *earth* anchored frame or as a relative position with respect to an object

[1] The specific acceleration is defined as the kinematics acceleration minus the local gravity vector. This is the only quantity measurable by an accelerometer. It can also be thought as the sum of the non-gravitational forces applied to the accelerometer. The fact that only specific acceleration is physically measurable strongly correlates the problems of attitude and position estimation. This is actually one of the reasons for the difficulty of doing inertial navigation.

[2] for the specific case of strapdown montage as defined in Section 1.1.2

fixed in the earth frame. The art of high-end inertial navigation is to correctly model the earth dynamic and its coupling with the estimation error.[3]

Yet, INS have some flaws that make their usage not fitted for many situations. These caveats emerge mainly from the fact that dead-reckoning accumulates even the sightliest error made at each instant of the trajectory. As a consequence, high-end INS are big, bulky and expensive, and still presents a limited but non-zero drift. Cheaper sensors are available, but their drift can be so gross that trajectory reconstruction is out of reach. For this reason, the main point of designing an INS is to choose a strategy to mitigate his drift with information from other sensors. Hence, from the point a view of an inertial navigation engineer, most of the present thesis boils down to address this unique issue, investigating different complementary sensors and algorithms.

Finally, there are some cases where inertial information is of weak interest. Consider the case of relative positioning with respect to an object moving with unknown and non-trivial dynamic. An instance of this problem can be the tracking the position of a pedestrian in a large mobile platform such as a boat. Pure exteroceptive sensor solution could, however, be of interest in such corner case.

1.1.2 Strapdown IMU

The IMU we use in this work are all of the *strapdown* kind described by [Savage, 2000]. In this kind of IMU, accelerometers and gyrometers are rigidly attached on a solid platform, in contrast to a gyrostabilized platform where accelerometer orientation is stabilized in the inertial frame. This has two implications: (i) the attitude is not given by the gyroscopes directly, but has to be integrated from the rotational velocity by the computer; (ii) the accelerometer measurement reading has to be rotated in the inertial frame using the current attitude estimate. The main flaw of strapdown montage was historically the additional pressure on the computer part of the INS caused by the need to integrate the rotational rate and the higher frequency of processing required to deal with the strong coupling of attitude estimate and translational drift. However, with today embedded computations power, this is hardly ever a problem.[4] Nowadays, the majority of IMUs are strapdown montage because they are easier to build mechanically, are smaller and lighter.

1.1.3 MEMS IMU and Inertial Navigation

Strapdown systems are of particular interest also because they can be implemented using only very cheap MEMS (MicroElectroMechanical System) chips. This allowed to put IMUs into a variety of mass market electronic systems, the best example being the smartphones, for which high-end products have gained some augmented reality capabilities.[5] The lack of accuracy of these sensors prevents their use for positioning, but the attitude can actually be well estimated if one assumes that accelerometers measure the gravity direction when the device is at rest (used as an inclinometer).

1.1.4 Example of Complementary Sensors

In order to correct inertial sensors integration, several complementary sensors have been investigated. The sensor modality depends highly on the application. For instance, on the high-end part of the spectrum, satellite rely on star tracker, boats are measuring speed relative to the water with speed logs, aircraft are using Pitot probes to measure their airspeed. The GNSS measurement can also complete all these systems, either using their output of position or using directly the phase of the signal received – a technique sometimes called GPS speed measurement. On these vehicle applications, a model of the motion dynamic can also provide valuable information. For pedestrian and robots navigation, where low-cost, light IMU prevails, we usually distinguish *outdoor* and *indoor* applications. Outdoor applications are easier as GNSS provides a solution for positioning with meter

[3] Which can lead to surprising results as, for instance, the famous 84 minutes period Schueller oscillation, that involves a coupling between the error of position and the direction of gravity in a spherical earth. [Savage, 2000]
[4] specially if comparing with the heavy visual processing involved in the next chapters
[5] Through companies frameworks *ARKit* or *ARCore* (respectively from Apple and Google)

accuracy and magnetometers can be used as a compass for heading. *Indoor*, the situation is more complicated: systems relying on an external infrastructure exist, for instance, based on purposely placed ultra-wideband emitters or Bluetooth beacon or based on an opportunistic signal such as WiFi. Visual sensors – [Li and Mourikis, 2013], [Leutenegger et al., 2015] – or LIDAR – [Zhang and Singh, 2015] – also provides a way for helping the navigation, without necessarily relying on an infrastructure. The magneto-inertial dead-reckoning technique, at the heart of this thesis, also provides valuable correction.

1.2 Inertial Sensor Model

In this section, we describe the mathematical model of sensors and the strapdown equations that will be used throughout this work.

1.2.1 From Raw Sensor Signal to Physical Quantity

1.2.1.1 Inertial Vector from 3 Inertial Single-axis

The tri-axis inertial sensors we use are actually built from three single-axis MEMS sensors. These single-axis sensors are designed to output a signal proportional to the quantity of interest (rotational velocity for gyrometers, acceleration for accelerometers). However, their models have to be calibrated individually before using them as measurements of acceleration or rotational velocity. We describe here the calibration model for MEMS accelerometers and gyrometers.

We model the signal output by a single-axis sensor x_m at time t with the following linear form:

$$x_m(t) = \mathbf{f}^\mathsf{T}\mathbf{x}(t) + b, \tag{1.1}$$

were $\mathbf{f}$ is a 3-components vector scale factor, and b a scalar bias of the single-axis. $\mathbf{f}$ encodes both the direction and the scale factor of the measurement.

Moreover, the combination of the three axis is never totally orthogonal because of the precision of the mechanical assembly. Assuming the three sensors are located at the same point, which is reasonable for inertial sensors integrated into a chip, the full tri-axe sensor measures actually:

$$\mathbf{x}_m(t) = \underbrace{\begin{bmatrix}\mathbf{f}_1^\mathsf{T}\\ \mathbf{f}_2^\mathsf{T}\\ \mathbf{f}_3^\mathsf{T}\end{bmatrix}}_{\mathbf{F}} \mathbf{x} + \underbrace{\begin{bmatrix}b_1\\ b_2\\ b_3\end{bmatrix}}_{\mathbf{b}} \tag{1.2}$$

Inverting this relation gives the relation between the measurement vector and the physical quantity of interest:

$$\mathbf{x}(t) = \mathbf{F}^{-1}\left(\mathbf{x}_m(t) - \mathbf{b}\right). \tag{1.3}$$

The calibration of an inertial tri-axe is the estimation of the quantities $\boldsymbol{F}^{-1}$and $\mathbf{b}$. As in practice, these coefficients can also be influenced by environmental factors such as the temperature, or even gravity direction; this effects also have to be taken into account. We call the application of (1.3) to raw signal sensor *calibration compensation* step. All the online estimators we develop afterward deal with these compensated measurement $\mathbf{x}$ as input and rely on a simplified model of the body and earth dynamic that we present in the next section.

1.3 Mechanization Equations

1.3.1 Flat-Earth Approximation

The modelization of inertial navigation on earth uses traditionally the following reference frames [Savage, 2015]:

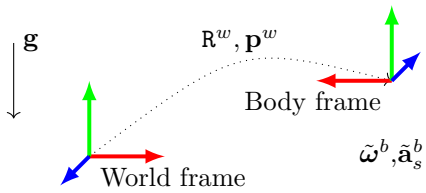


Figure 1.1: Reference frame used for writing the mechanization equation in this work. World frame is aligned with z direction pointing upwards and with random x and y direction. The measurement from accelerometers and gyrometers are given in the body frame.

- The *body* frame, b, which is the frame whose directions are parallel to the calibrated accelerometer sensitive axes.
- The *navigation* frame whose z axis is parallel to the local gravity direction, in the opposite direction. The other axis can be chosen freely depending on the application, for instance with x pointing towards the north at each time.
- The *earth* frame, which is a frame anchored to a point on earth, this frame is rotating with the earth.
- The *inertial frame*, which is the true inertial frame in which earth is rotating.

However, throughout this work, we will assume a non-rotating flat earth approximation that simplifies the reference frame definition along with mechanization equations. More precisely, we assume that:

- The *earth* frame is actually an – non-rotating – inertial frame.
- The direction of gravity does not change in this frame.

The first assumption neglects the rotational velocity of the earth, which is relevant if the gyrometers is not precise enough to measure it correctly. Within this first assumption, the second assumption is then valid only for a relatively constrained area around one position on earth. This assumption fits in practice our targeted pedestrian application well, where the area covered does not exceed a few square kilometers

Within these assumptions, inertial, earth and navigation frame can be defined to be the same at each time. In practice, we will call this frame the *world* frame, w. It is defined with the z-axis pointing upwards, its origin at the switch-on time of the IMU, and two other axis x and y chosen at random to form a direct frame.[6] The reference frame are depicted on Figure 1.1.

1.3.2 Continuous Model in World Frame

The full continuous equations in the general case – including all reference frame and Coriolis effect – can be found in [Savage, 2015, Sec.2] and are not given here.

With the previous approximation, these continuous time derivatives of inertial quantities boil down to:

$$\dot{\mathtt{R}}^w(t) = \mathtt{R}^w(t)[\boldsymbol{\omega}^b(t)]_\times \tag{1.4}$$

$$\dot{\mathbf{v}}^w(t) = \mathtt{R}^w \mathbf{a}_s^b(t) + \mathbf{g} \tag{1.5}$$

$$\dot{\mathbf{p}}^w(t) = \mathbf{v}^w(t) \tag{1.6}$$

Where $\boldsymbol{\omega}^b(t)$ is the *rotational velocity* of world frame versus body frame and $\mathbf{a}_s^b(t)$ is the *specific acceleration* measured in body frame, that is the acceleration of the body frame in world frame after removing the gravitational part of it. These quantities are precisely the one measured and sampled by a strapdown IMU.

[6] This randomness comes from the difficulty to measure heading.

1.3.3 Integration of the Model in the World Frame

Adapting the integral formulation of [Savage, 2015, Sec. 3] to the previous approximation, we get between time t_k and t_{k+1} (note that we drop hereafter the $X(t_k)$ notation for the more concise X_k):

$$\mathtt{R}^w_{k+1} = \mathtt{R}^w_k \widetilde{\Delta\mathtt{R}}_{kk+1}, \tag{1.7}$$

$$\mathbf{v}^w_{k+1} = \mathbf{v}^w_k + \mathbf{g}^w \Delta t + \mathtt{R}^w_k \widetilde{\Delta\mathbf{v}}_{kk+1}, \tag{1.8}$$

$$\mathbf{p}^w_{k+1} = \mathbf{p}^w_k + \mathtt{R}^w_k \mathbf{v}^b_k \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t^2_{ij} + \mathtt{R}^w_k \widetilde{\Delta\mathbf{p}}_{kk+1}, \tag{1.9}$$

This integration is exact – within our assumption – with $\widetilde{\Delta\mathbf{v}}_{kk+1}$ and $\widetilde{\Delta\mathbf{p}}_{kk+1}$ defined by the following continuous integrals:

$$\widetilde{\Delta\mathtt{R}}_{kk+1} \overset{\text{def}}{=} \Delta\mathtt{R}_k(t_{k+1}), \tag{1.10}$$

$$\widetilde{\Delta\mathbf{v}}_{kk+1} \overset{\text{def}}{=} \widetilde{\Delta\mathbf{v}}_k(t_{k+1}), \tag{1.11}$$

$$\widetilde{\Delta\mathbf{p}}_{kk+1} \overset{\text{def}}{=} \int_{t_k}^{t_{k+1}} \widetilde{\Delta\mathbf{v}}_k(\tau)\mathtt{d}\tau, \tag{1.12}$$

with the notation (1.13)

$$\Delta\mathtt{R}_k(\tau) \overset{\text{def}}{=} \mathbf{I}_3 + \int_{t_k}^{\tau} \Delta\mathtt{R}_k(s) \left[\boldsymbol{\omega}^b(s)\right]_\times \mathtt{d}s \tag{1.14}$$

$$\text{and } \widetilde{\Delta\mathbf{v}}_k(\tau) \overset{\text{def}}{=} \int_{t_k}^{\tau} \Delta\mathtt{R}_k(s)\mathbf{a}^b(s)\mathtt{d}s. \tag{1.15}$$

The strategy employed to compute these integrals will depend on trade-off between accuracy, implementation complexity, and computation time. Alternative goes from the simple Euler method to high order Runge-Kutta integration scheme. Note that [Savage, 2015] decomposes integrals (1.11)-(1.12) further in order to simplify their computation with assumption of piecewise constant acceleration and rotational velocity in the *body* frame, as well as to manage the approximation in case of non constant values.[7]

In practice, the INS algorithm computes these integrals from the measurement received by accelerometer and gyrometers, and the result is affected by the residual error of calibration compensation. The model of the error has to be known when designing Bayesian inference methods as will be done in this thesis; we thus describe one popular model that will be used in the thesis in the following section. The structure of this model will be discussed further and advantageously used in the optimization process described in Chapter 5.

1.4 Residual Error Models in Compensated Measurement

Even after the calibration compensation, the resulting vector measurement suffers from inaccuracy, that will propagate into the integrals computation (1.10)-(1.12). These inaccuracies have different

[7] The author introduces an angular vector ϕ, an delta speed η, and a delta position κ that are linked to ours such that:

$$\Delta\mathtt{R}_{kk+1} = \mathrm{Exp}\left(\boldsymbol{\phi}\right) \tag{1.16}$$

$$\widetilde{\Delta\mathbf{v}}_{k;k+1} = \left[\mathbf{I}_3 + \left(\frac{1-\cos\phi}{\phi^2}\right)[\boldsymbol{\phi}]_\times + \left(\frac{\phi-\sin\phi}{\phi^3}\right)[\boldsymbol{\phi}]^2_\times\right]\eta \tag{1.17}$$

$$\widetilde{\Delta\mathbf{p}}_{k;k+1} = \left[\mathbf{I}_3 + 2\left(\frac{\phi-\sin\phi}{\phi^3}\right)[\boldsymbol{\phi}]_\times + \frac{1}{\phi^2}\left(\frac{1}{2} - \frac{1-\cos\phi}{\phi^2}\right)[\boldsymbol{\phi}]^2_\times\right]\kappa \tag{1.18}$$

These expressions have the advantage that under the assumption of constant rotational velocity and acceleration in *body* frame, ϕ, η and κ stems from trivial integration. Whereas ours $\widetilde{\Delta\mathbf{v}}_k(t_{k+1})$ and $\widetilde{\Delta\mathbf{p}}_k(t_{k+1})$ derive from trivial integration under the assumption of constant acceleration *in world frame*.

physical sources: sensor mis-calibration or calibration aging, sensor bias random walk, sensor bias reset at switch on, ADC (Analog to Digitial Converter) quantization, MEMS thermodynamical white noise, etc.

In this study, we will use the following residual noise sensor model for gyroscopes and accelerometers.

$$\tilde{\mathbf{a}}_k^b = \mathbf{a}^b(t_k) + \mathbf{b}_a + \boldsymbol{\eta}_{\mathbf{a}}, \quad \boldsymbol{\eta}_{\mathbf{a}} \propto \mathcal{N}(0, \sigma_a^2 \mathbf{I}_3) \tag{1.19}$$

$$\tilde{\boldsymbol{\omega}}_k^b = \boldsymbol{\omega}^b(t_k) + \mathbf{b}_g + \boldsymbol{\eta}_{\boldsymbol{\omega}}, \quad \boldsymbol{\eta}_{\boldsymbol{\omega}} \propto \mathcal{N}(0, \sigma_g^2 \mathbf{I}_3) \tag{1.20}$$

$\boldsymbol{\eta}_{\mathbf{a}}$, $\boldsymbol{\eta}_{\boldsymbol{\omega}}$ are Gaussian noise corrupting the measurement and $\mathbf{b}_g$, $\mathbf{b}_a$ the biases of the sensors. The noise distribution is assumed to be isotropic, for sensor symmetry reasons. We assume the biases follow a 1st order Gauss-Markov model, which is, in practice, a common assumption in visual-inertial literature, as for instance in [Leutenegger et al., 2015]:

$$\dot{\mathbf{b}}_g(t) = -\frac{1}{\tau_g}\mathbf{b}_g + \eta_{bg} \tag{1.21}$$

$$\dot{\mathbf{b}}_a(t) = -\frac{1}{\tau_a}\mathbf{b}_a + \eta_{ba} \tag{1.22}$$

Generating noises η_{bg} and η_{ba} both satisfy:

$$\text{(zero mean)} \qquad \forall t, \qquad \mathbb{E}\left(\eta_{b\mathbf{x}}(t)\right) = \mathbf{0}_{6\times 1}, \tag{1.23}$$

$$\text{(no time correlation)} \qquad \forall t_1, t_2 \qquad \mathbb{E}\left(\eta_{b\mathbf{x}}(t_1).\eta_{b\mathbf{x}}(t_2)^\mathsf{T}\right) = \boldsymbol{\mathcal{W}}_{c\mathbf{x}}\delta(t_2 - t_1), \tag{1.24}$$

$$\text{with} \qquad \boldsymbol{\mathcal{W}}_{c\mathbf{x}} = \text{diag}(\sigma_{b\mathbf{x};c}^2 \mathbf{I}_3), \tag{1.25}$$

Where δ is the kronecker symbol : $\forall x \in \mathbb{R}^* \ \delta(x) = 0, \delta(0) = 1$. The units of τ_b, τ_a, $\sigma_{bg;c}$ and $\sigma_{ba;c}$ are respectively s, s, $\frac{\text{rad}}{s^2}\frac{1}{\sqrt{Hz}}$ and $\frac{\text{m}}{s^3}\frac{1}{\sqrt{Hz}}$ and are characteristics of the IMU. These parameters model the stochastic behavior of the inertial sensor. These can be determined by the Allan Variance method. A technical explanation of this method for application to optic fiber gyrometer is given in [IEEE, 1998] and a more digest review in [El-Sheimy et al., 2008].

The main advantage of this model compared to a pure random walk model (with $\tau_{\mathbf{x}} \to \infty$), is that the bias evolution model is bounded, which is more consistent with the *a priori* knowledge one generally assume on the residual bias after sensor calibration compensation.

Discretization of the evolution of biases leads to the following equations:

$$\mathbf{b}_{g\,k+1} = -\exp\left(\frac{\Delta t_{ij}}{\tau_g}\right)\mathbf{b}_{g\,k} + \boldsymbol{\eta}_{bg} \tag{1.26}$$

$$\mathbf{b}_{a\,k+1} = -\exp\left(\frac{\Delta t_{ij}}{\tau_a}\right)\mathbf{b}_{a\,k} + \boldsymbol{\eta}_{ba} \tag{1.27}$$

that can be used in a discretized filter for instance. The $\boldsymbol{\eta}_{\mathbf{x}}$ appearing in (1.26) and (1.27) will be then modeled as discrete random variables with Gaussian density $\mathcal{N}(0, \boldsymbol{W})$, $\boldsymbol{W}$ being computed as $(t_{k+1} - t_k)\boldsymbol{\mathcal{W}}_c$ [Simon, 2006, p. 231].

Chapter 2

Magneto-inertial Dead-reckoning

In this chapter we describe the principles and the history of magneto-inertial navigation. We present in Section 2.2 the MIMU *sensor and explain how local gradient of magnetic field is measured from a magnetometer array. The Section 2.3 presents how this gradient is used to infer information on the trajectory of the* MIMU *and presents fundamentals equations that will be used throughout the thesis. The Sections 2.4 and 2.5 shows results obtained by a dead-reckoning approach using only the* MIMU *data, and try to identify opportunities for fusing with visual sensors.*

2.1 Magneto-inertial Principles and History

In order to correct the drift of pure inertial navigation, the magneto-inertial dead-reckoning technique leverages local stationary disturbances. The general idea of the technique is the following: by modeling spatially the disturbances $\mathbf{B}(\mathbf{p}, t_1)$ in the vicinity of the device at time t_1 we can retrieve the displacement of the device between t_1 and t_2 by: (i) assuming a stationary disturbance, (ii) assuming that the function $\mathbf{p} \rightarrow \mathbf{B}(\mathbf{p}, t_1)$ is bijective *in the vicinity of the device position* at time t_1 and t_2. The second assumption can be seen as very strong for a general vector field, but as the field is a continuous physical quantity, the inverse function theorem states that provided a non-singular gradient at point $\mathbf{p}$, this relation is at least locally invertible. This condition will thus be fulfilled if the sensor is small enough and does not move too fast.

We draw in Figures 2.1 contour plots representing the logarithm of the gradient norm of the magnetic field in a room along with a point cloud of the environment as reconstructed by the VINS system of Chapter 6. The contour plot aligned with the point cloud shows that the three objects in the room generate a magnetic field perturbation around them. These perturbations are actually present everywhere indoor and are a valuable source of information if exploited with magneto-inertial technique.

This idea of using local stationary magnetic disturbance to help inertial navigation was first introduced in the seminal paper [Vissiere et al., 2007] that tried to use this information into a purely inertial dead-reckoning system, only using one magnetometer. The idea was then finally fully developed by [Dorveaux, 2011] that presented a non-linear observer fusing information from a magnetometer array with an IMU to reconstruct velocity. The estimator was proven to be asymptotically stable, and the authors showed experimental 2D trajectory results. Other first-hand details over the magneto-inertial navigation principles along with further indoor 3D trajectories can be found in the author Ph.D. thesis [Dorveaux, 2011]. Further theoretical results on observability were also discussed in [Batista et al., 2013] and an application to registration of indoor terrestrial laser scanner data has been presented in [Hullo, 2013].

Very recently, tremendous progress has been shown with the technology. The authors of [Chesneau et al., 2016] build a dead-reckoning EKF around the MIMU sensor extending ideas of [Dorveaux, 2011] with the aim of improving the dead-reckoning robustness to non-stationary fields. The same author also studied how to handle the non-observable heading problem in indoor facilities by leveraging the MIMU sensor characteristics in [Chesneau et al., 2017]. The authors show impressive results with final drift error typically of the order of 1% of trajectory length. Interestingly, this

recent progress came both from an algorithm improvement but also thanks to Sysnav's effort to master the design and hardware of the MIMU sensor.[1]

2.2 Hardware: the Strapdown MIMU

The MIMU hardware can be described as regular IMU extended with an array of magnetometers spatially distributed around the tri-axis accelerometer. The diagram on Figure 2.2 depicts one possible configuration. This section shows how to use a general configuration of sensors to retrieve the value and gradient of magnetic field.

2.2.1 Computing Magnetic Gradient from Magnetometers Network

The magnetic sensors of the MIMU are treated slightly differently than the inertial sensor: part of their raw measurement are reduced to a measure of the local gradient of magnetic field.

For each individual single-axis magnetometer the same model as inertial sensor is used:

$$B_{mi} = \mathbf{f_i}^\mathsf{T}\mathbf{B} + b_i, \tag{2.1}$$

where B_{mi} is the sensor signal and $\mathbf{f_i}$ the sensitivity vector and b_i the bias of the measurement.

The fundamental idea of using the array of magnetometers is a smoothness assumption on the value of the field inside the convex hull of the sensors positions. We can for instance assume that the field follows a law described by a truncated Taylor expansion around a central magnetometer so that:

$$\mathbf{B}(\mathbf{p}) \simeq \mathbf{B}(\mathbf{p}_0) + \nabla\mathbf{B}(\mathbf{p}_0)(\mathbf{p} - \mathbf{p}_0) + \dots \tag{2.2}$$

Where $\mathbf{B}(\mathbf{p}_0)$ and $\nabla\mathbf{B}(\mathbf{p}_0)$ are respectively the magnetic field and its gradient at the central magnetometer (thus labeled 0, see Figure 2.2). Note that in the present manuscript, we limit the development up to the first order terms to avoid complexifying notation, yet the method could theoretically cope with higher orders. This would even be necessary if the size of the array were significant compared to the typical distance of field variation.

Within this first order assumption, the magnetometer i placed at $\mathbf{dx_i}$ with respect to the center of the network measures:

$$B_{mi} = \mathbf{f_i}^\mathsf{T}\left(\mathbf{B}(\mathbf{p}_0) + \nabla\mathbf{B}(\mathbf{p}_0)\mathbf{dx_i}\right) + b_i \tag{2.3}$$

The online estimation algorithms will use as input these two quantities and not the individual magnetometers measurement. The following describes how they are computed from each magnetometers measurement:

Using identity (3) to vectorize the gradient matrix, the measurement equation equivalently writes:

$$B_{mi} = \left(\left[1, \mathbf{dx_i}^T\right] \otimes \mathbf{f_i}^T\right) \begin{bmatrix} \mathbf{B}(\mathbf{p_0}) \\ \mathrm{Vec}\left(\nabla\mathbf{B}(\mathbf{p_0})\right) \end{bmatrix} + b_i \tag{2.4}$$

Concatenating these equations for the entire array of n single-axis magnetometers yields:

$$\mathbf{B}_m = \underbrace{\begin{bmatrix} \left[1, \mathbf{dx_1}^T\right] \otimes \mathbf{f_1}^T \\ \vdots \\ \left[1, \mathbf{dx_n}^T\right] \otimes \mathbf{f_n}^T \end{bmatrix}}_{\mathbf{C_{mag}}} \begin{bmatrix} \mathbf{B}(\mathbf{p_0}) \\ \mathrm{Vec}\left(\nabla\mathbf{B}(\mathbf{p_0})\right) \end{bmatrix} + \mathbf{b} \tag{2.5}$$

[1] The reader will notice that the last two papers were published during the second and third year of the doctoral work presented in this report. One key aspect of the thesis work, which might not be reflected in the subsequent chapters, is the fact the author had to deal with an always improving MIMU sensors, which had some consequences of the fusing strategy tried and used.

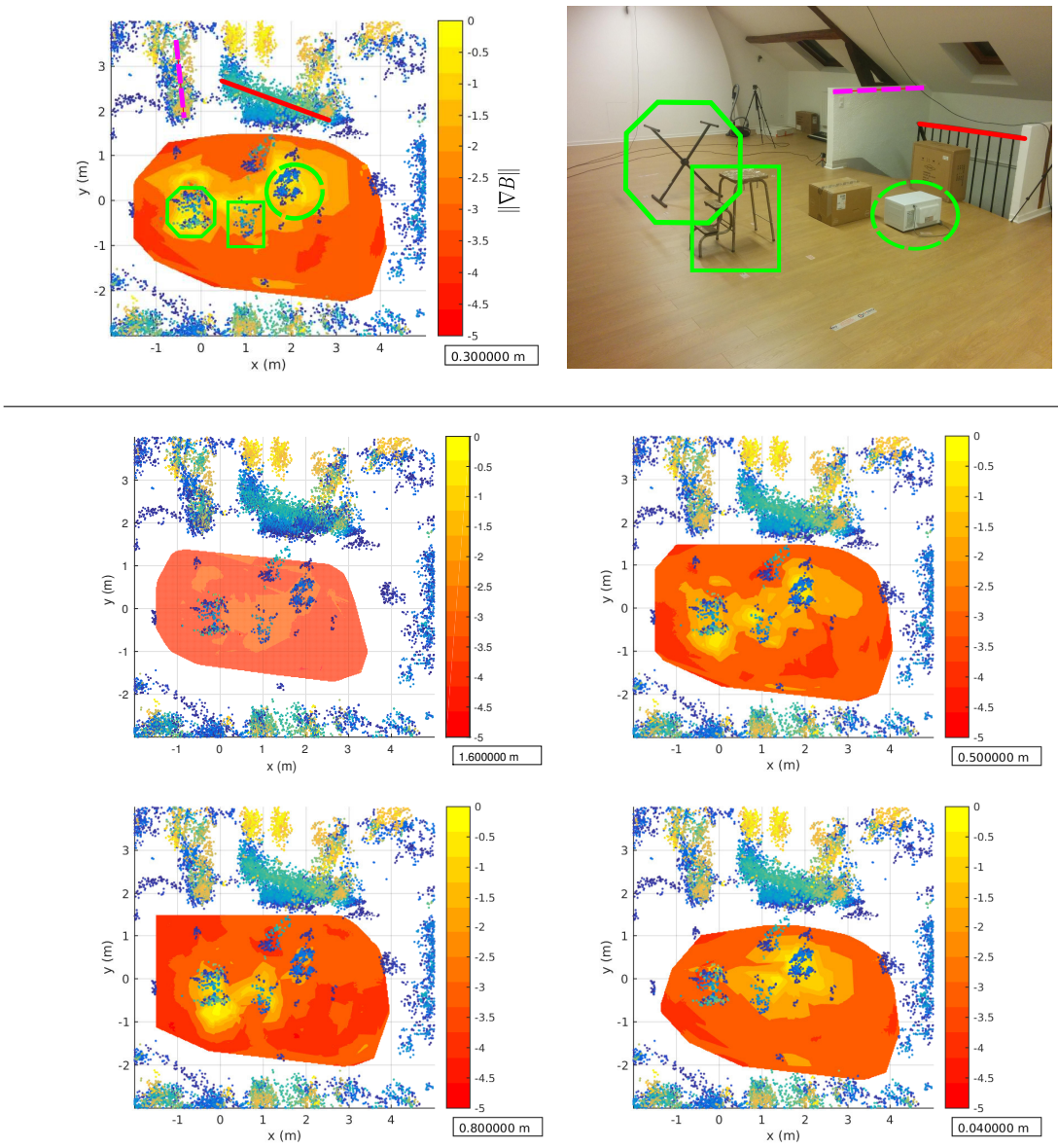


Figure 2.1: Volumetric map of the gradient of magnetic field sliced at different heights (see boxed values) of an indoor room scene shown in the top right image. The contour plot depicts the spectral norm of the gradient. The blue/yellow point cloud is estimated by the vision-based estimator of Chapter 6 and,in spite of its roughness, helps to localize the objects generating magnetic perturbations. The volumetric map is built with a moving MIMU sensor in a motion capture room and completed with a standard interpolation technique. Gradient unit are arbitrary.

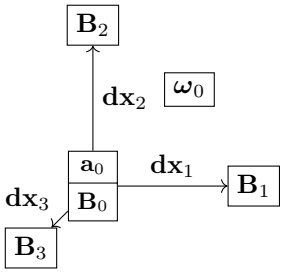


Figure 2.2: Schematic of the MIMU harware principles. It combines a central accelerometer and magnetometer with a network of peripherals magnetometers. Every sensors are rigidly attached, a gyrometers is used to measure the rigid body rotational velocity.

Figure 2.3: MIMU system built by the company Sysnav. The white box contains the sensorboard with MEMS accelerometer, gyrometers and the array of magnetometers. The black boxes contains a battery, control, and recording facilities.

The magnetic field and gradient are deduced by a least squares minimization as:

$$\mathbf{B}_0^*, \mathbf{Vec}(\nabla \mathbf{B}_0^*) = \underset{\mathbf{B}_0, \mathbf{Vec}(\nabla \mathbf{B}_0)}{\arg\min} \left\| \mathbf{C_{mag}} \begin{bmatrix} \mathbf{B}(\mathbf{p_0}) \\ \mathrm{Vec}\left(\nabla \mathbf{B}(\mathbf{p_0})\right) \end{bmatrix} + \mathbf{b} - \mathbf{B}_m \right\|^2 \tag{2.6}$$

This can be solved with a linear solution. The following conditions need to be satisfied though:

- $\mathbf{C_{mag}}$ and $\mathbf{b}_i$ have to be known. As for the inertial sensors, this is the role of the offline calibration.
- $\mathbf{C_{mag}}$ has to be full rank. This has to be guaranteed during the design of the magnetometer network.

The second condition can be partially relaxed taking into account particularity of the magnetic field.

Reduced geometrical constraint with Maxwell equation

The fact that $\mathbf{C_{mag}}$ has to be full rank imposes strong constraint over the geometry of the magnetometers network: in particular, their position vector have to form a basis of 3D space. Some of these constraints can be mitigated by exploiting the Maxwell equation of electromagnetism in the vacuum (without any charge or current density). These equations state differential constraint on the electric and magnetic field $\mathbf{E}(\mathbf{p}, t)$, $\mathbf{B}(\mathbf{p}, t)$:

$$\nabla . \boldsymbol{E} = 0 \tag{2.7}$$

$$\nabla . \boldsymbol{B} = 0 \tag{2.8}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2.9}$$

$$\nabla \times \boldsymbol{B} = \frac{1}{c} \frac{\partial \boldsymbol{E}}{\partial t} \tag{2.10}$$

The second equation states that the divergences (defined as the trace of the gradient) of the magnetic field is zero. The fourth one states that, in a stationary electric field, the curl of the magnetic field is also null, which translates directly into the symmetry of its gradient.[2]

Keeping that in mind, we can actually solve the following constrained minimization problem instead:

$$\begin{aligned} \mathbf{B}_0, \mathbf{Vec}(\nabla \mathbf{B}_0) = \underset{\mathbf{B}_0, \mathbf{Vec}(\nabla \mathbf{B}_0)}{\arg\min} & \left\| \mathbf{C_{mag}} \begin{bmatrix} \mathbf{B}(\mathbf{p_0}) \\ \mathrm{Vec}\left(\nabla \mathbf{B}(\mathbf{p_0})\right) \end{bmatrix} + \mathbf{b_i} - \mathbf{B}_m \right\|^2 \\ \text{s.t.} \quad & \mathrm{tr}\left(\nabla \mathbf{B}_0\right) = 0 \\ & \nabla \mathbf{B}_0 = \nabla \mathbf{B}_0^{\mathsf{T}} \end{aligned} \tag{2.11}$$

Which is equivalent to the reparametrized problem:

$$\mathbf{B}_0, \mathbf{Vec}(\nabla \mathbf{B}_0) = \underset{\mathbf{B}_0, \mathbf{Vec}(\nabla \mathbf{B}_0)}{\arg\min} \left\| \mathbf{C_{mag}} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathcal{P}_{\nabla \mathbf{B}} \end{bmatrix} \begin{bmatrix} \mathbf{B}_0 \\ \boldsymbol{g}_{\mathbf{B}} \end{bmatrix} + \mathbf{b_i} - \mathbf{B}_m \right\|^2 \tag{2.12}$$

Where $\boldsymbol{g}_{\mathbf{B}} \in \mathbb{R}^5$ and $\mathcal{P}_{\nabla \mathbf{B}} \in \mathbb{R}^{9 \times 5}$. $\mathcal{P}_{\nabla \mathbf{B}}$ is defined as the matrix of the application generating the vectorized gradient matrix from minimal gradient coordinates:

$$\begin{array}{ccc} \mathbb{R}^5 & \to & \mathbb{R}^9 \\ \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{pmatrix} & \mapsto & \mathrm{Vec}\left(\begin{pmatrix} g_1 & g_2 & g_3 \\ g_2 & g_4 & g_5 \\ g_3 & g_5 & -g_1 - g_4 \end{pmatrix} \right) \end{array} \tag{2.13}$$

[2] In practice, considering the electric field as stationary is often a valid assumption: the term $\frac{1}{c} \frac{\partial \boldsymbol{E}}{\partial t}$ can be neglected, as the inverse of the speed of light appearing in factor is very small; signal with frequency high enough to bother us would not even be in the bandwidth of the sensor we use.

This matrix will be used at several places over the document, mainly to propagate noise from a magnetic gradient to an expression in which it appears. In this last formulation, only the matrix $\mathbf{C_{mag}}\begin{bmatrix}\mathbf{I}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathcal{P}_{\nabla\mathbf{B}}\end{bmatrix}$ has to be invertible. This allows more freedom in the geometrical configuration of the sensor. It makes it possible to reduce the size of the magnetometer array along one dimension and to build a planar sensor as pictured in Figure 2.3.

2.2.2 A Word on Calibration of the MIMU Sensor

As for inertial sensors, the $\mathbf{f}_i$ and $\mathbf{b}$ parameters are calibrated by an offline process. This process involves recording the MIMU response to a uniform field under different orientations. The scale factor and biases parameters are found as the ones that make the output norm invariant by rotation. Simultaneously, sensors are registered in the same reference frame by using the fact that every single-axis sensor measures a projection of the same uniform field.

The calibration algorithm is described in more details in the two publications [Dorveaux et al., 2009, 2010], alternatives methods have also been proposed in [Renaudin et al., 2010]. Contrarily to inertial sensors, magnetometers are exteroceptive sensors: they measure a physical field related to the environment, and not to their own motion. For this reason, they need to be calibrated in their final environment: (see [Gebre-Egziabher et al., 2001]). In our particular case magnetometers calibration is essential, because, contrarily to inertial sensors, we will assume in the following that the calibration compensated magnetometers are *not* residually biased, so that we do not need to estimate magnetometers biases in the online process.

2.2.3 Compensated Sensor Noise Model

We will assume compensated measurements are corrupted by Gaussian noises $\boldsymbol{\eta}_{\mathbf{B}_k^b}$, $\boldsymbol{\eta}_{\nabla\mathbf{B}_k^b}$ such that:

$$\tilde{\mathbf{B}}_k^b = \mathbf{B}_k + \boldsymbol{\eta}_{\mathbf{B}_k^b} \tag{2.14}$$

$$\mathrm{Vec}\left(\nabla\tilde{\mathbf{B}}_k^b\right) = \mathrm{Vec}\left(\nabla\mathbf{B}_k\right) + \mathcal{P}_{\nabla\mathbf{B}}\boldsymbol{\eta}_{\nabla\mathbf{B}_k^b}, \tag{2.15}$$

where $\boldsymbol{\eta}_{\mathbf{B}_k^b} \propto \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\mathbf{B}}\right)$, $\boldsymbol{\Sigma}_{\mathbf{B}} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\eta}_{\nabla\mathbf{B}_k^b} \propto \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\nabla\mathbf{B}}\right)$, $\boldsymbol{\Sigma}_{\nabla\mathbf{B}} \in \mathbb{R}^{5\times5}$. The units of $\boldsymbol{\eta}_{\mathbf{B}_k^b}$ and $\boldsymbol{\eta}_{\nabla\mathbf{B}_k^b}$ are respectively Gauss and Gauss per meters. The covariances can be deduced from magnetometers white noise and calibration of uncertainty corrupting $\mathbf{C_{mag}}$.

2.3 Magneto-inertial Dynamical Equation for Dead-Reckoning

This section describes the fundamental equation that will be used throughout this work and also introduces some new notations common to the chapters Chapters 5 to 7.

2.3.1 Continuous Model

With ambient magnetic field described as a general function of space and time: $(\mathbf{p}, t) \mapsto \mathbf{B}^w(\mathbf{p}, t)$, the measurable magnetic field by of a non-rotated sensor that follows a path $\mathbf{p}(t)$ writes at each time $\mathbf{B}^w(\mathbf{p}(t), t)$ and its evolution is expressed with the total derivative $\frac{d\mathbf{B}^w(\mathbf{p},t)}{dt}$. Using the chain rule, one has:

$$\left.\frac{\mathrm{d}\mathbf{B}^w(\mathbf{p}(t),t)}{\mathrm{d}t}\right|_{(\mathbf{p}(t),t)} = \left.\frac{\partial\mathbf{B}^w(\mathbf{p},t)}{\partial\mathbf{p}}\right|_{(\mathbf{p}(t),t)} \left.\frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t}\right|_t + \left.\frac{\partial\mathbf{B}^w(\mathbf{p},t)}{\partial t}\right|_{(\mathbf{p}(t),t)} \tag{2.16}$$

which can be noted, introducing the gradient notation and recognizing the velocity in world frame, as:

$$\left.\frac{\mathrm{d}\mathbf{B}^w(\mathbf{p}(t),t)}{\mathrm{d}t}\right|_{(\mathbf{p}(t),t)} = \nabla\mathbf{B}^w(\mathbf{p},t)\mathbf{v}^w(t) + \left.\frac{\partial\mathbf{B}^w(\mathbf{p},t)}{\partial t}\right|_{(\mathbf{p}(t),t)} \tag{2.17}$$

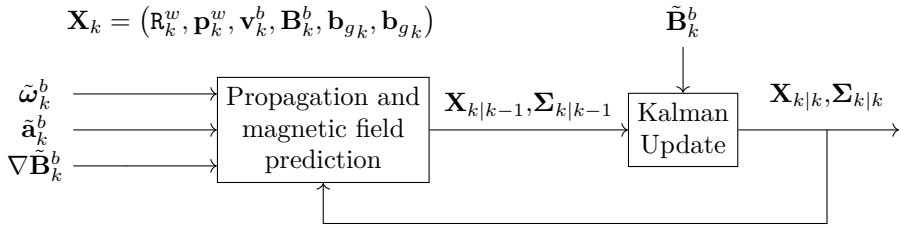


Figure 2.4: Diagram of the data flow of the MI-DR EKF filter of [Chesneau et al., 2016]

which states that the variation of the field seen by the magnetometers is related to the time derivative of the magnetic field in *world* frame, its spatial gradient, and the velocity of the magnetometers also in world frame. Note that the gradient notation here denotes the perfect gradient of the field at $\mathbf{p}$ of which the measurement as expressed in Section 2.2.1 is only an approximation. Assuming a stationary field, and dropping the explicit field dependence on time and position, we are left with the following differential equation:

$$\dot{\mathbf{B}}^w = \nabla\mathbf{B}^w\mathbf{v}^w, \tag{2.18}$$

In a strapdown setup, the gradient is measured in body frame so that we prefer to express the equation in the following way:

$$\dot{\mathbf{B}}^w = \mathtt{R}^w\nabla\mathbf{B}^b{\mathtt{R}^w}^\mathsf{T}\mathbf{v}^w, \tag{2.19}$$

or even, with the magnetic field or speed expressed in body frame:

$$\dot{\mathbf{B}}^b = -\left[\boldsymbol{\omega}\right]_\times \mathbf{B}^b + \nabla\mathbf{B}^b\mathbf{v}^b, \tag{2.20}$$

2.3.2 Integration of the Model and its Discretization

The continuous model can be integrated between two instant with general numerical integration method or used in a continuous Kalman filter formulation. One way to integrate the differential equation will be presented in Chapter 5 when dealing with preintegrated measurement.

2.4 Navigation Performance, Limits, and Discussion

2.4.1 The Magneto-inertial Dead-Reckoning Filter

The straightforward way to make use of the magnetic equation (2.20) is to build an EKF (Extended Kalman Filter) around it. One possible filter dataflow simplified from [Chesneau et al., 2016] is depicted on Figure 2.4. In such filter, the discrete state is:

$$\mathbf{X}_k = (\mathtt{R}_k^w, \mathbf{p}_k^w, \mathbf{v}_k^b, \mathbf{B}_k^b, \mathbf{b}_{g_k}, \mathbf{b}_{g_k}) \tag{2.21}$$

The magnetic field in body frame is included into the state, even if it is directly measured by the MIMU: its direct measurement is used to correct the Kalman filter with the trivial measurement function:

$$\mathbf{h}(\mathbf{X}_k) = \mathbf{B}_k^b \tag{2.22}$$

In turn, equation (2.20) combined with the IMU discrete prediction (1.7)-(1.9), Page 15, are used to compute state propagation.

2.4.2 Results of Pure MI-DR

In favorable magnetic environment, MI-DR shows accurate trajectory results with MEMS sensors. Figure 2.5 shows results of the MI-DR, a full featured filter implementation of [Chesneau et al., 2017]

on a pedestrian competition at the Indoor Positioning and Indoor Navigation (IPIN) conference 2016. The metric used for ranking in the competition was the third quartile of translational errors at control points. On trajectory of Figure 2.5, its value is 2.32m and can be read on the histogram Figure 2.5e.

2.4.3 Validity of MI-DR Hypothesis and Failures Mode

The main failure modes of the MI-DR are related to the assumption made about the magnetic field and how its gradient is measured. Three major failure cases can be identified:

1. the magnetic field gradient can be singular for an extended period along the path. Outdoor, this is the general case, as illustrated Figure 2.6 that shows results of the MI-DR filter on an outdoor/indoor trajectory recorded with our monocular/MIMU setup. The blue part of the curves denotes area with low-gradient, for instance, in the outdoor part. However, this can also happen locally indoor. In this case, the velocity disappears from the field propagation equations. This is a problem, especially for low-end IMUs targeted by the present work, that are unable to keep a consistent positioning without relatively high-frequency correction. Because of the necessarily limited accuracy of the gradient measurement, the field needs only to be *almost* singular in order these effects to impact MI-DR. The real quantity to monitor to detect this case is, therefore, the gradient magnitude related to the accuracy of its measurement.

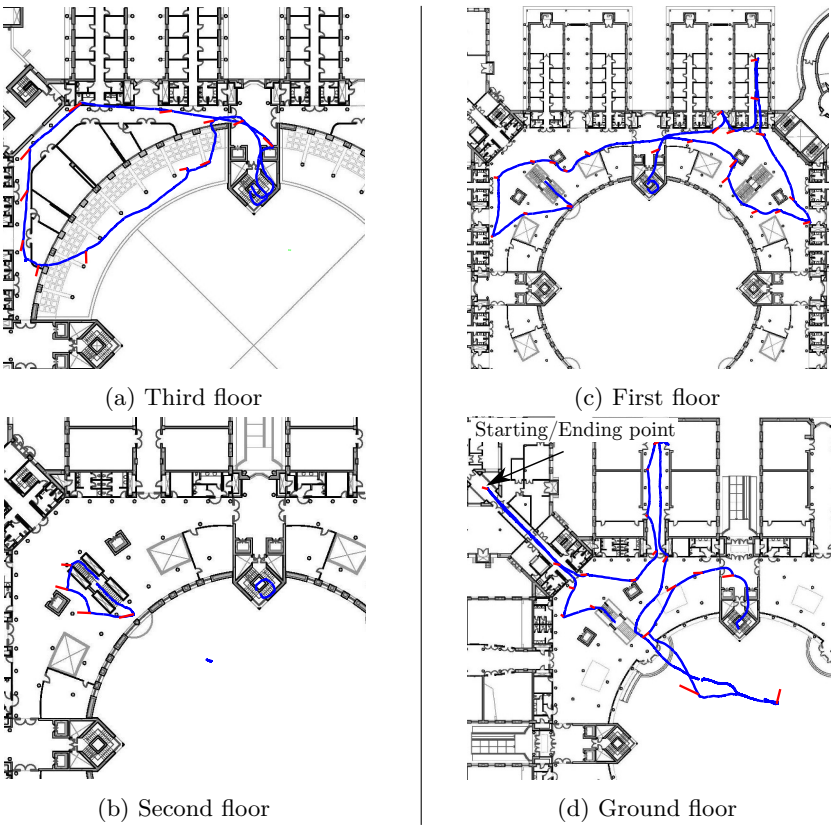
2. the field spatial higher order derivatives can be very strong close to ferrous materials. This has two effects: (i) if the higher order effects are strong in the volume occupied by the array of magnetometers itself, the spatial discretization when computing gradient will not be able to represent the field correctly in the vicinity of the sensor, leading to integration error in the computation of (2.20); (ii) if the higher orders cannot be neglected over the distance between positions at which two consecutive magnetometers sample occurred, the field value predicted by the propagation step of the filter will be corrupted by errors. These effects will also depend on the way the integrals are actually computed. [Dorveaux, 2011, Section 3.4] provides some insight on these issues on a one-dimensional toy case. Solutions involve mainly low-level signal processing and sensor design issues that are not discussed further in this thesis.

3. The field in practice is not stationary. This fundamental assumption is the main practical limitation of the method. Luckily, the high-frequency components used for telecommunication (radio, WI-FI or GSM network) are far out of the bandwidth the pedestrian dynamic involves, and does not perturb MI-DR technique. Yet, a general indoor environment can break this assumption in a various number of expected and unexpected way. Some examples are non-static ferrous material structures due to elevators, electric engines, metallic coins close to the sensors in the pocket of the pedestrian, etc. Note however that progress was made recently to relax this stationarity assumption: in the work of [Chesneau et al., 2016], the authors estimate the instationarity of known frequency arising from power line interference along with the position and orientation of a MIMU sensor.

Let us briefly comment on the difficulties associated with these failure modes. The first two cases are easily detected with instantaneous measurement[3] at time k respectively with a threshold on the smallest singular values of the gradient and with the residual of the least squares optimization (2.11). The last case cannot be detected solely from raw measurement at time k, which is problematic. In particular, raw Kalman filtering is very sensitive to such wrong models, and alternative estimation techniques have to be found.

One of the simplest ways to increase the filter robustness is to deactivate the magnetic feedback on the state when environmental conditions are *explicitly* considered as bad. However, this can

[3] Assuming we are not in, highly unlikely, pathological cases where the first order model of magnetic field fits well the network array measurement but does not represent the field in between sensors because of higher orders.

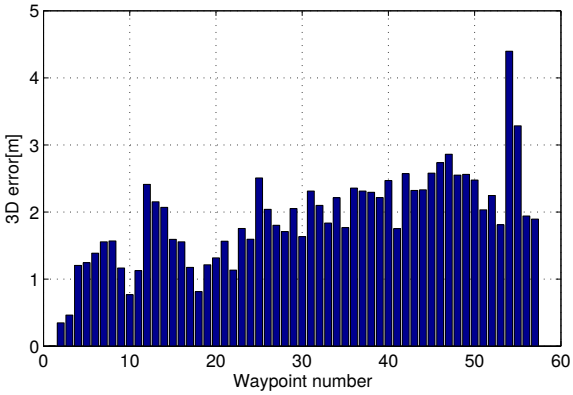


(a) Third floor

(c) First floor

(b) Second floor

(d) Ground floor



(e) Error histogram for all control point.

Figure 2.5: Results of the MI-DR of [Chesneau et al., 2017] on the competition track of IPIN 2016. Blue trajectory is the estimate, red line are error at control point. The histogram represents the translational error distribution at control points. Images and results are from Chesneau et al.

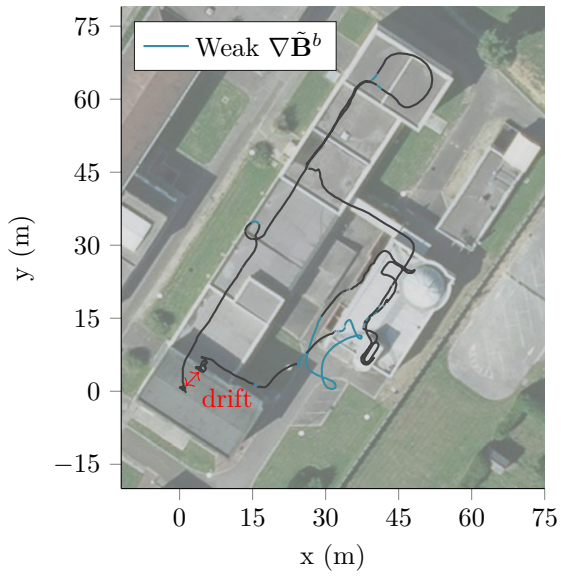


Figure 2.6: Trajectory reconstructed by MI-DR of [Chesneau et al., 2017] in an indoor/outdoor trajectory. Strong drift appears in outdoor part.

be thought defeating the purpose of magneto-inertial systems: since magnetic information can be unusable for an eventually extended period, IMU integration error would thus become significant. A complete navigation solution would necessarily require either a very high-performance IMU, or either additional exteroceptive sensors to handle these corner case.

2.5 Conclusion and Opportunities for Fusing with Visual Sensors

This section exposed basics of magneto-inertial dead-reckoning technique and discussed sensors, models, performance, and limits. We concluded that, even if MIMU systems can reach good accuracy in its nominal conditions, a 100 % available system would, however, need additional sensors, especially if aiming for full available augmented-reality localization application. That being said, this conclusion can actually be made for a lot of alternative infrastructure-less positioning solutions for AR, particularly for those being majoritarily developed the last few years: monocular or stereo VINS or depth-sensor based SLAM.

In fact, there are some reasons why fusing MIMU based navigation with these alternatives would be of practical interest:

Reducing the drift by unit of time Visual sensors are often used in a SLAM context, where a map of the environment is maintained and used for the localization. The best examples in literature are found in [Klein and Murray, 2007] [Strasdat et al., 2011] [Engel et al., 2014a] [Whelan et al., 2015] [Mur-Artal et al., 2015]. In these systems, drifts occur during the construction of the map and these methods can be drift free once the map of the operation area is built. This is in contrast to the dead-reckoning approach where drift will occur proportionally to the time.

Increasing operating range to outdoor scenes Visual-based systems have the benefits to also work reliably outdoor. This is in contrast with MI-DR which is not able to cope with the low gradient magnitude of general outdoor scenes.

Increasing operating range to dark scenes Analogously, passive visual-based systems struggle with low light areas or HDR (High Dynamic Range) environment mainly because of the limited dynamic of the camera sensors. In contrast, MI-DR performance does not depend on the lighting conditions. Active vision sensors such as depth sensor do not suffer from dark scenes but have a very limited range.

Providing more information for outlier rejection In a visual-magneto-inertial setup, accelerometers and gyrometers can nearly be always trusted as they virtually do not depend on environment assumptions. Advanced VINS estimator have logic to reject the error of their model (outliers) that is based on the integrity of inertial sensors. These heuristics often relies on a prior on the motion estimates fed with IMU and previous *inliers* sensor measurements. ([Civera et al., 2010]). The corruption of this prior with outliers would have dramatic consequences on the future rejection. Using multiple sensor modalities improves this prior, and in turn, can make outliers measurement detection more powerful. As a result, in practice, a consistent MI-DR filter would helps such outlier detection logic.

Reducing the power consumption of visual-based navigation Visual SLAM or odometry are generally not efficient from a power consumption point of view. This is mainly because of the need for heavy computation – from dense image processing to optimization based SLAM algorithm. More advanced methods developed in academia even involve GPGPU (General-purpose Processing on Graphics Processing Units) computing and depth sensors that are likely to be too demanding for the targeted embedded purposes. In contrast, MI-DR is based on a few MEMS sensors on a filtering framework with a low-dimensional state – the simplified filter of 2.4.1 has a state of dimension 18. A full-featured MI-DR has only a few more states, which is way less than state-of-the-art VINS filters. It is likely that the MIMU hardware could be even fully integrated on a chip in order to reduce its consumption further. In our opinion, MIMU then offers opportunities to reduce visual-based navigation power consumption by leveraging the quality of speed estimate to reduce the need for high frame-rate from the camera. Visual-based navigation power efficiency is mainly an industrial issue, and few academic papers targeted low-power systems explicitly. Some recent works dealing with the subject are described in [Zhang et al., 2017c], [Boikos and Bouganis, 2017] and [Hong et al., 2014].

Chapter 3

A first grasp of the fusion problem

This chapter introduces the problem of fusing a MIMU *sensor with visual sensors through a first example. The Section 3.1 describes the camera hardware we will use, and some prerequisites on calibration, harmonization, and synchronization of sensors. Then Sections 3.2 to 3.6 describe a first example of an estimator fusing the* MI-DR *techniques with a depth image alignment technique designed to work with commercial grade depth sensor. This estimator is tested on different difficult scenarios with data from real sensors, showing higher robustness compared to depth sensor based or* MIMU*-based method.*

This chapter has been the subject of the first year of the doctoral work in was the object of a communication at the International Indoor Positioning and Indoor Navigation (IPIN) 2016 [Caruso et al., 2016].

3.1 Hardware, Calibration and Synchronization Prerequisites

Throughout this thesis, the MIMU sensor from the company Sysnav depicted in Figure 2.3 is rigidly mounted with either conventional or depth cameras. In the present chapter, we use a commercial RGBD (Red-Green-Blue-Depth) sensor – the Asus Xtion Pro – providing a depth image registered with a RGBD image. The hardware is depicted in Figure 3.1. The second part of the thesis will focus primarily on the monocular case, where the vision sensor is a simple grayscale camera.

The use of the information provided by these different sensors requires identifying beforehand some parameters of the combined hardware, namely the camera intrinsic calibration and the spatial harmonization of the camera and MIMU sensor. An accurate fusion also relies on accurate timestamping of all the data. Thus, a synchronization strategy must be implemented.

This section gives some details about the camera model we employ, its calibration, the harmonization of sensors, and how we handled synchronization in the hardware prototype developed.

3.1.1 Camera Models

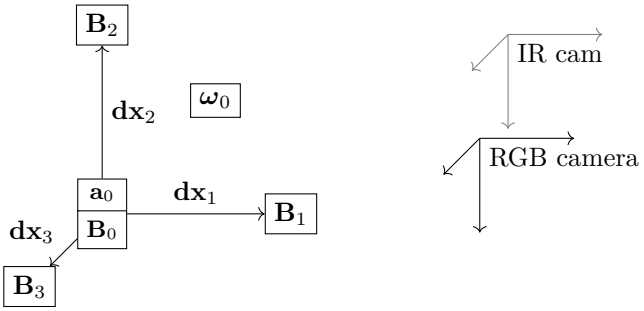
A camera captures the projection of a 3D environment onto a 2D image. It is associated to a projection functions $\pi : \mathbb{R}^3 \to \mathbb{R}^2$, that projects a 3D point *in camera frame* to a 2D location on the image.

The geometrical calibration process aims to identify this function precisely.

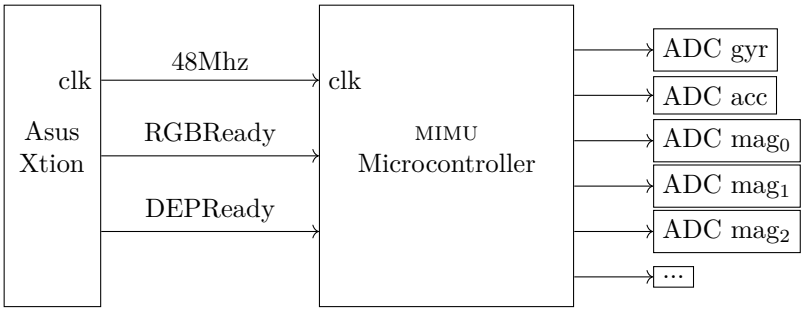
The projection function π is commonly searched in a parametrized family of function which depends on the a priori knowledge on the camera projection. The most commonly used family of projection functions is the *pinhole projection model* (Figure 3.2), that can be used for low field-of-view imaging device (up to roughly 70 deg) and that writes

$$\pi_{\text{pinhole}} : \begin{array}{ccc} \mathbb{R}^3 & \to & \mathbb{R}^2 \\ \mathbf{l}^c & \mapsto & \begin{bmatrix} f_x \frac{\mathbf{l}^c_1}{\mathbf{l}^c_3} + c_x \\ f_y \frac{\mathbf{l}^c_2}{\mathbf{l}^c_3} + c_y \end{bmatrix} \end{array} . \tag{3.1}$$

(a)



(b)



(c)

Figure 3.1: Hybrid sensor combining MIMU + RGBD camera used in this chapter. (a) Views of the system; (b) schematic view of the coordinate frames at play; (c) Schematic of electronic of the prototype built for the experiment of this chapters. Note that MIMU microcontroller is cadenced with Asus Xtion clock (clk arrow). This simple hack allows timestamping consistently Asus Xtion image with respect to MIMU digitization of sensors.

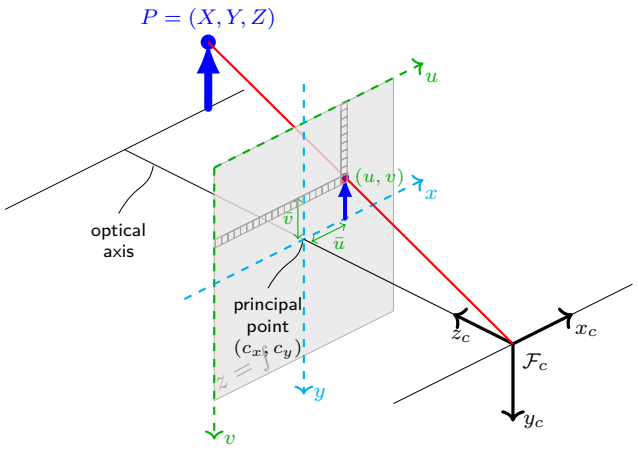


Figure 3.2: Pinhole camera model (from OpenCV documentation).

This family of function is parametrized by the focal length f_x, f_y and the position of optical center c_x, c_y. This model is handy for geometrical image processing: its main feature is that lines in the 3D world projects into lines in the image coordinates. Such property has interesting application, for instance for designing fast stereo matching algorithms.

Nonetheless, this model rarely fits practical lens and a generic camera image generally exhibits *distortion* compared to this idealized model. Literature suggests several ways to model these distortions, with different parametrizations. In visual SLAM or odometry, the distortion correction is often seen as a pre-processing step: first, the image is numerically undistorted – which involves pixel interpolation – then the SLAM algorithm is run on these undistorted images using an ideal pinhole assumption. However, distortion correction techniques have some limits: first, The validity of the obtained pinhole model is limited to one half-space in front of the camera, with a singularity on the plane $z = 0$; secondly, they involve substantial interpolation if the distortion coefficients are high. It is possible to use other parametric camera models dedicated to wide field-of-view cameras, with interesting benefits for visual navigation, as demonstrated in our previous work [Caruso et al., 2015].

Being aware of induced limitations, we will nevertheless, for the experimental part of the thesis, use the distortion pre-correction paradigm. Yet we will try to present equations using a generic projection function π, so that the algorithm described could be formally applied to a broader class of sensors.

We will also use the notation π^{-1} for the "retroprojection" function, either mapping pixel point coordinate onto bearing vector $\pi^{-1} : \mathbb{R}^2 \to \mathcal{S}^2$ or either mapping pixel coordinate and depth to 3D point location in camera frame $\pi^{-1} : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}^3$ depending on the context. Note, that it is a slight abuse of notation, because π as defined above, is not invertible.

A camera model should also ideally include the temporal behavior of the camera, such as exposition time, or rolling shutter effects. In practice, we observed that ignoring these temporal effects may degrade the quality of the position estimate.

3.1.2 Camera Calibration Process

The simplest way to calibrate a camera is to use planar checkerboard pattern [Zhang, 2000]. The idea is to minimize the reprojection error from checkerboard corners into a set of images viewing the checkerboard under different points-of-view. This minimization is done over the full model parameters, including distortion, and, in doing so, sets the camera reference frame. Special care must be taken to cover the whole 3D field-of-view. The technique using planar targetboard were presented in [Zhang, 2000]. We have used here the implementation of the Kalibr calibration

toolbox ([Maye et al., 2013]).

Remarks: Camera calibration can also refer to a photometric calibration of the imaging system, which is necessary for some SLAM methods relying on photoconsistency (see [Engel et al., 2018]). In this thesis, we see a camera as a pure and ideal geometrical imaging device and disregard the photometric properties of the camera, hence ignoring effect such as focus blur, chromatic aberration, etc.

3.1.3 Extrinsics Calibration and Camera/Imu Synchronization

In order to fuse the visual information with the MIMU sensor precisely, the transformation from camera frame to the MIMU frame is required. This transform is generally referred as the extrinsic camera transform. Two strategies are employed in the literature. Some authors estimate this transform offline (prior to the use of the system) while other authors estimate the transform online (during the run of the position estimation algorithm). We mainly focused on the first strategy in this work.

For offline estimation of the camera/IMU transform, we again relied on the Kalibr calibration toolbox from ETHZ. The calibration problem is expressed as a large minimization problem that estimates the trajectory of the accelerometers in the checkerboard frame along with the extrinsic camera transformation. The error terms of the optimization stem from the accelerometers reading, the gyrometers reading, the targetboard corners detection and the stochastic model of the biases. The originality of the toolbox is that it expresses the position as a continuous function of time, leveraging Bspline expressiveness. In this framework, a rotational velocity relates directly to the "orientation spline" derivative, while the acceleration relates to the "position spline" second derivative.

Additionally to spatial calibration, the camera and the MIMU also need to be synchronized. For practical reasons, in the considered prototype, the imaging sensor and the MIMU are not fully integrated, their driving electronics are not aware one of another. Hence, the internal timestamps given by the two sensors are out-of-sync and require correction. If we assume that the quartz clocks driving their electronics are synchronous – which is the case for the hardware used in this chapter thanks to an electronic hack, see Figure 3.1c – we still have to estimate an offset between the timestamp of the camera and the timestamp of the MIMU data. The continuous position expression in Kalibr allows easily computing the Jacobian of the optimized cost function with respect to this offset and thus estimating it alongside the other quantities within the same gradient-based algorithm. We observed generally sub-millisecond reproducibility for this time offset estimation.

We have not used all the features of the Kalibr toolbox. It is, for instance, compatible with multiple cameras, and has been extended recently in [Rehder et al., 2016] to estimate IMU scale factor and sensitive axis position, with very impressive accuracy.

In this remaining of this chapter, we describe an instance of a simple estimator fusing data from the MI-DR EKF output and from a pose difference inferred from the depth image stream of the Asus Xtion Pro.

3.2 Depth Sensor Based Navigation

3.2.1 Related Work

One of the fundamental problem faced by passive computer vision is the 3D understanding of the environment. This is also true for the visual odometry and SLAM problem. The 3D structure can be inferred either by using multi-view geometry with a passive device or exploiting active devices such as time-of-flight (ToF) or structured light sensors.

"Depth" Camera has been popularized in the robotic community since the release of the technology of the company Primesense in the Microsoft Kinect gaming device. The most popular provided

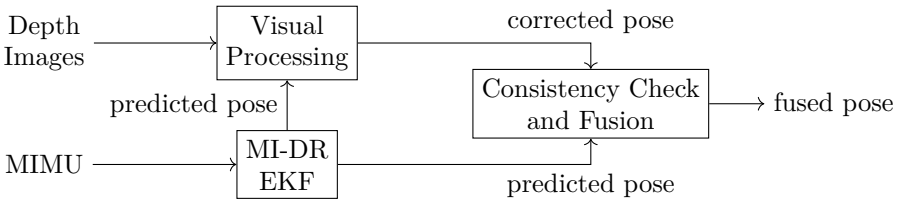


Figure 3.3: Architecture of the proposed pose estimation algorithm

a depth image registered with conventional color images to form a 4-channels RGBD image. This sensor had a lot of exciting features: it provided direct access to a high resolution (640x480) image where each pixel encoded a depth measurement and offered a direct 3D information from the environment, without having to deal with the difficulty of building a custom-made stereo ring and implementing an efficient stereo matching algorithm. The technology also has some strong advantages compared to classical stereo ring: because the triangulation relies on a projected infrared pattern, it is robust to the lack of texture in the scene, is a bit less sensitive to motion blur, and does not suffer from repetitive patterns in textures. Furthermore, its price was very competitive. However, it shares some caveats of the standard stereo-vision: it requires a significant baseline between the IR projector and the receiver to obtain a good triangulation accuracy, its depth image exhibits holes, corresponding to shadows created by foreground objects, and it fails in the presence of specular reflections (mirror effect). Besides, structure sensors are highly sensitive to sun IR illumination, which makes them mostly useless outdoor during the daytime.

In the years following Kinect release, numerous works have been dedicated to this new sensor for trajectory and 3D environment reconstruction. Some works use sparse visual features detected on an RGB image [Henry et al., 2014] to infer movement while others use a dense alignment that minimizes either geometric [Izadi et al., 2011] [Jaimez and González-Jiménez, 2015] and/or photometric criteria [Kerl et al., 2013b]. A few methods are hybrid spares/dense techniques though, combining advantages from both worlds: sparse features are used as an initial tracking guess that is refined by dense tracking [Henry et al., 2014].

Secondly, as visual odometry is prone to drift, many works construct a map of the environment in order to achieve higher accuracy [Henry et al., 2014; Kerl et al., 2013a; Izadi et al., 2011; Whelan et al., 2012, 2015], but this significantly increases computational workload and memory requirements.

Some works also combined structure sensor with an IMU for localization purpose, among the ones using a RGBD sensor [Guo and Roumeliotis, 2013] uses the association between the RGB and depth image to form 3D measurement of an image feature in an EKF filter. In [Qayyum et al., 2013] the authors extend the range of application of their EKF by introducing a measurement equation corresponding to a directional constraint computed from the essential matrix when the depth image is not available. [Brunetto et al., 2015] focuses on mobile device and use a loosely coupled approach based on two EKFs, one for translation, one for orientation, taking as input the pose of a SLAM algorithm and the inertial sensors. They explicitly note that they do not use magnetometer information because of field disturbances. Finally, the authors of [dos Santos Fernandes et al., 2013] use sparse features detected on the RGB image for coarse alignment against a keyframe and refine with a dense photometric error constrained on the measured depth image. The inertial information helps sparse features matching and loop closure detection. Their method leverages magnetic field as a bearing constraint and, as noted by the author, will not work in magnetically disturbed fields. For drift reduction, they build and solve a pose-graph optimization problem with SE(3) constraints resulting from photometric alignment solely, without inertial data.

3.2.2 Inspiration, Choices, and Expected Gains

A simplified outline of our estimator is given in Figure 3.3. The algorithm solves for the pose of the system and is built on a prediction/correction principle:

The Prediction Step

The prediction step generates a *predicted depth image* by warping a stored *depth keyframe* according to an input *predicted pose*. It is described in Section 3.3. The predicted pose is computed by integrating MI-DR EKF speed output between the previous pose estimate and the current camera pose.

Correction step

The correction step leverages a depth image alignment algorithm to refine the predicted translation between the current frame and the last keyframe. It is described in Section 3.4.

These predicted and corrected poses are actually maintained relative to the stored reference keyframe. Thus, the last step composes this relative pose with the current keyframe pose to return the final pose in the world frame (the world frame is thus anchored at the first keyframe position). We use such a keyframe scheme to annihilate slow drift in case of no or minimal motion. We select a new keyframe only when enough movement is detected – and we forget the previous keyframe. We introduced at different steps of the algorithm s aimed at rejecting the contribution of one or the other block of sensors.

Let us underline some choices that were made for our system:

- *No map of the environment is built.* In contrast with [Izadi et al., 2011] [Henry et al., 2014] and [dos Santos Fernandes et al., 2013], we focused on short term motion consistency, which is the expected gain provided by the MI-DR technique. If, without doubt, mapping will be necessary for applications, it requires a process an order of magnitude more computationally demanding compared to the tracking. Besides, in our opinion, (i) focusing on an as robust as possible tracking component would simplify the mapping component at the end and (ii) the mapping technique would highly depend on the quality of the tracking component. Mapping is undoubtedly a natural extension of this work though.
- *The correction step does not refine the rotational part of the movement.* First, we noticed that adding these extra degrees of freedom sometimes leads to instability in the image alignment process, mainly when the environment does not contain much information. Avoiding injection of short-term visual information into the attitude estimate also makes it independent of the visual environment and depth image noise/flaws. Furthermore, this refinement is actually not needed as the orientation estimation from the MI-DR is much cleaner than the output of the alignment process. One could argue that a slightly bad rotation prediction (because of bias, mis-calibration, or synchronization error) could bias the estimate of translation in the alignment process. In practice, such a bias has not been observed and is probably hidden behind other error sources from the depth sensor.
- *We do not use the RGB image in the alignment process,* contrarily to [Kerl et al., 2013b] for instance. Indeed, we have found that this choice makes our system more robust to changing illumination conditions, motion blur, specular material, and rolling shutter artifacts appearing on the Asus Xtion RGB sensor. It admittedly will be less robust in non-geometrically structured scenes. We could have circumvented the bad quality of the Asus Xtion RGB camera by using yet another conventional camera, but it would have complexified the offline extrinsic calibration process.
- Finally, our algorithm for depth alignment is close to the range flow based odometry presented in [Jaimez and González-Jiménez, 2015], except that: we use keyframing, we do not use their weight expression but a robust estimator instead, and the rotations parameters are not optimized on during alignment.

We expect to demonstrate a gain of robustness in various challenging situations for our hybrid odometry system:

- Compared to the MI-DR alone, we expect the depth alignment algorithm to improve dead-reckoning performance in the case of a non-stationary perturbation or when the magnetic gradient is not strong enough to correct the integration of inertial sensor.
- Compared to visual odometry methods alone, we expect to increase the frequency bandwidth of the system and to complete the navigation when the environment renders the movement unobservable to vision.
- By sensor data cross-validation, we hope to be able to detect big inconsistencies between blocks of sensors and to choose the most reliable estimate of the transform between current pose and last keyframe pose. This cross-validation should allow filtering out errors that are harder to detect *a priori* with solely the data from one or the other block of sensors.

The next sections will describe the details of the algorithm and experimental results obtained on a real dataset.

3.3 Depth Image Warping and Prediction

3.3.1 Depth Image Warping

First, we describe the synthesis of a warped depth image from a source depth image $\mathbf{D}_{\text{src}}$ and an input rigid transform $\boldsymbol{\xi}$. Let $\boldsymbol{\xi}$ be the transform between the camera frame at source image timestamp and camera frame at destination image time, a pixel $\boldsymbol{o}$ from the source image with depth value $\mathbf{D}_{\text{src}}(\boldsymbol{o})$ should be transformed in the destination image at a location $\boldsymbol{o}'$ with depth value d', according to the following warping function $\mathbf{w}_{\boldsymbol{\xi}}$:

$$\mathbf{w}_{\boldsymbol{\xi}} : (\boldsymbol{o}, d) \rightarrow (\boldsymbol{o}', d') \in \mathbb{R}^2 \times \mathbb{R}^+ \tag{3.2}$$

where $\boldsymbol{o}'$ and d' are defined with the following operations:

$$\text{(unproject 3D point from source)} \qquad \mathbf{X} = \pi^{-1}(\boldsymbol{o}, \mathbf{D}_{\text{src}}(\boldsymbol{o})) \tag{3.3}$$

$$\text{(warp pixel)} \qquad \boldsymbol{o}' = \pi\left(\boldsymbol{\xi}\mathbf{X}\right) \tag{3.4}$$

$$\text{(transform depth)} \qquad d' = \left[\boldsymbol{\xi}\mathbf{X}\right]_3 . \tag{3.5}$$

Forward warping A difficulty of that kind of image warping is that $\boldsymbol{o}'$ will not have, in general, integer coordinates in the destination image. Thus, it is necessary, after computing the warp function for every pixel, to reconstruct the virtual image by an interpolation, on the regular pixel grid of the destination image, of the warped values, which are lying on a non-regular grid. This process could reveal cumbersome.

Reverse warping A classical way to avoid the problematic interpolation is to use the reverse-warping function. If the warp-function is invertible and its inverse easy to compute, we can retrieve the value of a pixel in destination image by applying the inverse function on it, find the pixel floating coordinate in the source image, and interpolate over the regular grid of values in the source image. This process is more straightforward than the interpolation on an irregular grid.

Forward warping with splitting on the destination grid However, this is impossible to do for the prediction of a new depth image from the stored depth keyframe image, because the warping function (and thus its inverse) requires the depth value at the pixel *source* coordinates. We thus adopt the following strategy: we use a forward image warping to propagate each depth value to a new 2D position in the destination image, and we split the contribution of this warped pixel over the four closest pixels in the destination image. Each adjacent pixel is associated with a weight that depends on its center distance to the warped coordinate. We accumulate into two temporaries arrays for each pixel the sum of these contributions and weights. The final image is then computed as the division of the temporary value array by the weight array.

Care must be taken during this process, because occlusion could create a smoothed depth at the border of objects, by averaging far and close points. We handle these cases the following way: if the difference between the accumulated value (divided by weight) in the temporary array and a new value is significant – we choose three times the depth noise level – then we only retain the smaller depth value for the destination pixel. The interpretation of this situation is that the farthest point is actually occluded by the closest one in the new point of view and thus should be discarded.

Note that for the residual computation presented in next section, this process is not necessary. This is because the residual we use can be computed by interpolation of the current image at the keyframe-warped coordinates $\boldsymbol{o}'$, which is a simple bilinear interpolation on a regular grid.

3.3.2 Depth Image Prediction from Last Keyframe

Here we describe the construction of the predicted image built from both the MI-DR EKF output and the depth image of the last keyframe.

We assume that the previous frame $\mathbf{D}_k$ (with timestamp t_k) already has a pose estimate relative to the last keyframe, we call it $\boldsymbol{\xi}^{\text{kf}\leftarrow k}$. When a new depth image $\mathbf{D}_{k+1}$ is received (with timestamp t_{k+1}) we exploit the MI-DR EKF output since time t_k to compute the displacement $\boldsymbol{\xi}_{\text{mi}}^{k\leftarrow k+1} = \left(\mathtt{R}_{\text{mi}}^{k\leftarrow k+1}, \mathbf{T}_{\text{mi}}^{k\leftarrow k+1}\right)$ as follows:

$$\mathtt{R}_{\text{mi}}^{k\leftarrow k+1} = \mathtt{R}_{\text{mi}}(t_k)\mathtt{R}_{\text{mi}}(t_{k+1})^{\mathsf{T}} \tag{3.6}$$

$$\mathbf{T}_{\text{mi}}^{k\leftarrow k+1} = \mathtt{R}_{\text{mi}}(t_k) \sum_{p\in\boldsymbol{\mathcal{V}}} \mathtt{R}_{\text{mi}}(t_p)^{\mathsf{T}} \mathbf{v}_{p;\text{mi}}^{b} \left(t_p - t_{p-1}\right) \tag{3.7}$$

In the latter equation, $\{t_p\}_{p\in\boldsymbol{\mathcal{V}}}$ denotes the timestamp of the data emitted by the MIMU sensors and integrated in the EKF between t_k and t_{k+1}. Since the sensors are not synchronized, but only timestamped in the same time-frame, we use linear interpolation for velocity and a SLERP interpolation for the rotation matrix when necessary. This is to correct for slight differences of timestamps between MIMU sample and images. The predicted transform $\boldsymbol{\xi}_{\text{pred}}^{\text{kf}\leftarrow k+1} = \boldsymbol{\xi}^{\text{kf}\leftarrow k}\boldsymbol{\xi}_{\text{mi}}^{k\leftarrow k+1}$ is used for warping the last keyframe image depth $\mathbf{D}_{\text{kf}}$ to synthesize a predicted image $\mathbf{D}_{\text{pred}}$ with algorithm of Section 3.3.1. This predicted image is used as a starting point to the depth alignment algorithm presented in Section 3.4 (with splitting of transformed pixel contributions on neighbor pixels)

As an alternative to MI-DR based prediction, visual-only navigation literature often leverages a simple damped motion model where the translational speed evolution verifies an exponential decay:

$$\dot{\mathbf{v}^w} = -\alpha\mathbf{v}^w \tag{3.8}$$

with $\alpha \in \mathbb{R}^+$ being a fixed constant, parameter of the algorithm, that drives how fast the speed should go down to zero in the absence of visual information measurement. This translates in discrete form, assuming a constant frame rate $\frac{1}{\Delta t}$, to:

$$\mathbf{T}^{k\leftarrow k+1} = -e^{-\alpha\Delta t}\mathbf{T}^{k\leftarrow k-1} \tag{3.9}$$

which can be computed from the previous corrected version of $\mathbf{T}^{k\leftarrow k-1}$. We use this model as a default prediction when the output of MI-DR cannot be trusted.

3.4 Robust Depth Image Alignment for Motion Estimation

In this section, we describe the depth image alignment algorithm used for the correction of the predicted pose. Its role is to compute a correction vector $\delta\mathbf{T}^*$ that best aligns the current image with the predicted depth image. Correcting the translation solely is sufficient because the predicted depth is already rotation corrected by the prediction step described in Section 3.3.2.

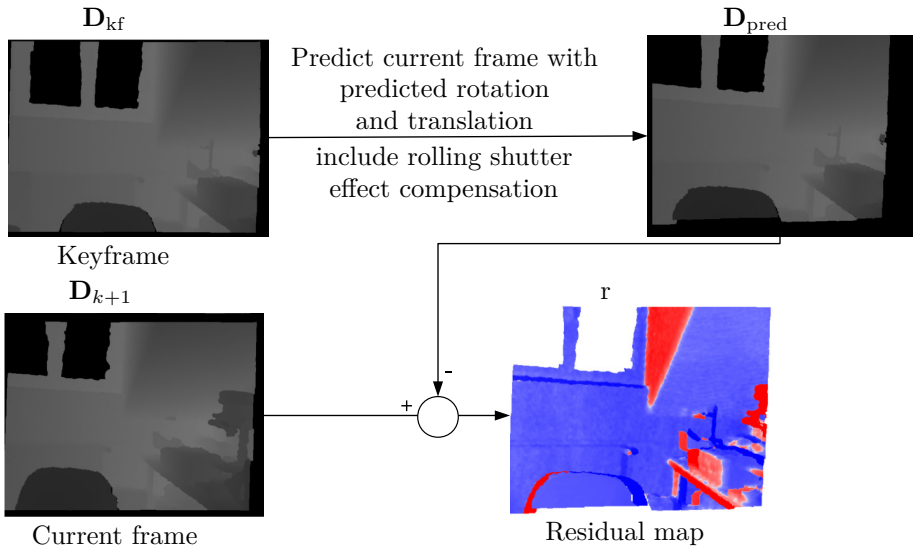


Figure 3.4: Residual computation. Rolling shutter correction is explained in Section 3.5.1

3.4.1 Alignment Error Function

More precisely, from the predicted depth image $\mathbf{D}_{\text{pred}}$, we aim to find the optimal translation – as a kind of rigid body transform – that parametrizes the warping function $\mathbf{w}_{\boldsymbol{\xi}}$. We search it by optimizing a depth alignment criterion.

This criterion relates the difference between the warped version of the (already warped) $\mathbf{D}_{\text{pred}}$ and the current depth image $\mathbf{D}_{k+1}$.

Mathematically, the criterion writes:

$$\boldsymbol{\delta}\mathbf{T}^* = \arg\min_{\boldsymbol{\delta}\mathbf{T}} \left(\sum_{u \in \mathcal{C}} \rho \left(\frac{r_u(\boldsymbol{\delta}\mathbf{T})}{\sigma_u(\boldsymbol{\delta}\mathbf{T})} \right) \right) \tag{3.10}$$

with:

$$r_u(\boldsymbol{\delta}\mathbf{T}) = \begin{cases} \mathbf{D}_{k+1}(\boldsymbol{o}'_u) - d'_u & \text{if } \mathbf{D}_{k+1}(\boldsymbol{o}'_u), \mathbf{D}_{\text{pred}}(\boldsymbol{o}_u) \text{ are valid} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

and:

$$\text{(a robust norm)} \qquad \rho : \mathbb{R} \to \mathbb{R} \tag{3.12}$$

$$\text{(unprojected 3D point)} \qquad \mathbf{X}_u = \pi^{-1}(\boldsymbol{o}_u, \mathbf{D}_{\text{pred}}(\boldsymbol{o}_u)) \tag{3.13}$$

$$\text{(warped pixel)} \qquad \boldsymbol{o}'_u = \pi\left(\mathbf{X}_u + \boldsymbol{\delta}\mathbf{T}\right) \tag{3.14}$$

$$\text{(transformed depth)} \qquad d'_u = [\mathbf{X}_u + \boldsymbol{\delta}\mathbf{T}]_3 \tag{3.15}$$

Note that, this involves the same operations as the warping function $\mathbf{w}_{\boldsymbol{\xi}}$ defined in previous section, but we rewrote them in the special case where the $\boldsymbol{\xi}$ rotation is Identity.

$\mathcal{C}$ denotes here the entire pixel array: we sum all residuals between pair of pixels in warped and current image that are both valid; if there are invalid pixels (because of shadows, out-of-range measurement or out-of-bound warping) in either the warped or current image, they participate in the cost as (constant) zero residuals. Note that the facts that a pixel participates in the cost depends on $\boldsymbol{\delta}\mathbf{T}$ and thus will change across iterations. The Figure 3.4 shows a residual image before alignment.

3.4.2 Weighting

The precision at which this criterion should be true differs for each pixel. The cost function accounts for this through the weighting functions σ_u and ρ.

In one hand, closer pixels have less noise in their depth measurement, which is encoded in the cost through the weight $\frac{1}{\sigma_u^2}$. σ_u is an estimate of uncertainty of the residual, and can be approximately propagated from uncertainty on $\mathbf{X}_u$ and on $\mathbf{D}_{k+1}$ with a first order propagation:

$$\sigma_u^2(\boldsymbol{\delta}\mathbf{T}) = \begin{cases} \frac{\partial r_u}{\partial \mathbf{X}_u}\Big|_{\boldsymbol{\delta}\mathbf{T}} \boldsymbol{\Sigma}_{\mathbf{X}_u} \frac{\partial r_u}{\partial \mathbf{X}_u}\Big|_{\boldsymbol{\delta}\mathbf{T}}^{\mathsf{T}} + \sigma_z^2 & \text{if } \mathbf{D}_{k+1}(\boldsymbol{o}'_u) \text{ is valid} \\ 1 & \text{otherwise} \end{cases} \tag{3.16}$$

In the above formula, σ_z is the uncertainty on the depth image value $\mathbf{D}_{k+1}$ at pixel location $\boldsymbol{o}'$ – we use for this quantity, the axial noise developed in [Nguyen et al., 2012] – and $\boldsymbol{\Sigma}_{\mathbf{X}_u}$ is the covariance of the position of point $\mathbf{X}_u$; it depends originally on the uncertainty of the depth estimated in the *original* keyframe (and not the predicted depth image computed from original keyframe). Propagating the uncertainty rigorously would be cumbersome. Rigorously, the uncertainty of the depth should propagate through the prediction warping. However, we assume here the main source of uncertainty comes from the noise in the depth measurement in the keyframe and that we have a small rotation and translation between the two frames to be aligned. Within these conditions, we can use the following simple form:

$$\boldsymbol{\Sigma}_{\mathbf{X}_u} \simeq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} \tag{3.17}$$

Where we use the same σ_z as previously defined, and the first terms of (3.16) is simplified as:

$$\sigma_u^2 = \left(\frac{\partial r_u}{\partial [\mathbf{X_u}]_\mathbf{3}}\Big|_{\boldsymbol{\delta}\mathbf{T}} \right)^2 \sigma_z^2 + \sigma_z^2$$

This simplification can seem pretty rough. But we decided not to over-complicate the algorithm, considering even the main assumption of an independent zero-mean noise per pixel on depth image was observed to be very wrong with the real sensor: the Asus Xtion Pro we used, exhibited strong point-cloud distortion and quantization noise.[1]

In the other hand, this iterative scheme also encodes a robust t-Student weighting function through ρ, in the line of [Kerl et al., 2013b], which leads to an improved alignment robustness to high residuals arising from edges, occlusions, nonrigid scenes or depth sensors high noise. One issue is that the cost function (3.10) is not occlusion aware: if the corrected translation creates new occlusions, then two pixels of different predicted depth could project to the same pixel in the newly warped image, and we can expect at least one of the two residuals to be very high. Nevertheless, provided that the translation initialization is good enough the predicted image should be sufficiently close to the current image and this effect would appear on a minimal number of pixels. This will easily be mitigated by the robust estimator, which would thus down-weight the residual coming from the farthest point of the two warped coordinates.[2] Note that the use of a robust norm ρ, is equivalent to weighting each residual by a weight function $\mathbf{w}_\rho$ that changes at each iteration.

[1] For distortion, the authors of [Teichman et al., 2013] propose to correct these distortions by applying correction stored in a look-up table. We thought it was overkill for our purposes, particularly when considering the sensitivity of this correction to mechanical constraint applied to the camera showed by the same author on one's author blog (http://alexteichman.com) as a follow-up remark. For quantization noise, the authors of [Bonnabel et al., 2014b] prefer to refine the estimated covariance of alignment using a non-independent Gaussian error model. We have not thoroughly investigated these issues in the present work.

[2] One solution to prevents high residual in the case of new occlusion could be to recompute the full warp again from non-warped keyframe depth image at each new iteration using the predicted orientation and the corrected value of the translation, as this prediction warping process handles some part of self-occlusions effects, in contrast with the direct cost function evaluation.

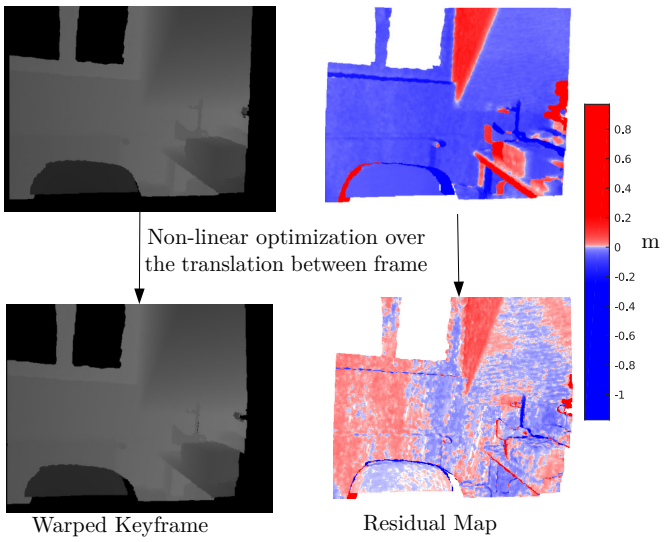


Figure 3.5: Typical effect of the depth alignment on the residuals. The residuals are drastically improved. High residual remains around the edges, but are down-weighted by the robust estimator. Also, we can perceive clear circular distortion in the residual map after optimization.

3.4.3 Optimization

This criterion is minimized with an IRLS (Iteratively Reweighted Least-Square) strategy and a Gauss-Newton like[3] algorithm so that at the iteration i we solve the following linear problem:

$$\min_{\boldsymbol{\delta T}_i} \sum_u \frac{w_\rho^2(\frac{r_u}{\sigma_u})}{\sigma_u^2(\boldsymbol{\delta T}_{\text{i-1}})} \left(\left. \frac{\partial r_u}{\partial \boldsymbol{\delta T}} \right|_{\boldsymbol{\delta T}_{\text{i-1}}} \boldsymbol{\delta T}_{\text{i}} - r_u \right)^2 , \tag{3.18}$$

where $w_\rho : \mathbb{R} \to \mathbb{R}$ is the weighting function associated with the chosen robust norm. This can also be written with the following matrix form:

$$\min_{\boldsymbol{\delta T}_i} \left\| \mathbf{W_i} \left(\mathbf{J_i} . \boldsymbol{\delta T}_{\text{i}} - \mathbf{r} \right) \right\|_2^2 . \tag{3.19}$$

We describe how this linear least squares is solved in the next section.

In order to increase convergence basin, we use a pyramidal scheme with a number of resolution levels that depends on whether the inertial prediction was judged reliable or not, according to a criterion described in Section 3.5.2. For real-time reasons, we set the number of iterations to a fixed number per level. The effect of the optimization on the residual image is shown on Figure 3.5; as can be seen, the residual image magnitude drops drastically, but exhibits a faint circular pattern. This pattern is a systematic error which comes from a kind of mis-calibration in the depth sensor, that is not corrected for.

Once the optimal translation $\delta\mathbf{T}^*$ is retrieved, and, if optimization result is trusted (see conditions in Section 3.5.2), this vector is added to the predicted translation, the sum of the two being the corrected translation estimate between current frame and keyframe.

3.4.4 Dealing with Underconstrained Estimation

It might happen, in some scenes, that the two input depth images do not entirely constrain the 3D translation. This occurs, for instance, when viewing solely parallel planes or, more generally, when the normal vectors of all surfaces perceived do no form a proper basis of $\mathbb{R}^3$.

[3] GN-"like" because the number of residual, thus the cost definition - changes at each iteration

In this case, the correction translation becomes unobservable along one or two directions. This is dangerous and can lead to uncontrolled divergence of the optimization along these directions. We thus try to detect this case explicitly in the algorithm. This carefulness is even more critical in our fusion problem. We would like to retain the initialization values for the translation along the unobservable directions – as this values stem from the meaningful prediction done by the MIMU – but still optimize along the observable directions. We solve this issue by explicitly constraining the increment at each iteration along the subspace of the observable directions. We employ the method described hereafter to do so.

Recall the linear least-square problem one need to solve at each iteration:

$$\min_{\boldsymbol{\delta}\mathbf{T}} \|\mathbf{W}\left(\mathbf{J_i}\boldsymbol{\delta}\mathbf{T_i} - \mathbf{r}\right)\|_2^2 . \tag{3.20}$$

Before constructing the normal equation, we start by identifying the rank of the matrix $\mathbf{WJ_i}$ with the help of a rank-revealing QR decomposition[4]. The rank tolerance is set, empirically, higher than the machine epsilon in order to account for noise in the data. Depending on the rank value, two cases can be distinguished. If its rank is 3, we simply use this decomposition to solve the least squares problem. Otherwise, they are numerically non-observable directions. In this case, we explicitly form the approximated Hessian matrix $\boldsymbol{\mathcal{I}} = \mathbf{J_i^T W^T W J_i}$, and compute an approximate inverse $\boldsymbol{\mathcal{I}}^\dagger$ using a truncated eigenvalues decomposition:

$$\boldsymbol{\mathcal{I}}^\dagger = \sum_{k=1}^{\text{rank}} \frac{\mathbf{1}}{\lambda_\mathbf{k}} \mathbf{u_k u_k^T} \tag{3.21}$$

Where λ_k is the k-th largest eigenvalue and $\mathbf{u_k}$ the associated eigenvector. The rank is determined as the number of eigenvalue above an empirically chosen threshold. A truncated increment is then:

$$\boldsymbol{\delta}\mathbf{T_i} = \boldsymbol{\mathcal{I}}^\dagger \mathbf{J_i^T W^T r} \tag{3.22}$$

This strategy constrains the optimization process over the locally observable subspace and prevents polluting the initial solution with noise due to bad matrix conditioning.

3.4.5 Limitations of Depth Alignment for Trajectory Reconstruction

One main limitation of this iterative depth alignment algorithm is its convergence basin. A lousy initialization would surely lead to a wrong translation estimate. Actually, the use of this algorithm without any prediction leads to very wrong trajectories on our dataset. If the pyramidal strategy for depth-only based navigation improves, indeed, the convergence basin, it shows limits, for instance when the trajectory involves very high dynamic motion.

Furthermore, a non-static scene will drastically degrade the accuracy of the alignment. For instance, assume that a large rigid body moves in front of the depth sensor; in that case, the cost function is likely to have two strong local minima: one aligning the static scene, and one aligning the moving rigid body (see an example in Figure 3.7). Without proper initialization, the wrong one could be chosen. Of course, the robust loss could cope with local motions of the scene, but it reveals inefficient when a big part of the image is not static.

Moreover, the number of iterations before convergence is hard to define and, being a dense algorithm, the alignment algorithm is computationally demanding, especially on lowest levels of the pyramid, which is a problem if targeting embedded devices.

Another limitation comes from the sensor itself: its operational range. Obstacles after 3 meters are noisy, and not even detected after 5 meters. This is a problem in larger rooms where, often, the only good obstacle detected is the floor, which provides only one observable direction for the depth alignment algorithm.

[4] We use ColPivHouseholderQR function from C++ Eigen library.

For all of these limitations, the initialization from MI-DR estimate helps: if the initialization is good enough, it reduces the number of iterations and makes the optimization choose a minimum closer to the one aligning the static part of the scene. The *switch* strategy presented in the next section is an attempt to exploit this and to handle failures cases of sensors.

3.5 Preprocessing and a Switch-based Fusion Strategy

This section details the full estimator pipeline. And in particular its *switch* strategy. Its main idea is to leverage prior knowledge of failure modes and failure detection logic in order to select – *switch between* – the *best* estimate of the translation components. On the diagram of Figure 3.6, this selection is represented by the switches 1 and 2; the Section 3.5.2 explains their behaviors.

3.5.1 Depth Image Preprocessing and Rolling-shutter Compensation

As often done when dealing with this kind of low-cost depth sensor – for instance in [Izadi et al., 2011] – we apply an adaptive bilateral filter on all received depth images. In particular, this makes the gradient of the depth image smoother, which is beneficial for the differentiation of the residual of the cost function.

Furthermore, low-cost sensors are generally equipped with a rolling shutter mechanism (RS): every row of the image is actually exposed at slightly different time. This phenomenon is hard to model in the cost function (3.10) (see [Kerl et al., 2015]). We propose to account for the RS by a dedicated preprocessing step. More precisely, we correct solely for the RS distortions that are created by the rotational part of the movement $\mathtt{R}^{k\leftarrow k+1}$. We proceed by assuming a constant rotational velocity over the entire image capture time. $\boldsymbol{\omega}_{1\leftarrow 2} = \frac{\log \mathtt{R}^{k\leftarrow k+1}}{t_{k+1}-t_k}$, we use the depth image warping process described in Section 3.3.1 using for each line the rotation:

$$\mathtt{R}_{\text{line}} = \exp_{\mathfrak{so}(3)}\left(\boldsymbol{\omega}_{k\leftarrow k+1}(t_{\text{line}} - t_{\text{center line}})\right) \tag{3.23}$$

as the rigid transform $\boldsymbol{\xi}$. t_{line} is the timestamp of the current line, and $t_{\text{center line}}$ is the instant of exposure of the centerline. The difference between the two is proportional to the distance of the current line to the centerline, the coefficient being roughly calibrated offline by filming a TV screen with a known refreshing rate.

Admittedly, rolling-shutter artifacts created by translation would still impact the image alignment algorithm negatively. However, they are generally of lower magnitude compared to the one created by rotational movement, at least for the kind of movement we are interested in.

Note that, the previous formula implicitly says that all inter-frame transforms estimated by the alignment algorithm are between instants of the center line exposure.

3.5.2 Correction and Consistency Checks

The switches 1 and 2 are driven by consistency checks between prediction steps and depth alignment. Consistency checks are embedded in the pyramidal optimization process of (3.10). The predicted depth map is first synthesized at a coarse spatial resolution corresponding to the highest level of the multi resolution pyramid. With this image we compute a rough initial value of the cost (3.10) which is used to assess the consistency of visual and magneto-inertial information by a threshold test (corresponding to the switch 1 in Figure 3.6).

If the initial cost is smaller than some threshold th_c, the two information are flagged as consistent and the pyramidal optimization proceeds (Switch 1 connects the EKF block). Otherwise, an inconsistency is detected. We then discard the predicted translation and use the default motion model (3.9) instead (Switch 1 points towards block 'Default prediction MM", MM standing for "motion model").

In both cases, the pyramidal optimization described in Section 3.3 is run. We then examine the final (minimal) cost with another threshold test (Switch 2 in Figure 3.6). If this cost is too high, it

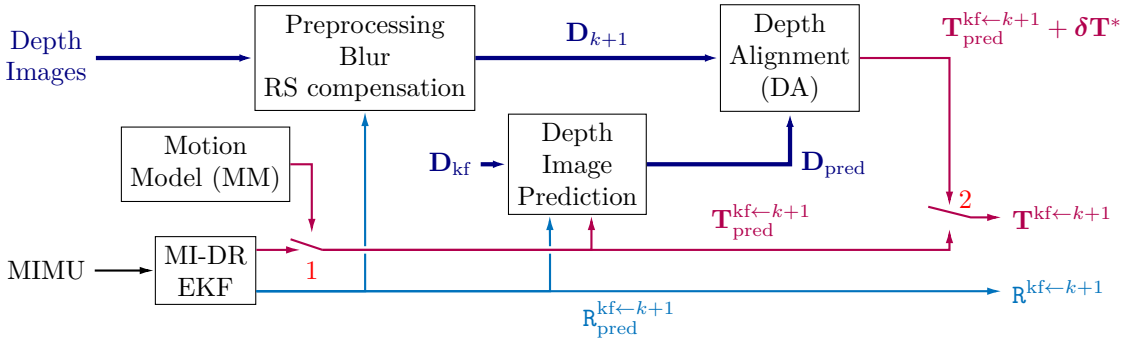


Figure 3.6: Detailed data flow of the estimator

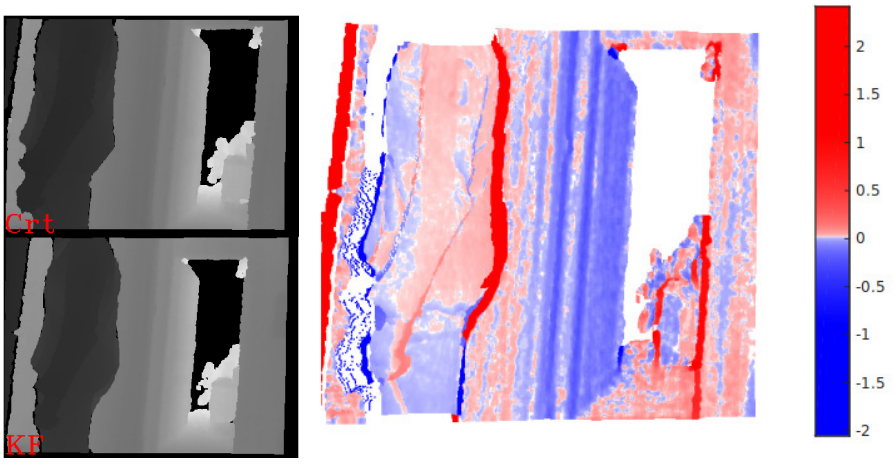


Figure 3.7: Depth alignment on non static environment. Here a moving pedestrian was registered instead of the static environment. (see the high residual on the door in the background, while the body of the pedestrian shows smaller residual, except for occlusion effects)

means that there was a problem with the visual data (for instance because of a non-rigid scene): we discard them completely and the MI-DR prediction is rehabilitated and promoted as the estimator output (Switch 1 points towards EKF and Switch 2 points to the predicted translation). Otherwise, the visual information is considered relevant, and the corrected translation can be used (Switch 2 points towards the corrected translation).

These heuristics allow our estimator to be more robust to the violation of the assumptions related to one block of sensors or the other. This will be demonstrated on real data in the experimental part. Note, as already mentioned, that the rotational part of the MI-DR filter is always kept as the final orientation estimate.

3.6 Result in Indoor Environment

3.6.1 Implementation

The complete algorithm has been implemented in C and C++ language within the ROS (Robotic Operating System) framework. Due to real-time constraints, we limit the computation of the depth alignment algorithm to 2 iterations per pyramid level and do not iterate at the lowest level – whose size is 640x480 px. The entire pipeline processes a new frame faster than the 30 ms between subsequent images on an Intel NUC with an Intel Core i5-6260U CPU.

We present results in examples demonstrating sensor complementarity in various challenging situations, and we show quantitative comparisons results in a motion capture equipped room.

For analysis, we compare the results of our framework under different configurations:

- MI-DR: the MI-DR filter of Chapter 2.
- MI-DR+DA: presented system with described depth alignment as visual correction step.
- MM+DA: system (1) using only the default motion model – switch 1 forced to MM.
- MI-DR+DVO: presented system with DVO [Kerl et al., 2013b] as the visual correction step – thus also using *image intensity* alignment instead of depth alignment.
- MM+DVO: system (3) using only the default motion model – switch 1 forced to MM.

Note that DVO from [Kerl et al., 2013b] does not incorporate unobservable directions handling and uses an additional RGB error term in the cost function. Moreover, for fairness, we let DVO optimize only over the translational components – original author's implementation optimizes on full rigid body transform.

We also show results of the pure integration of the output velocity of the MIMU filter MI-DR and ground truth (GT) when available. When comparing with ground truth data, the estimated positions were aligned with the ground truth transform *on the first pose* only.

3.6.2 Typical result

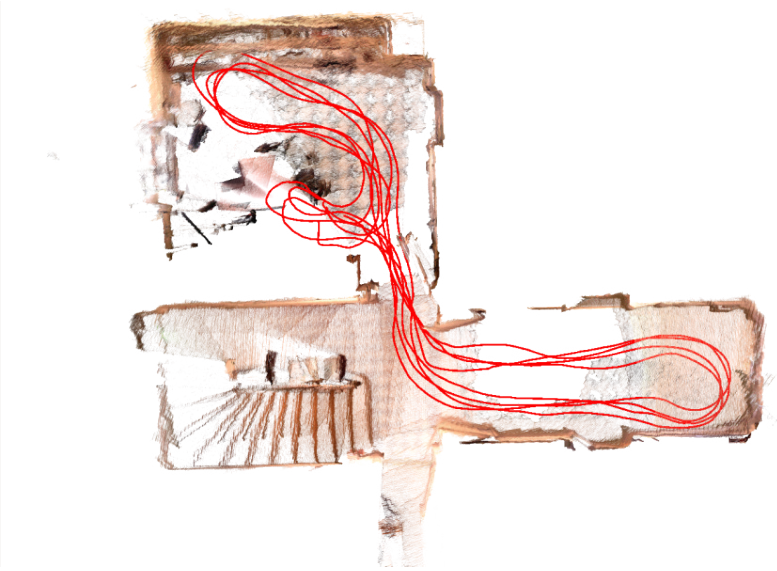
The Figure 3.8 shows a typical result from our estimator. This trajectory does not show any major difficulties, except for its relatively high dynamic compared to traditional RGBD dataset – translational speed is nearly always between 1 or 2 meters by second. The pedestrian is walking several times the same loop indoor, carrying the systems in a traditional Normandy house. The space in which the pedestrian evolves is rather narrow, such that the limited range of the device is not a problem, and the gradient of the magnetic field is adequate for MI-DR technique most of the time. The Figure 3.8a shows the trajectory along with a map of the environment reprojected *from the first loop estimate only* for visualization. Note how better the loops superimpose in Figure 3.8c compared to Figure 3.8b. The final drift is reduced by 1.1 to 0.2 % of the trajectory length compared to the MI-DR estimate.

3.6.3 Robustness Gain Compared to the MI-DR filter

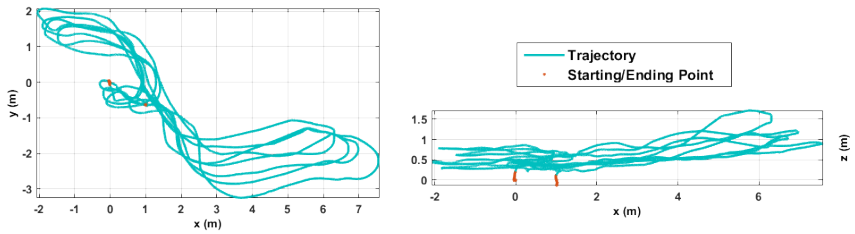
Non-stationary magnetic perturbation (Figure 3.9) In this experiment, we take the system by hand from a table, move it along some circle in the horizontal plane 20 times. During the first ten circles, no magnetic perturbation is generated, except the natural stationary one of the indoor location. In the next ten circles, a magnet is chaotically moved close to the MIMU sensor. Reconstructed trajectories are depicted in Figure 3.9.

The circle movements of the hand-held system are well reconstructed by MI-DR during the first 30 second despite a slight drift, which is corrected by the image alignment process on MI-DR+DA version. After the magnet starts moving, the effect of the perturbation over the MI-DR estimate is clearly visible and induces a massive drift along the y-axis and minor perturbations along the x-axis. This drift is mainly corrected by the depth based step of our estimator. Here, the final drift is 10 centimeters for 14meters of trajectory. This drift has been computed by comparison with the trajectory estimated by the map-based methods of [Kahler et al., 2015], that is virtually drift-free in this situation where the visual system always looks at the same place.

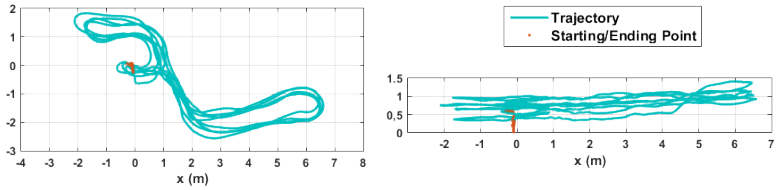
Environment with no magnetic gradient (Figure 3.10) In this experiment, we recorded a trajectory in the center of an empty motion capture room where the gradient is particularly weak. We display in Figure 3.10 the reconstructed speed along with the color-coded magnitude of the gradient. On blue regions, the MI-DR prediction is rejected directly, and the default motion model is used as a prediction. Figure 3.10 clearly shows the speed error introduced in low gradient areas on MI-DR estimate and the better speed reconstruction of our MI-DR+DA hybrid solution.



(a) Trajectory and point-cloud unprojected from estimated position during the entire first loop.

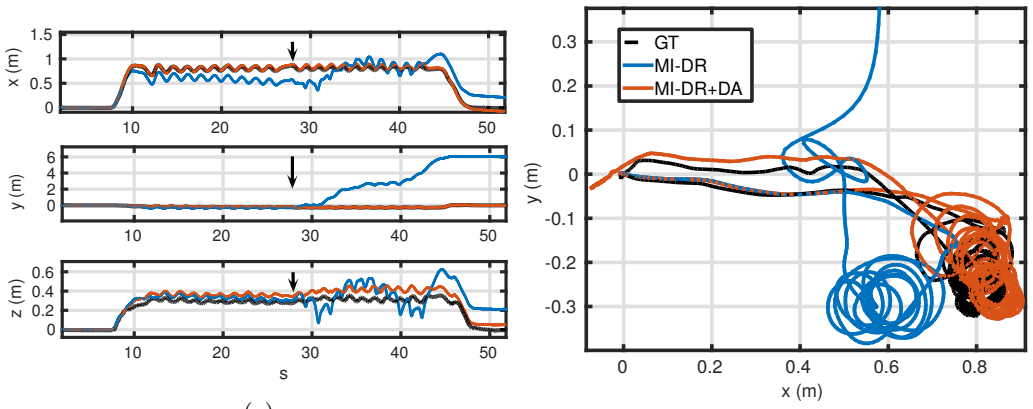


(b) MI-DR estimate



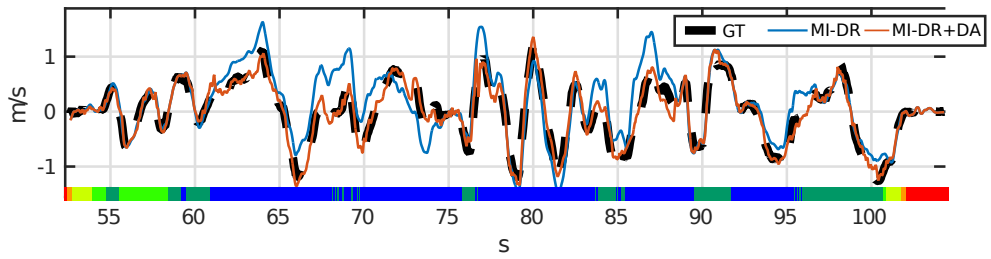
(c) MI-DR+DA estimate

Figure 3.8: Example of typical result in favorable environment. Loop error decrease from 1.1 % to 0.2 % of trajectory length with the proposed fusion scheme compared to the MI-DR estimate.

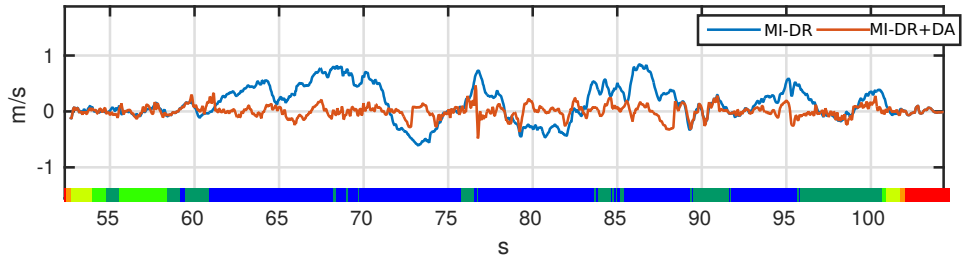


(a) (b)

Figure 3.9: Trajectory outputted in presence of non stationary magnetic perturbation, see Section 3.6.3. The perturbation starts at timestamp pointed by the black arrow. Figure 3.9a Position estimated from the MI-DR filter and from the proposed estimator. Figure 3.9b: Zoom over the trajectory path (top view). In this experiment the camera points constantly into the same direction, we build a pseudo-ground truth from a map building method [Kahler et al., 2015] depicted in black here.



(a) Speed estimate



(b) Error compared to ground truth

Figure 3.10: Speed estimate in an environment with weak magnetic gradient. Color coded bar of measured gradient is depicted with same code as in Figure 3.12b. In case of low gradient our fused estimate still allows tracking accurately the speed of the sensor.

3.6.4 Robustness Compared to Depth Map Alignment Based Navigation (Figure 3.11)

To prove this point, we tried our system in a staircase. Staircase environments are not such a problem for depth sensor pointing downwards because of the stair step that provides a well-constrained alignment cost. On the other hand, this narrow environment is a challenge if the camera points straight ahead, and thus for two reasons:

- First, because of *minimal* range limitation of the sensor: in this small environment the sensor can be blocked by a wall at less than a few dozen of centimeters, below its minimal range.
- Secondly, even if the range is acceptable, chances are that the sensor is seeing only one or two flat walls, making some directions of translation unobservable.

This experiment is depicted on Figure 3.11. In this experiment, we are starting from the first floor, going down to the ground floor, walking up to the second floor and finally going back to the initial position with a hand-held system. We depict the results of our system compared to the motion model based systems. With or without MI-DR prediction, we observe poor performance from MI-DR+DVO and MM+DVO here. This is mainly because of the bad quality of the photoconsistency assumption DVO is based on: inside the staircase the contrast is low compared to the vignetting errors of the sensor, as we can see in the RGB images timeline on Figure 3.11. Vignetting attracts the estimated transform towards an erroneous stationary solution, whatever the prediction input. The final position error of the proposed MI-DR+DA here is 0.12 % of the total length: 4.7cm over 40 meters of trajectory.

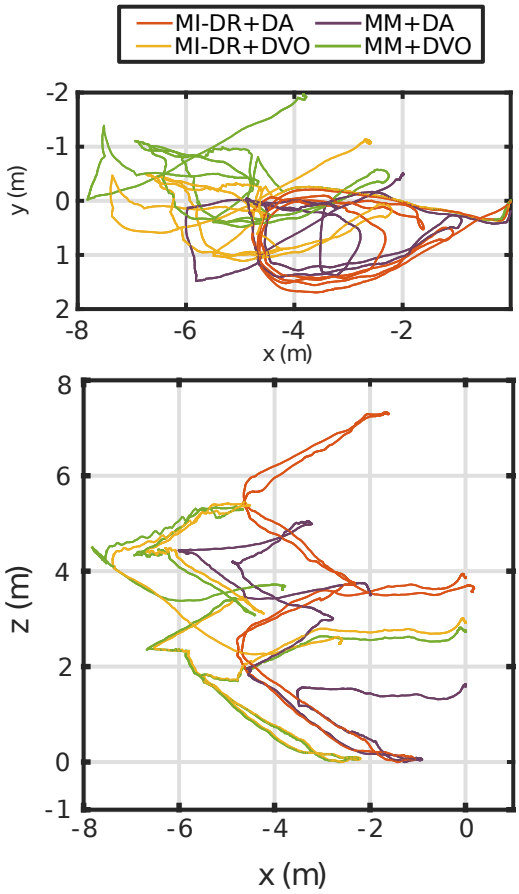
3.6.5 Experiment Within a Motion Capture Room

We recorded a trajectory with our system in a motion capture room. The room was populated with a desktop, closet and some electronic devices to create texture, depth variability, and magnetic gradient. Note however that the environment is unfavorable because the room is rather large. This translates into issues with the limited range of the vision sensor, and also implies areas with low magnetic gradient as shown in Figure 3.12b which depicts 3D groundtruth trajectory is plotted colored according to local magnetic gradient norm.

Figures 3.12a, 3.12c and 3.12c present position and speed compared to ground truth for the different configurations. Several comments can be made. First, Figure 3.12a shows that the MIMU prediction reduces the total drift: the MI-DR+DA and MI-DR+DVO trajectories are both closer to the ground truth trajectory than MM+DA and MM+DVO. Second, compared to the MI-DR EKF predictions all vision-based corrections significantly reduce the drift.

Third, if looking finely at the error over each axes in the inertial frame, we can notice that, while indeed greatly reducing the drift on the horizontal plane, the MIMU prediction introduces a small drift along the vertical axis. Indeed, along z axis, MM+DA and MM+DVO show less drift than MI-DR+DA and MI-DR+DVO. On this trajectory, still along z axis, MM+DVO and MM+DA perform sensibly the same with a slight advantage for MM+DA. We suspect this upward drift comes from small residual biases of accelerometers or other mis-calibrations of our system. However, the total drift with MI-DR+DA remains very limited: 37 cm for a 70m trajectory (0.53%).

Finally, speed estimation is evaluated in Figure 3.12c. Note that the speed estimates are obtained by position differentiation, except from MI-DR solutions for which they are read directly in the output of the filter. Here motion model based estimation sometimes loses track in case of poor visual information (for instance between 70 and 75s or just before 90s). This is probably due to the limited range of the sensor compared to the size of the room. Note also the bias appearing near areas with weak magnetic gradient (blue zone in the ground truth trajectory in Figure 3.12b).



(a)

(b)

Figure 3.11: Results of trajectory reconstruction in a situation of lost observability from vision block sensor. Depth and image based movement reconstruction fail because the data is not informative enough. The trajectory spans over three floors. Details in Sec. 3.6.4. 3.11a: All estimated trajectories aligned on ground floor level. 3.11b: Images that were captured by camera every second between first and second floor,showing poor visual environment.

3.7 Conclusion of this Chapter

The chapter first gave some details on the framework of the thesis. We described the hardware we used, some general prerequisites for using a camera combined with the MIMU sensors, and practical solutions to the platform calibration and synchronization we used. Finally, we presented a way to fuse information of a depth camera with those of a MIMU in order to gain some robustness in challenging environments. We showed a series of first results on challenging trajectories, each one focusing on a failure mode of one or the other sensor block. The employed strategy shows good results in the limited cases depicted here but also has some weaknesses.

Simple odometry This estimator is purposely kept simple (no map, no loop closure) in order to be computationally lightweight. It could be extended further, optimizing in inverse depth domain (as for instance in [Gutierrez-Gomez et al., 2016]), refining the occlusion handling in the cost function, or as already said, by doing image to model tracking, building a map and handling loop-closures. As mapping comes with a significant increase in memory requirement and computational load, it would depend on the applications specific requirements and constraints. However, we argue that even using a mapping strategy, robust odometry is critical.

The sensor modality chosen cannot handle some situations If the sensors complement themselves well in presented scenarios, they both fail at least on the following scenario: consider an outdoor or semi-outdoor case. As already said, the depth alignment algorithm would only be able to detect the floor and correct the estimate vertically, and thus would rely extensively on the input prediction for horizontal. However, chances are in this kind of environment that the magnetic gradient will also be weak, that would also deteriorate MIMU-baked input prediction. Consequently, the fusion strategy would be useless in this case.

More generally, even if the environment is structured enough for depth alignment, they are a lot of practical cases where the convergence of alignment would fail with a default motion model. As we can not exclude the magnetic field to be unusable for an extended period, the "fallback" visual-inertial systems should work as accurately as possible and should thus be a premium focus of the work. This imply, for instance, information feedback from vision to the inertial biases estimates.

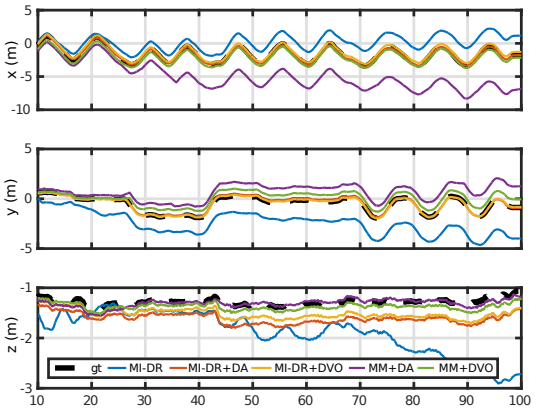
We will advocate in next chapters that tight fusion of magneto-inertial and visual information is the right path for further robustness improvement.

Hardware engineering was difficult The hardware used in the described work is an issue by itself. Depth sensors are not standard technologies: even if some competitors have made other depth sensor products since Primesense (for instance, Intel, PMDtech, structure.io, etc.). These sensors are often packaged into a consumer product which makes them difficult to sync with an external clock. Also, differences in technology and performance of depth sensors, lead to very different frame rate, depth range, and resolution. It could render the algorithm performance very sensor dependent, a situation far from ideal. General cameras are much more standard, simple and more straightforward to integrate.

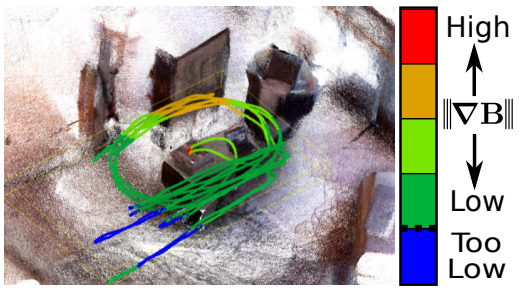
In a sense, it also did not feel right to combine such a finely tuned and mastered MIMU sensor with such a bad quality visual hardware. If developing algorithms for a stock of already distributed sensors can be an appealing challenge – for instance making VINS works on standard smartphones for Augmented Reality purposes – the use of a MIMU sensor would inevitably involve hardware designed explicitly for AR. It is not unreasonable to think this specialized hardware would have been thought entirely for this purpose, with a completely integrated data acquisition pipeline. We argue that the development of the algorithm should not be separated from the development of the hardware and that ideally, algorithm and hardware should be developed concurrently with a co-conception approach.

The second part of the thesis will focus on methods that do not require a depth image as

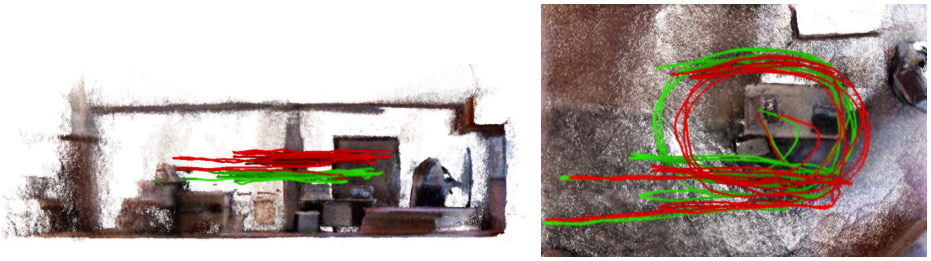
input. Precisely, we will describe dead-reckoning algorithm relying on a MIMU sensor and a regular monocular camera (well calibrated and of superior quality).



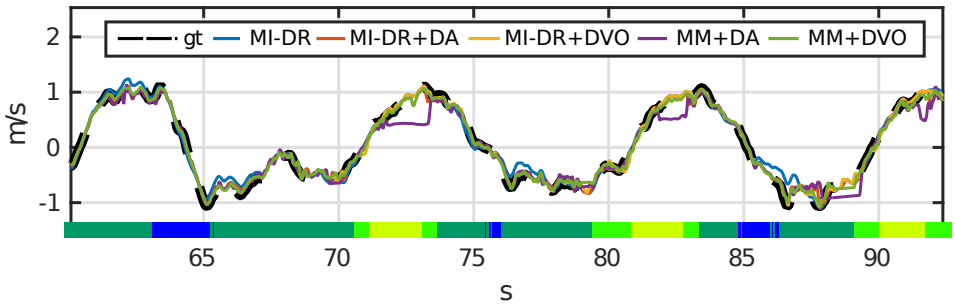
(a)



(b)



(c)



(d)

Figure 3.12: Result for a trajectory in motion capture room; (**a**): Position in inertial frame. The MIMU position estimate present significant drift that is corrected by image alignment, whatever the alignment method used. In turn, the MIMU predicted translation configuration shows more accurate position estimate than with the default motion model, except along the z axis (see Section 3.6.5); (**b**): GT trajectory with color-coded magnetic gradient norm; (**c**): 3D trajectory of the best performing configuration against ground truth. Environment is visualized by reprojecting point cloud from ground truth poses; (**d**): Speed in inertial frame along x axis and local gradient level: Motion Model estimates are sometimes loosing track of the speed and a bias appears on MIMU speed estimate near no-gradient area.

Part II

Tight Monocular Magneto-inertial Fusion for Navigation

Table of Contents

Chapter 4

A Non-exhaustive Review of the State-of-the-Art in VINS

This chapter is an attempt to make a digest state-of-the-art review of existing VINS. *We propose one way to classify* VINS *algorithm and focus mainly on the case where the visual sensor used is a monocular camera. We present selected works from the recent literature that are highly relevant, in our opinion. The review is done from the point of view of algorithm mainly. We will use the presented classification to introduce algorithms presented in subsequent chapters.*

4.1 Objective of this chapter

An exhaustive state-of-the-art presentation about general vision-based navigation – including also standalone vision navigation system – would be too rich for this thesis chapter. The reader interested in a more general review about visual-based navigation and SLAM could refer to the following review papers: [Cadena et al., 2016; Santoso et al., 2017]; we will reference here nearly exclusively methods that use an IMU.

Also, the goal is not to retrace the history of SLAM or VINS systems, and we admittedly exhibit a strong bias towards recent to very recent publications presenting concrete implementations of VINS in generic, AR or drone applications. We also target a comprehensive and contextualized list of pointers towards the most interesting – in our opinion – publications, open-source codes, and datasets. We think these would interest anybody willing to start designing a VINS nowadays.

Despite our bias towards recent work, the reader must keep in mind that rare are the concepts and ideas in visual-inertial navigation that were not already proposed two or even three decades ago: application, context, implementation, and engineering matter.

4.2 Utils: Bayesian Inference, Manifold and Lie Group.

4.2.1 Bayesian Inference in a Nutshell

The most powerful tool for determination of valuable parameters from noisy data is Bayesian inference. In our context, this technique aims at answering the following questions: "Knowing all the observations from my sensor, their noise model, and some prior knowledge on the parameters I seek, how evolves my knowledge about these values? What is then the most representative value of my parameters?"

One way to answer the second question is to solve for the value of parameters whose probability is maximal knowing all data one has about the system. We write formally the MAP (Maximum A Posteriori) estimator as:

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} p(\Theta = \theta | Z, I). \tag{4.1}$$

In the Bayesian approach, the parameter θ to be estimated – in our particular application, this will represent trajectory or geometrical parameters of the movement – is seen as a realization of a

random variable Θ, Z is the random variable corresponding to received observations (polluted by a noise), and I denotes information one has *a priori* on Θ. p denotes here a probability distribution of Θ knowing realization of Z and prior information; this distribution is called the posterior distribution $p(\Theta|Z = z_{obs}, I)$ because it can be written after the observation z_{obs} has been recorded, thus the notation MAP for the estimate.

In practical situations encountered in the problem at hand, the MAP problem can be translated into a non-linear least square minimization problem, which is generally solved by local descent algorithm. This translation goes like this:

Because of Bayes rule one has:

$$p(\Theta|Z = z_{obs}, I) \propto p(Z|\Theta, I)p(\Theta|I) \tag{4.2}$$

Because I carries only information on Θ , that simplifies in:

$$\propto \underbrace{p(\quad Z \quad | \quad \Theta \quad)}_{\text{direct model of sensor}} \underbrace{p(\quad \Theta \quad | \quad I \quad)}_{\text{prior on parameter}} \tag{4.3}$$

Where the direct model of the sensor is assumed to be known beforehand. Applying the monotone function $-\log$ yields another expression of the MAP estimate:

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} \left[-\log(p(Z = z|\Theta = \theta)p(\Theta = \theta|I))\right] \tag{4.4}$$

$$= \arg\min_{\theta} \left[-\log(p(Z = z|\Theta = \theta)) - \log(p(\Theta = \theta|I))\right] \tag{4.5}$$

And, in the case where the sensor model writes: $z = h(\theta) + \eta$ with η a realization of a noise following a Gaussian distribution $\mathcal{N}(0, \Sigma_{obs})$ and that prior also follows a Gaussian distribution on Θ, $\mathcal{N}(\theta_p, \Sigma_p)$, the previous MAP problem writes as a NNLS (Non-Linear Least Squares) problem:

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} \|h(\theta) - z_{obs}\|_{\mathbf{\Sigma}^2_{obs}} + \|\theta - \theta_p\|_{\mathbf{\Sigma}^2_p} \tag{4.6}$$

However in the general case where the measurement errors are not Gaussian distributed we have to write instead the more general formulation:

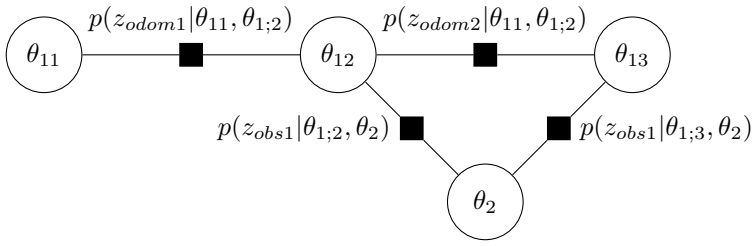
$$\arg\max_{\theta} p(\Theta = \theta|Z = z_{obs}, I) = \arg\min_{\theta} f(h(\theta), z_{obs}) + g(\theta, \theta_p) \tag{4.7}$$

and f and g are log-likelihood terms stemming from hypotheses on the sensor model and prior[1].

One critical aspect here is the correct modeling of the conditional independence of parameters and observations in the inference problem. Indeed, in the case when θ is a vector of parameters and z a vector of observations, it is likely some observations are independent from one set of parameters, conditionally on another set of parameter. From a distribution probability point of view it means we can decompose $p(Z = z|\Theta = \theta)$ into a *product* of factors $p(z_{ij}|\theta_{pk})$ where z_{ij} and θ_{pk} are sub-vector of parameters. Each one of these factors would then corresponds to a new element in the NNLS sum (4.6).

These dependencies can be formalized on a graph such as a Bayesian Network or a Factor Graph, and several in the SLAM estimation community are based on this formalism: [Dellaert and Kaess, 2006; Carlone et al., 2014; Kaess et al., 2012]. These conditional independencies also ultimately reflect the sparsity of the matrices involved in the non-linear numerical solvers, that can and must be exploited for very fast inference or very large-scale problem. The Figure 4.1 shows an example of probability distribution decomposition through its conditional independence pattern. It pictures its associated factor graph, on which black square are representing factors (here the probability of measuring the observation knowing the state). Round elements are representing parameters. The figure also shows a translation of this structure as a least squares cost function that assumes that all factors are Gaussian distributed with unit covariance.

[1] Chapter 5 will suggest some way to choose f and g for our problem



$$p = p(z_{odom1}|\theta_{11},\theta_{1;2})p(z_{odom2}|\theta_{1;1},\theta_{1;2})p(z_{obs1}|\theta_{1;2},\theta_2)p(z_{obs1}|\theta_{1;3},\theta_2)$$
$$C = \|\theta_{2;2} - \theta_{1;1} - z_{odom1}\|^2 + \|\theta_{3;3} - \theta_{2;2} - z_{odom2}\|^2 + \|f(\theta_2,\theta_{1;2}) - z_{obs1}\|^2 + \|f(\theta_2,\theta_{1;3}) - z_{obs2}\|^2$$

Figure 4.1: Representation of a factor graph, its translation in probability density function, and in a NNLS function in the case where all distributions are Gaussian.

4.2.2 Inference on Manifold

In geometric computer vision algorithm and SLAM, differentiable manifolds have a first-class function. Indeed, one has often to solve for a variable that belongs to a Manifold: for instance a rotation of the camera belongs to SO(3), a 3D pose to SE(3) or a similarity transform between two frames (as can be derived by a monocular SLAM system) lies in Sim(3).

However, methods to solve the inference problem, such that the EKF or Gauss-newton optimization algorithm, often implicitly assume a Euclidean space for their variables. This is a problem: first, these methods do not deal well with an over-parametrized version of the element of a manifold, as they explore the parametrization space as if it was euclidean; for instance a gradient descent update on a unit-quaternion by component-wise differentiation would lead to an unnormalized quaternion; secondly, it is often not possible to build a global, and minimal, parametrization of the manifold that does not have singularity points ; for instance there are specific configuration where, infinitesimal perturbation of Euler-angles parameters does not form a basis of the SO(3) tangent space, leading to a degradation of local descent performance.

In order to solve this problem, one can rely on a *local* parametrization of the state when differentiating. This local parametrization is also often used to define an uncertainty distribution: for instance a noisy rotation would be expressed as a average rotation perturbated by a "small" rotation element. This small rotation distribution would be expressed in the tangent space of the average one. The coordinate of the local perturbation could then be assumed to follow for instance a Gaussian distribution.

4.2.3 Filtering Versus Optimization

Two numerical tools are generally employed to solve for the MAP estimate. The first one is iterative optimization that will converge to the MAP estimate if the problem can be cast with a convex formulation, or to a local minimum otherwise – that one generally *hope* to be the MAP. The second is Kalman filtering method, that process the data incrementally as times goes and converges, for linear problems, to the MAP estimate for state variables corresponding to the current time. The Kalman methodology is often applied to non-linear problems though, even if one generally loses their convergence properties. For VINS, both approaches have been proposed: [Leutenegger et al., 2013; Forster et al., 2017; Konolige et al., 2010; Usenko et al., 2016] all rely on an optimization paradigm while [Mourikis and Roumeliotis, 2007; Davison et al., 2007; Paul et al., 2017; Brossard et al., 2017] rely on filtering. In practice, these two tools are generally mixed together in solution of literature. Some work are thus more exactly seen as intermediary between pure non-linear optimization and pure filtering.

The two approaches can be seen as solvers or approximate solvers for the same probability model

and cost function, so that one can think separately of the problem definition (through the model) and of the algorithmic design (through the solver).[2]

Because of the number of parameters to estimate in VINS (the entire trajectory, thus the position and inertial state at each instant), the iterative optimization on the full problem is often intractable: for real-time systems, some approximations have to be made. One approximation is to marginalize from the posterior probability the parameters for which one already has good estimates. These can be, for instance, positions of the device at timestamps too old compared to the current timestamp – in which case we obtain a fixed-lag smoother – but other marginalization strategies can be imagined like the one we describe in Chapter 5.

An important advantage of optimization-based approach compared to pure filtering are their capabilities to update the linearization point of a block of parameters in θ when θ estimation is refined – of course at the expense of more computation. This is of particular interest because linearization error has repeatedly been shown to be a source of significant issues in VINS and visual SLAM filter.

4.3 Two Different Kinds of Approaches for Fusion

From the rich literature on VINS, we can distinguish two approaches to fuse the information from a camera and an IMU. First, a part of vision-community started developing VO (Visual Odometry) and SLAM disregarding the IMU completely. These pure visual systems – whose examples are for instance found in [Mouragnon et al., 2006; Sanfourche et al., 2013; Engel et al., 2013; Forster et al., 2014; Engel et al., 2018] – thus relied exclusively on the images processing to infer the motion. This has some drawbacks : one of them being the relatively low temporal bandwidth of traditional visual sensor. As a result, visual-odometry algorithms performances degrade with the camera is following high dynamic motion. Furthermore, if the framerate of cameras is sometimes set higher than the traditional 20-30Hz (e.g., in [Forster et al., 2014; Engel et al., 2013]) in order to cope with the issue, the computational load associated with high-frequency image processing is not negligible. The idea of using an IMU to help these visual systems emerged in this community as a necessity, however often more as an afterthought. For this reason, a lot of VINS in academic research were developed as extensions of purely visual systems, often using a low-cost IMU, sometimes with rough calibration and integration.

Among instances of this first approach we can cite : [Forster et al., 2014, 2017] which integrates into their advanced VO pipeline a motion prior whose rotational part arises from gyroscopes. They do not fully leverage the IMU, as accelerometers are not used; they prefer a motion model for the translational part of the motion prior instead. They hence avoid the difficulty of precise attitude estimation for gravity addition to the accelerometer's signal. [Forster et al., 2015] is a step forward compared to [Forster et al., 2014]. The odometry algorithm presented in [Forster et al., 2014] is used to initialize a bundle adjustment integrating statistically consistent inertial constraints. [Konolige et al., 2010] uses the IMU in a very simple way: VO provides incremental pose estimation through local stereo bundle adjustment and the IMU is used as inclinometer (absolute roll and pitch) reference and yaw rate measurement. They thus rely on a quasi-static assumption: a sound approximation for the slowly moving robot they used in their experiments that would fail in general. [Tardif et al., 2010] presents a similar approach in which the incremental pose estimation is used to correct the full mechanization equation integration in an EKF. [Mur-Artal and Tardos, 2017] uses the full IMU signal mainly to initialize and find the vertical direction and scale of their visual-constructed map. Albeit they smooth the position estimate with both gyrometers and accelerometers, the original SLAM system [Mur-Artal et al., 2015] is pretty robust on its own already, and it is probable that

[2] However, depending on the methodology chosen for a particular application, some issues related to non modeled effects would be handled quite differently. Robustification in an optimization framework would generally involve slightly tuning the cost function, while in a filtering framework it would be instead done through an adaptive gain scheme. Also, non-linear effects would have to be handled differently, so that the two paradigms involve finally somewhat different viewpoints.

the localization accuracy of the fused system comes mainly from its mapping strategy. The IMU benefits being reduced to very short-term prediction and gravity direction estimation.

This approach has advantages: these methods can cope pretty well with low-cost, not cleanly integrated sensors while being able to estimate quantities not observable with cameras solely: attitude (i.e. gravity direction) and scale (unobservable in the case of a monocular camera). However, these also have some disadvantages: the IMU is often not leveraged to its full potential. One consequence is that these methods require full pose observability *at each instant* from the vision pipeline and can hardly cope with a total loss of visual information. They also tend to be computationally expensive, as the visual pipeline has to be highly robust by itself.

In contrast, another approach aims to use visual information to correct a "good quality" inertial propagation. This is closer to high-end inertial navigation and can be seen as the reciprocal of the previous approach. This latter approach is used for instance in [Mourikis and Roumeliotis, 2007; Li et al., 2014; Paul et al., 2017; Bloesch et al., 2015] or in [Leutenegger et al., 2015; Qin et al., 2017]; the first set of work are using a filtering paradigm, while the second one are using a more computationally demanding optimization framework.

The difference between the two approaches distinguished here is, admittedly a bit blurry and sometimes quite subjective. However, we noticed that this distinction is quite relevant. Furthermore, the author's chosen approach can often either be explained partly by technical reasons – the wish to use off-the-shelf already integrated sensors – and partly by expertise domain reasons: computer-vision researcher and laboratory tends to disregard inertial navigation, and not to be able achieve a high quality integration of sensor by themselves.[3]

In contrast, we argue that the choice of algorithm can *not* be decoupled from the targeted hardware and that both should be developed concurrently, with the same effort: there is at least as many benefits to take from hardware improvement as from advanced algorithm. We tried, during this work, to keep this idea in mind; we were lucky enough to work on nicely integrated MIMU hardware and to get help from experimented engineers at Sysnav in order to use it.

A terminology often used in literature to classify the fusion algorithm is their *tightness*. In principles, *loosely coupled* systems fuse the output of a purely visual position estimator with the output of inertial navigation position, without interaction between both estimators. In contrast, *tightly coupled* approaches fuse information from camera with information from the IMU in the same fusion estimator. This terminology superimposes roughly with the two approaches distinguished above but, again, this distinction does not partition without ambiguity the diverse set of visual-inertial systems. Yet some "level of tightness" is of interest to describe a VINS solution, we will define three level in the following and use them to classify works from literature.

4.4 Dead-eckoning/Vision-Inertial Odometry

4.4.1 Extracting Information from Image Sequences

The exploitation of images for motion and localization requires extracting the information in a series of images. This extraction can be done in diverse ways, which can be classified both by the amount of data extracted, and by the nature of extracted information. The amount of data leads to the *sparse* and *dense* terminology. A sparse method extracts very localized areas of the image that are considered of particular interest, while a dense method uses the whole image. Orthogonally, the nature of extracted data leads to the *indirect* and *direct* terminology. The indirect methods translate the information of the image into geometrically meaningful value – for instance pixel coordinates of a geometric features – while the direct method "recovers the unknown parameters [in our case movement parameters] directly from *measurable image quantities* at each pixel in the image" ([Irani and Anandan, 2000]). VIO algorithms can be classified according to these two criteria in the following four families.

[3] One reason for this fact is the requirement for highly specialized test bench to calibrate IMU with high-precision. Such setups are generally not directly accessible to computer-vision researchers in "standard laboratory".

Indirect+Sparse

Indirect and sparse approaches have been the most commonly used in literature. In this combination, images are summarized as a few geometrical features which are localized in the pixel array. The most common features are corners (i.e. a pixel in the image whose pixels around are majoritarily either darker or brighter) or edges (an area in the image where the image gradient is strong along one direction).

Once detected for the first time in an image, features have to be recognized from one image of the sequence to another. Two categories of methods are used for this temporal matching.

The first one is close to sparse optical flow techniques. These methods try to register (align) the local appearance of the feature in the image pattern translation. The final patch alignment gives the new coordinates of the features. This is generally done iteratively with subpixel accuracy ([Lucas and Kanade, 1981; Bouguet, 2000]). Doing so assume a high enough framerate in order to have small movement between frames, and assumes, a features is to be found around the same coordinates in the next images, then a gradient descent of the difference of intensity between patch to refine this first estimate. This features tracking method is used for instance in [Li et al., 2014; Tsotsos et al., 2012, 2015; Qin et al., 2017; Weiss et al., 2012].

This tracking method is adequate for corner tracking, but not for edges, because of the *aperture* problem (edges position is ambiguous along the direction parallel to the edge). It can be circumvented by restricting the search space to the epipolar line in the next image in the specific case where this line is known at matching step (i.e. if a reasonable estimate of translation direction and rotation is available. See for instance in [Yu and Mourikis, 2017]).

A second method for features tracking relies on an abstracted appearance descriptor. Descriptors are either a binary string or a set of real numbers, associated to a distance, which describe the local appearance of the image. A simple descriptor would be a vectorized version of intensity values around a corner; but this lacks robustness to rotation and scale, change of viewpoint and illumination. More advanced descriptors stem from statistical learning approaches and are designed to present "learned" invariance to the change of appearance created by a range of motions. In such descriptor-based approach, at each frame, instances of features are detected and described. Then the features are matched to features extracted from previous frames thanks to the descriptor distances and some epipolar geometry constraints. If a descriptor is discriminative enough, this allows recognizing features whatever their current location in the new image and thus does not require high video framerate.

Authors of following publications use a descriptor paradigm: [Weiss et al., 2012] uses simply patch intensity while [Konolige et al., 2010] uses a ZNCC descriptor. [Leutenegger et al., 2015] uses a more advanced descriptor called "BRISK" ([Leutenegger et al., 2011]) for corners tracking , BRISK are crafted for robustness to affine and scale transform and lightning changes. [Paul et al., 2017] gives a interesting comparison of the optical flow tracking versus frame-to-frame descriptor matching in a VINS context.

Whatever the tracking paradigm chosen, in an indirect + sparse approach the information of the image is translated into pixel coordinates to feed an estimator.

Direct+Sparse

This paradigm still reduces the image to a finite set of areas in the pixel array but avoids translating the image content into high-level geometrical features. Instead, the raw pixel intensity in these areas is directly given to the position estimator. The position estimator seeks to minimize the photoconsistency error in these areas through a patch alignment process – which is parametrized by quantities of interest for the localization problem, rather than feature position in the image space. For instance, [Bloesch et al., 2015, 2017] uses directly a photoconsistency error as a measurement equation in a Kalman filter whose prediction is computed by the IMU. [Forster et al., 2017] is based on the same idea, the authors combine direct and indirect methods in order to retain the large convergence basin of indirect methods, and accuracy of direct ones. Another relevant work is related in [Engel et al., 2018], using this time an optimization framework to estimate structure and motion, using photoconsistency error instead of the traditional reprojection error of bundle

adjustment algorithms. This work has not been adapted to use an IMU (yet).

Direct+Dense or Direct+Semi-Dense

Some works try to exploit most of the information available in the images. For instance [Engel et al., 2013, 2014a], introduced what they call *semi-dense* odometry, a direct formulation which exploit all areas of image where gradient magnitude is strong enough. Alignment of the entire image then leads to position tracking. [Usenko et al., 2016] extended this purely visual-based method with information from an IMU. Fully dense methods also exist in vision community (see [Newcombe et al., 2011]), but we are not aware of fully dense methods coupled with an IMU. Besides, fully dense methods are generally not efficient for localization because they waste computation budget on image areas that are not informative.

Indirect+Dense

It is not easy to figure what such a method would be. One could imagine for instance that the image processing steps would estimate a geometric quantity densely over the image, such as optical flow, depth or disparity, and then provide these estimate to the position estimator instead of the raw image. In fact, this is exactly what is done in methods relying on the 3D estimate on structured light depth sensor and RGBD sensor. Indeed, these sensors internally transform an *infrared image* to a *dense depth image* that is then used in the position estimator. Hence, our method presented in Sections 3.2 and 3.6 fits perfectly this category. This is similarly the case for all ICP variant of literature using Kinect-like sensors as input, for instance [Izadi et al., 2011; Niessner et al., 2014; Bonnabel et al., 2014a; Jaimez and González-Jiménez, 2015].

4.4.1.1 Requirement for a robust image processing pipeline

Bad features tracking or matching will degrade the estimate severely, features tracking thus needs to be robust both to non-modeled random events that could alter the appearance of features in the image, and to wrong matches due to algorithm mistakes. Several strategies are used; for descriptors matching, consensus-based strategy is the most widely used, generally through a randomized heuristics such as the RANSAC (RANdom SAmple Consensus) algorithm [Fischler and Bolles, 1981]. Depending on the problem and assumptions several variants are used in features tracking: 5pt-RANSAC for two central images, 2pt-RANSAC [Bazin et al., 2010; Troiani et al., 2014] for central images with known rotation. Another alternative is to model the probability of outliers (i.e. a data that does not fit the model assumption) explicitly into the estimator. This is the approach chosen in the features tracking component of [Forster et al., 2017].

However, this is impossible to guarantee that the visual processing frontend will not make any mistake; thus these outliers rejection schemes have to be completed by a robustification of the position estimator itself with respect to its inputs coming from image processing. This can be done for instance by optimizing robust loss instead of L_2 norm [Triggs et al., 2000] or by using *a priori* information on the motion to further remove gross image processing error (for instance 1-pt RANSAC if a covariance on the matching is available [Civera et al., 2010]).

4.4.2 Fusion with an IMU

While one usually distinguishes between *tight* and *loose* estimators, we propose here, unconventionally, to define three levels of "tightness", that correspond to how the image information is actually used in the algorithmic submodule fusing the information with the IMU (submodule which is called *fusion estimator* hereafter). Our terminology is also summarized in Figure 4.2.[4]

[4] Note that our tightness definition is not orthogonal to the choices of Indirect/direct Sparse/Dense visual method, although this terminology was only dealing with the visual processing pipeline, whereas the tightness notion is linked with how the IMU is used.

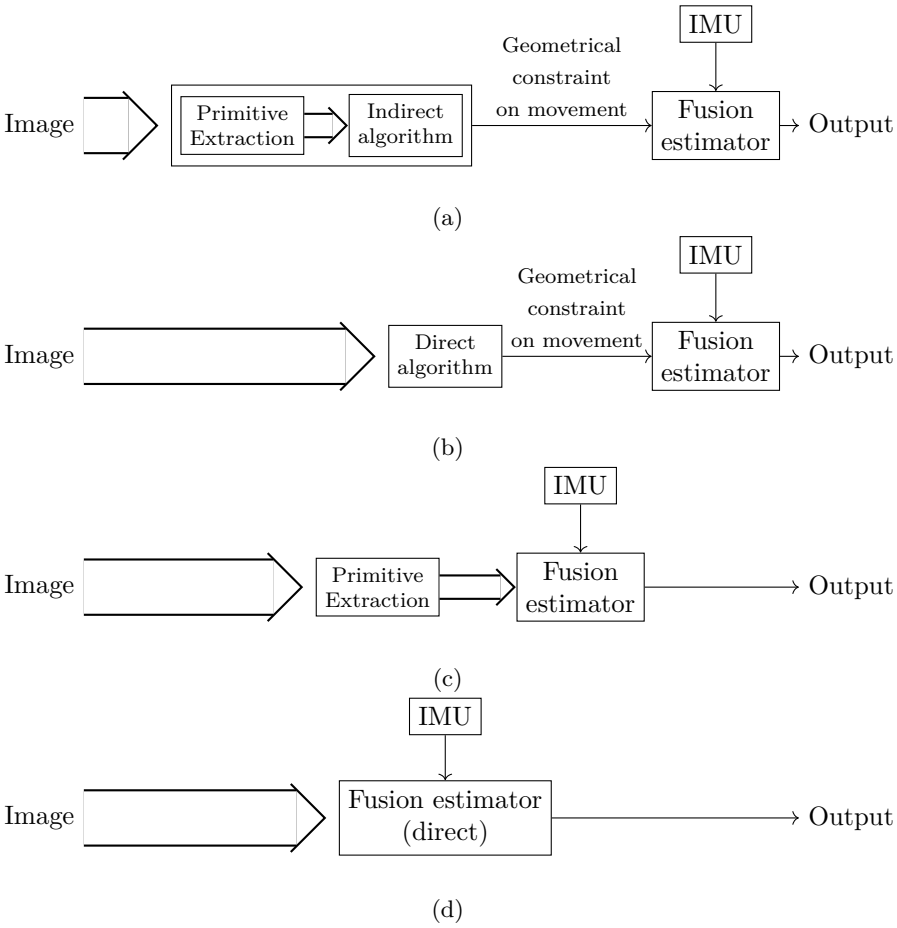


Figure 4.2: Different structure of estimator using our classification. (**a**)-(**b**) two examples of loose fusion based on sparse-indirect and direct visual processing; (**c**) tight fusion; (**d**) "supertight" fusion. (See Section 4.4.2.1)

4.4.2.1 Different fusion tightness

Loose: one vision estimator giving position or relative position that feeds a fusion estimator. In loosely coupled approaches, the images are summarized into geometrical information *on the movement of the camera*. This information is then, in a second step, fed to an estimator whose role is to fuse this piece of information with the IMU measurement. This is particularly handy in stereo ring settings, which are sufficient to retrieve relative metric translation and rotation. This is exploited for instance in [Tardif et al., 2010] and [Konolige et al., 2010].

However, loose approaches have also been tried also in a monocular context: for instance, [Weiss et al., 2012] exploits the epipolar constraint to derive a *scaled* velocity and use this as an update of an EKF. One difficulty of those approaches, though, is to cope with the metric scale estimation. Two scale-retrieval filters (one EKF and one non-linear) are compared in [Grabe et al., 2013]. Both works focus on the restrictive case where a dominant plane is present in the image – which is generally true for quadcopter applications but not in general, and specifically not in the case of interest of this thesis.

As the visual processing is isolated in the loose approach, it can be used indifferently with all combination of direct/indirect and sparse/dense visual algorithm.

Tight: image preprocessing that feeds an estimator It is well recognized that in the case of monocular and IMU fusion, loosely coupled approaches are not optimal. The tight fusion estimator approach can reach higher accuracy by using information extracted from images, without reducing them to a piece of information on the camera movement. This also allows finer selection of image data, leveraging easily information from IMU and past states for features tracking for instance.

This paradigm is used in numerous works. In some, the fusion estimator takes, corners or edges coordinates as input. [Kleinert and Schleith, 2010; Tsotsos et al., 2012; Hernandez et al., 2015; Li et al., 2014; Leutenegger et al., 2015; Paul et al., 2017] all use corner features into optimization or filtering based estimators. The "structureless" visual-inertial bundle adjustment in [Forster et al., 2015] can also be classified as tight fusion. [Yu and Mourikis, 2015, 2017] both work with line features coordinates as input. [Diel et al., 2005; Lynen et al., 2013b] use constraints arising from sparse optical flow measurement into the fusion estimators and work, again, with the assumption of a dominant plane in the image.

In some works (e.g.[Tsotsos et al., 2012; Qin et al., 2017; Leutenegger et al., 2015]), using such feature's image coordinates requires augmenting the estimated state with the 3D position of the corner or edges, in a full *Structure-from-Motion*-like approach.

We also classify into the tight category one approach based on dense visual odometry presented in [Usenko et al., 2016]. In this approach, one unique estimator minimizes an error combining photometric terms and inertial terms, which would advocate to classify it as supertight (see next section). However, the photometric terms expression relies on a dense depth map estimated on a reference image; this depth map is originating from a *separated* triangulation process, which uses both temporal and stereo matching. The fact that part of the image processing is done separately from the position tracking makes us classify this method in tight albeit the distinction starts to blur in this case.

Super-tight: one unique estimator handle IMU and raw image information directly Finally, the last approach, that we call here "super-tight" formulate directly one fusion estimator that takes as input areas of images, without any or with few image processing transformation (no features tracking or dense depth map generation). The perfect example of such an estimator is given in [Bloesch et al., 2017], where an Iterated Extended Kalman Filter is used for the fusion. The measurement equation derives from a photometric error term such that minimizing this term corresponds to an KLT (Kanade–Lucas–Tomasi feature tracker) iterative tracking of features, but whose iteration are constrained by the filter covariance on features depth and position.

One extension of the SVO algorithm of [Forster et al., 2017] can also be seen as an instance of super-tight fusion. The authors process, into the same optimization, relative rotation between

frames coming from gyroscopes with direct error measurement to estimate position and depth of features simultaneously. However, they disregard the estimation of sensors biases.

An extension of the direct optimization-based method of [Engel et al., 2018] with the inertial error of other indirect bundle adjustments (as in [Leutenegger et al., 2015]) would also lead to an instance of a super-tight algorithm. If no such work has been published yet, we guess this is only a matter of time.

In the end, which level tightness should we choose? Loose fusion is well-suited for integrating 3rd-party sensors into a system; for instance, a high-end IMU/GNSS product integrating closed source proprietary processing software. In this case, loose fusion can be exploited to leverage expertise contained in the commercial product software without having to develop it oneself. Loose fusion does not bother with accurate sensor modeling and is then easier – and thus faster – to implement. It could also be used in the case of very bad quality sensors, for which information could be seen more as a "hint" and not a proper measurement with Gaussian noise. However, as a drawback, loose fusion cannot leverage the sensor complementarity fully, requires a consolidated vision-based pose estimate at each instant, and has repeatedly been shown to be outperformed by tight estimator for VINS. We argue that except for the case of sophisticated high-end 3rd-party sensors, we should prefer tight fusion, especially in the problem at hand.

It is not clear which of the tight and super-tight category is better. They mainly differ from how the visual information is handled. The question of which to choose boils down to the question of which visual processing paradigm is better between direct and indirect. This question has not been soundly answered in the community (see for instances the very thorough comparison done in [Yang et al., 2017; Engel et al., 2016; Platinsky et al., 2017]).

In a nutshell, indirect methods are naturally less sensitive to lousy illumination conditions and mis-calibration while direct methods are more sensitive to these sources of error and require camera photometric calibration to reach their full potential. However, direct methods easily handle poorly structured scenes, with few strong corners or edges in contrast to corner-based methods. Also, indirect methods often provide a broader convergence basin as their cost function tend to be a bit more convex compared to direct methods as claimed in [Engel et al., 2018].

4.5 Dead-Reckoning plus Localization

One very useful practical application is the combination of dead-reckoning technology with a visual relocalization system. This allows estimating an absolute position in a map reference frame, whereas dead-reckoning output can only be given in its own reference frame. In contrast to full SLAM, these systems do not have to build the map simultaneously and thus are generally less complex and computationally more tractable. We foresee this DR+Reloc approach will be the instance of VINS the most useful in the future, especially in AR or autonomous automotive.

It is here assumed that a map (even incomplete or partially obsolete) of the environment is given. This map can, for instance, be constructed through batch visual-inertial bundle adjustment beforehand, or even exploiting eventually a more complete/expensive set of sensors. Being an offline process, this map construction could be built on an exact solution to a statistically sound MAP solution.

In this case, one thus needs a way to (i) relocalize into the map, (ii) use this information to correct the dead-reckoning process.

Generally, these methods are built around a fast and low-latency VIO algorithm for local motion consistency and a slow and high-latency correction for localization in the map.

[Middelberg et al., 2014] proposes for the relocalization step a remote localization server architecture that takes from VIO a local 3D map of features, matches it with a global 3D point-cloud in memory and sends a position correction back to the dead-reckoning system. The optimization

involved in VIO is then augmented with error terms that are built from differences of position between the local pose estimate and the global pose from the server.

[Lynen et al., 2015] uses an expensive image retrieval strategy based on bag-of-(binary)-words where word dictionary is actually a discretization of the space of features descriptors (image retrieval and descriptors matching is a vast subject on its own). They then retrieve 3D-2D matches that they use individually as a measurement equation into their tight fusion filter, effectively constraining the global position.

One problem the last approach is that the map is seen as exactly known at runtime, however, the construction of the map should be assumed to be polluted by error as well; [DuToit et al., 2016, 2017] extend previously cited references by integrating map uncertainty considerations, and studying the effect of the relocalization pipeline on the consistency of their VIO EKF. Compared to previous works, they also boost runtime performance by exploiting the prior position knowledge to bypass image retrieval logic except for the first map relocalization.

Similar techniques are used in the open source framework [Schneider et al., 2017].

Note that all these approaches exploit specifically created map. The exploitation of already existing data would be of a strong interest in relocalization application. One example of such development is given in [Larnaout et al., 2012; Antigny et al., 2017] that uses GIS data information or urban furniture position to constraint the localization in urban areas. If, for indoor applications, such data are often inexistent, no doubt that, this kind of data would be highly valuable for large-scale urban AR.

4.6 SLAM

Regarding VINS SLAM, loose fusion approaches have been used in [Shen et al., 2013; Lynen et al., 2013a; Engel et al., 2014b]. Among those, [Lynen et al., 2013a] estimates explicitly the scale factor of a full-featured monocular SLAM subsystem by feeding its output (a position in its scaled map) into an EKF taking also IMU as input – the scale factor is there a state of the filter. [Engel et al., 2014b] fuses the output of a monocular SLAM position with the navigation filter of a commercial drone: the SLAM position is scaled correctly through fusion with a pressure sensor, and an EKF fuses it with pose increment from the drone internal odometry and motion model. [Shen et al., 2013] uses similar ideas, but regularly retrieves the monocular map scale thanks to a very low-frequency second camera, they fuse this also scaled pose through an EKF that uses the accelerometer, gyrometers and mechanization equations for the prediction directly.

Among tighter fusion scheme, the majority of systems relies on a dual-layer scheme: lower-layer is an efficient VIO sub-modules – built with a tight approach – while the upper-layer handles long loop closure to correct the significant drift in the trajectory. This upper layer relies on full visual-inertial bundle adjustment cost function [Mourikis and Roumeliotis, 2008] or an approximation of it, [Qin et al., 2017; Nerurkar et al., 2014].

There also exist tight single-layer VINS SLAM. For instance [Mur-Artal and Tardos, 2017] integrate into their full-featured optimization-based monocular SLAM solution error terms arising from IMU. [Neunert et al., 2016] uses one filter with persistent positions of fiducial markers in the state, but their solution is hence restricted to small specially equipped areas.

From the point of view of statistical consistency, however, one difficulty of VINS SLAM, is to keep a consistent estimate while separating the inference problem into a fast low-latency dead-reckoning and a high latency map refinement process. All previous tight approaches are either not adequate for long-term operation or either rely on approximations of the MAP estimate – employed in the offline map building process.

We are aware of work trying to handle this specific issue ([Kaess et al., 2012]) explicitly, but their solution has not been widely used to our best knowledge. In our opinion, it is not clear nowadays how the frontier between the map inference/refinement processes and the dead-reckoning should be handled in the general case.

4.7 Other Considerations and Topic of Research

Previous sections focused on outline algorithm used in monocular VINS. The present section briefly relates interesting works in other research topics that are useful for monocular VINS design.

Estimator initialization

One of the issues of the VINS problem in its standard formulation is that, in general, the MAP estimator translates into a non-convex and non-linear problem. Thus, statistically consistent and precise solution of the problem relies on a good first estimate. At initialization time, no estimate of pose and – more importantly of attitude – exists a priori. Systems then either rely on assumptions on the world, on the dynamic or on the state, or either try to formulate an (approximated) convex or closed form solution to the problem to initialize more accurately iterative estimators. [Mur-Artal and Tardos, 2017] proposes to start tracking the movement with a pure monocular method and initialize attitude lately and scale through a cascade of optimization problem: first solving for gyroscopes biases, then attitude, and accelerometer biases. [Yang and Shen, 2016] proposes a linear approximation of VINS cost function that can be used to estimate attitude, velocity, accelerometer biases and camera-IMU extrinsic parameters and scene geometry. But gyrometers biases have to be known. In [Martinelli, 2013a; Kaiser et al., 2017], the authors propose a method to estimate speed, attitude, and biases, assuming the extrinsics are known. They particularly focus on the minimal data cases, in order to design fast initialization methods.

Embeddability and Power Consumption

Recently, some work focused on reducing VINS and specially VIO power consumption: [Hong et al., 2014; Zhang et al., 2017c,b] Several ways are explored, from algorithm performance/accuracy trade-offs to specific hardware choice (FPGA, general purpose or specialized co-processor, etc.). Particularly, the work of [Zhang et al., 2017c] is very complete. The authors show that a VIO based on the same principles that are presented in the Chapter 5 of the present thesis can be implemented to consume less than 2 watts of power – which would correspond, after a rough computation of ours, to more than eight hours of operation on a standard smartphone battery.

Estimate consistency

The consistency of one estimator is linked to its capability to predict its own error. Having an accurate error estimation is useful in some VINS applications *per se*. In the case of a double layer VIO+Reloc or of SLAM, this uncertainty can be used by the upper-layer ([Nerurkar et al., 2014]) or by the relocalization submodule [DuToit et al., 2017].

The final consistency of one estimator depend on many factors: from precise sensors error modeling to smart handling of linearization errors. Visual navigation development has a long history of analysis of the algorithmic sources of such inconsistencies.

The NEES (Normalized Estimation Error Squared) metric can be used to assess the consistency of a Bayesian estimator. This consistency is cumbersome to assess on real data and is generally calculated in simulated environments, which is ideal for tracking algorithmic sources of inconsistencies.

The Chapter 7 will deal with estimator properties related to consistency and will review the literature briefly on the issue of consistency in VINS.

Alternative approaches: new sensors

Some approaches for VINS are more experimental at the time of writing. On a sensor side, some work leverage the so-called *event-camera*: instead of sending images at fixed framerate, these sensors send asynchronous pixel events, triggered by a change of received intensity for each pixel. In this particular case, a lot of visual processing have to be reinvented. Diverse teams are researching in this direction right now, and some of them fuse event-camera with an IMU for dead-reckoning: [Zhu et al., 2017; Vidal et al., 2018].[5] The main advantages of such a camera are their high temporal

[5] despite using a very different imaging device, these methods perfectly fits the *indirect tight* paradigm by adapting visual corner tracking on event time series instead of video stream

bandwidth and high dynamic range pixel, which could ultimately reduce the use of an IMU to a gravity alignment sensor.

Alternative approaches: the deep-learning paragraph
Finally, the author of this thesis admittedly disdained the deep-learning (r)evolution that still kept growing in computer vision during the three years of the doctoral work. In fact, visual navigation methods have been a safe-harbor: they were nearly unaffected by deep-learning hegemony until now. Attempts to apply black-boxes deep-learning techniques to the VINS problem does not show satisfactory results for 6DOF low-latency, high accuracy, high-frequency pose estimation (see [Rambach et al., 2016; Clark et al., 2017]).[6] Nevertheless, this situation might not last for long. If we firmly do not believe in end-to-end differentiable and learned VINS algorithm for our application, a lot of its submodules could be improved by deep-learning.

Letting aside the undoubted and obvious benefits from integrating high-level semantic information – which deep learning excels at inferring – into the SLAM and relocalization module, one might think about other direct benefits: on the image processing part [DeTone et al., 2017; Yi et al., 2016] method could supersede corner detection and tracking for more meaningful corners and a more robust tracking[7], [Czarnowski et al., 2017] ideas (deep-learned features registration for robust rotation inference between two images) could help bringing some robustness to direct methods; on the estimation part [Li et al., 2017] could help to recover the scale of monocular+inertial setup in the case of a uniform velocity motion through a strong learned visual prior; finally, activity recognition from inertial data [Yang et al., 2015; Um et al., 2016] could help classify the current motion at each timestamp, which in turn could be useful for adding learned activity-dependant motion prior into more traditional filtering methods.

Nevertheless, the main issue of the deep-learning techniques is their computational and memory requirement at runtime, which often exceeds by far traditional approaches for low-level image processing and computer vision. This has and would surely slow down their adoption for our use case.

4.8 Interesting Available Resources

In a tradition of publishing open-source versions of SLAM algorithms[8], some resources have been made available by some research teams for the the research community. We point towards the most interesting works we are aware of in Table 4.1 and Table 4.2. The former lists self-contained open-source estimators, while the latter lists datasets that can be used to develop, test, and benchmark VINS algorithms.

If these open-source algorithms are a good starting point, they surely are not as robust as industrial VINS. The datasets also are good baseline, but somewhat restricted, they would not suffice to engineer a complete and robust VINS in our opinion.[9]

4.9 Position of the Work Presented in Following Chapters

In this chapter, we have exposed concepts used and attempted a classification of VINS algorithms. We gave for each type, examples from very recent literature.

We are now going to use this terminology to position the work presented in Chapters 5 to 7.

First, we will only focus on the dead-reckoning subsystem (called VIO in this chapter). We still do not address the problem of localization, relocalization or SLAM. Secondly, we will use an

[6] and in our opinion, there is little to no sense applying deep-learning methods to learn the mechanization equations of inertial navigation.

[7] which is totally in phase with features detector/descriptor history: they became more and more based on a learning process across times.

[8] which has historically been demonstrated as being the best way to communicate about one's research in vision-based navigation (see the exceptional success and dissemination of [Klein and Murray, 2007])

[9] the late contribution of [Schubert et al., 2018], added after the review of the thesis report, increases drastically the quantity and quality of the available public resources.

Name	Publications	s/d	i/d	IMU	Inference	Type
ORB-SLAM	[Mur-Artal and Tardos, 2017]	s	i	not released	bacth optim	SLAM
DSO	[Engel et al., 2018]	s	d	not yet	optim+marg	DR
SVO	[Forster et al., 2017]	s	i/d	gyr only	optim	DR
OKVIS	[Leutenegger et al., 2015]	s	i	yes	optim+marg	DR
MSCKF	[Paul et al., 2017]	s	i	yes	EKF filter	DR
ROVIO	[Bloesch et al., 2015]	s	d	yes	EKF filter	DR
MAPLAB	[Schneider et al., 2017]	s	i/d	yes	EKF filter[i]	DR+reloc

[i] but uses offline batch optimization for map construction

Table 4.1: Released research VINS algorithm; s/d is for sparse/dense; i/d is for indirect/direct.

Name	Publications	synced	application
Euroc Datasets	[Burri et al., 2016]	yes	quadrotor (indoor)
Mars-VINS	[Paul et al., 2017]	clk-drift	pedestrian (indoor)
Canoe-VINS	[Miller et al., 2018]	yes	canoe (outdoor)
Zurich-Urban	[Majdik et al., 2017]	yes	quadrotor (outdoor)
PennCOSYVIO	[Pfrommer et al., 2017]	mixed	pedestrian (indoor)
MaplabCLA	[Schneider et al., 2017]	yes	pedestrian (indoor)
PerceptIn Ironsides	[Zheng et al., 2018]	yes	robotic (arm)
ShutTUM[ii]	NotReleasedYet	yes	pedestrian (indoor)
ViTUM	[Schubert et al., 2018]	yes	pedestrian (indoor)

[ii] IMU format given is not exploitable yet for serious VINS fusion,
it could be eventually fixed. For now this dataset is mainly useful for pure visual odometry.

Table 4.2: VINS Public Datasets

indirect approach that extracts corners in the image and tracks them by sparse KLT tracking. In our implementation features will be extracted independently from the position estimate; thus our methods would be classified as *tight* fusion. The choice has been made for two reasons: first, because it has repeatedly shown to be more fruitful than loose fusion, secondly, and in contrast to "supertight" fusion, an indirect tight approach still permits to separate the image processing step from the fusion estimator. This renders these approaches more easy to begin with, as we can prototype the vision frontend rapidly with the help of open-source resources and makes the solution architecture more modular, which is an advantage from an implementation point of view.

We will present two kinds of estimators, one based cost minimization in Chapter 5 and one based on filtering in Chapter 6.

Chapter 5

A joint-optimization approach

In this chapter, we solve the magneto-inertial monocular sensor fusion problem through a non-linear least-squares optimization technique. We build a joint minimization problem between poses, speed, magnetic field and visual landmark positions and present a way to solve it for real-time dead-reckoning purposes. Section 5.1 recalls the cost functions that are usually used in visual-inertial bundle adjustment and SLAM*, and introduces the notion of preintegrated measurement for accelerometers and gyrometers. The Section 5.2 highlights the main contribution of the chapter: the extension of the preintegration technique to express a full magneto-inertial robust error term. The Section 5.3 recalls a first order method to optimize the cost function on manifold. Finally, the Sections 5.5 and 5.6 show an application of the derived error term to a sliding window smoother solving for magneto-visuo-inertial dead-reckoning. A preliminary version of this work was presented at the* 2017 International Conference on Intelligent Robots and Systems *[Caruso et al., 2017b].*

5.1 Visual-inertial bundle adjustment

5.1.1 Visual Only Cost Function

As seen in Chapter 4, numerous modern monocular SLAM systems (for instance [Klein and Murray, 2007] or [Mur-Artal et al., 2015]) can be seen as a real-time incremental solution to the bundle adjustment problem. Bundle adjustment in a pure visual case is well studied and is described in [Triggs et al., 2000] as "the problem of refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameters (camera pose and calibration) estimates". The name "Bundle Adjustment" refers actually in the literature more to the cost function of the problem at hand than to the algorithm used to minimize it. This cost is often written as a non-linear least-squares. If we assume that the intrinsic calibration of each camera is known, the cost for an instance of the problem with cameras indexed by i and landmarks indexed by j writes:

$$\mathcal{E}(\mathbf{X}) = \sum_{(i,l)\in\mathcal{O}} \|\mathbf{r}_{\text{proj};il}(\boldsymbol{\xi}_i, \mathbf{l}_l^w)\|^2_{\boldsymbol{\Sigma}_\mathcal{C}}\,, \tag{5.1}$$

where $\mathcal{O}$ is the set of pair of index (i,l) for which camera i at pose $\boldsymbol{\xi}_i = (\mathtt{R}_i^w, \mathbf{p}_i)$ sees the landmark l at 3D position $\mathbf{l}_l$. $\mathbf{X}$ is an element of the Cartesian product of all variables spaces, $\boldsymbol{\xi}$ and $\mathbf{l}$ of the problem. The figure 5.1a sketches a situation associated to this cost function. The symbol $\mathbf{X}$ will serve as the general unknown of the minimization process throughout this chapter. We will name it, very generically, the *state*.

The residual $\mathbf{r}_{\text{proj};il}$ is generally chosen as the difference – in pixel coordinates – between the observation of the landmark in image i, $\boldsymbol{o}_{il}$, and the predicted observation, knowing the camera pose $\boldsymbol{\xi}_i$ and the landmark position $\mathbf{l}_l$. Formally, it writes:

$$\begin{gathered}\mathrm{SO}(3)\times\mathbb{R}^3\times\mathbb{R}^3 \to \mathbb{R}^2 \\ \mathbf{r}_{\text{proj};il} : \boldsymbol{\xi}_i, \mathbf{l}_l \mapsto \pi\left(\mathtt{R}_i^\mathsf{T}\left(\mathbf{l}_l^w - \mathbf{p}_i\right)\right) - \boldsymbol{o}_{il}\end{gathered} \tag{5.2}$$

with π the camera projection function.

Cost function (5.1) is generally solved by a gradient-based first order or quasi-Newton methods, such as a Gauss-Newton, Dogleg or Levenberg-Marquardt algorithms.

One of the problem of dead-reckoning or SLAM systems that relies upon only the cost (5.1), is that this cost is invariant by a global change of scale. Hence, the reconstructed poses and geometry scale factor is ambiguous. In approaches solving (5.1) incrementally on sliding window (such as [Mouragnon et al., 2006]), this can even lead to severe scale drift between different part of the trajectory. For this reason some constraints on the scale have to be found and integrated into the cost. These constrained can be derived from a priori knowledge on the environment [Lothe et al., 2010] or on the movement [Scaramuzza et al., 2009]. They can also be derived from other sensors measurement function, in our case, an IMU.

5.1.2 Visual-inertial Cost Function

One advantage of the cost function formulation is that it can be extended to account for other sensors. As these sensors do not necessarily relate frame poses and landmark positions directly, the price to paid to add additional sensor is often a growth of the state dimension through the inclusion of *secondary* variables. These variables are qualified here as *secondary* because they are not of direct interest but are unknown and necessary for the cost function evaluation and optimization. In fact, they are even sometimes called *nuisance parameters* or *ancillary variables*.

In particular, the bundle adjustment cost function can be extended to take into account the information from calibrated raw IMU data captured between two successive frames. Please refer to Figure 5.1b for a schematic situation associated to such cost function.

In this case, it is required to include 3 secondary variables by frame: (i) the instantaneous speed at frame timestamp, (ii) the accelerometers biases at frame timestamp, and (iii) the gyrometers biases at frame timestamp. We name this set the *imu state* and note their union $\boldsymbol{s}_i = \left[\mathbf{v}_i, \mathbf{b}_{g_i}, \mathbf{b}_{a_i}\right]$

The monocular cost function is then extended as:

$$\mathcal{E}(\boldsymbol{X}) = \sum_{(i,l)\in\mathcal{O}} \|\mathbf{r}_{\text{proj};il}(\boldsymbol{\xi}_i, \mathbf{l}_l)\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} + \sum_{i=0}^{N-1} \left\|\mathbf{r}_{\text{imu};i}(\boldsymbol{\xi}_i, \boldsymbol{s}_i, \boldsymbol{\xi}_{i+1}, \boldsymbol{s}_{i+1})\right\|^2_{\boldsymbol{\Sigma}_{\text{imu};i}} \tag{5.3}$$

where we introduced the function $\mathbf{r}_{\text{imu};i}$ that correspond to a constraint between subsequent poses arising from IMU data, and $\boldsymbol{\Sigma}_{\text{imu};i}$, the covariance of this residual. We also introduced N, the number of frame, and we assume now that these frames are temporally ordered.

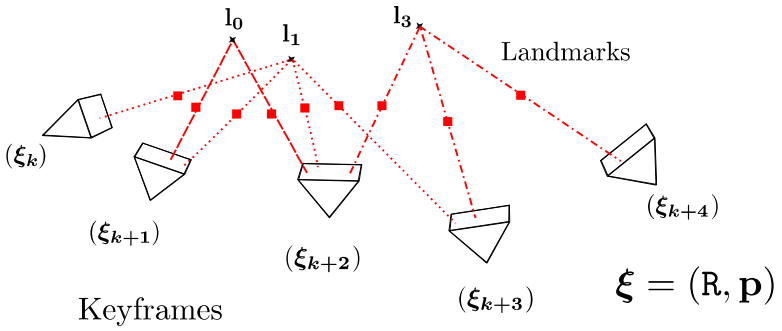
How to efficiently build the residual $\mathbf{r}_{\text{imu};i}$ was the object of recent publications [Lupton and Sukkarieh, 2012], [Forster et al., 2015], [Eckenhoff et al., 2016]. We will recall their technique in Section 5.1.3.

Remark: In the cost function (5.3) and now for the other cost functions that will appear in this chapter, the $\boldsymbol{\xi}$ variables will represent the pose of the *IMU frame* – generally centered around the accelerometer position – and not the one of the camera frame, as it was the case in the previous section. Reprojection error computation will take into account additional transformation due to the rigid body transform between the camera frame and the IMU frame.[1]

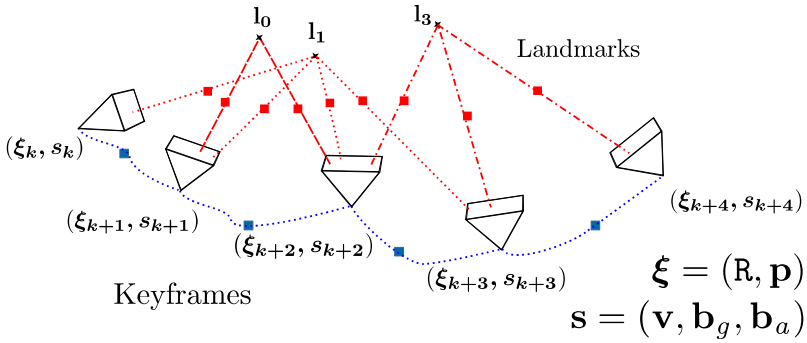
5.1.3 Inertial Residual and Preintegrated Inertial Measurement

This section describes how the IMU residual $\mathbf{r}_{\text{imu};i}$ and its covariance $\boldsymbol{\Sigma}_{\text{imu};i}$ can be computed. We first present in Section 5.1.3.1 a general way to transform any continuous evolution model as an error term in a non-linear optimization algorithm, then present the efficient approach traditionally used in the visual-inertial case as popularized by [Lupton and Sukkarieh, 2012] and [Forster et al.,

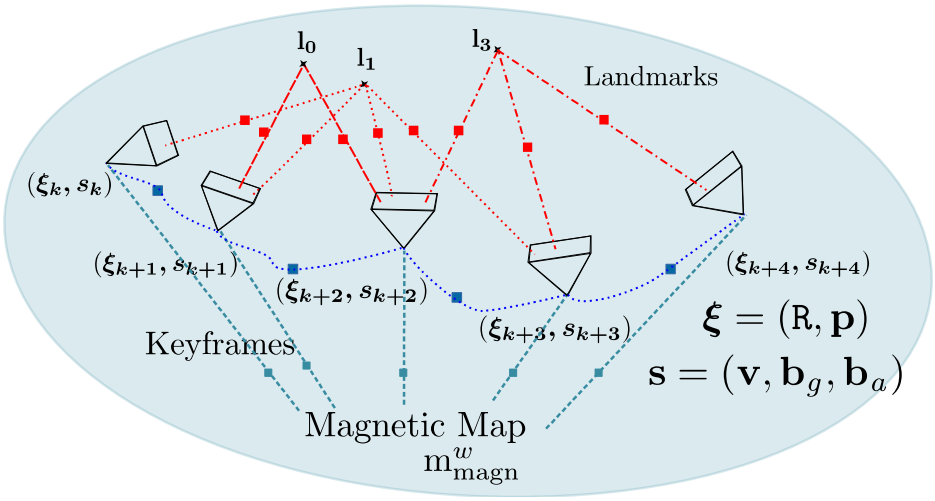
[1] We assume this transform is known after offline calibration step, but it could also be part of the state and be estimated as a secondary variable.



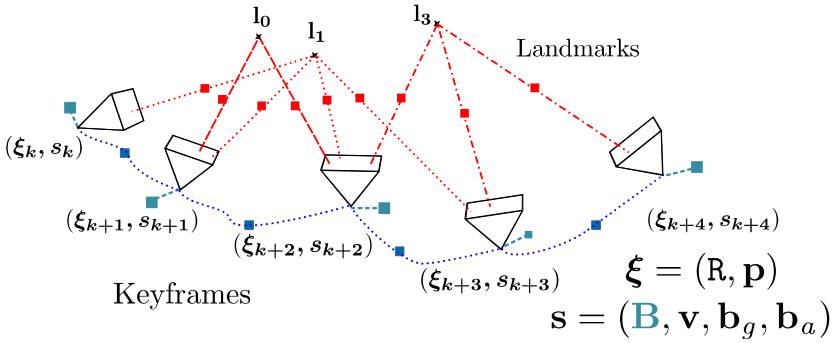
(a) Structure of bundle adjustment problem.



(b) Structure of visual-inertial bundle adjustment problem.



(c) Structure of visual magneto-inertial problem with magnetic map.



(d) Structure of visual-magneto-inertial problem used in this work.

Figure 5.1: Bundle adjustment problems

2015]. The aim of this section is to build the foundation on which we can base a useful and efficient MIMU error term, which will be done in Section 5.2.

5.1.3.1 A general way to use continuous propagation model as an error term between subsequent frame

The continuous model of Chapter 1 fed by received input measurement between time t_i and time t_j could be used to propagate from the pose/imu estimate at time t_i to the pose/imu estimate at time t_j:

$$\mathcal{N}\left((\hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{s}}_i), \mathbf{0}\right) \underbrace{\longrightarrow}_{\text{propagation}} \mathcal{N}\left((\boldsymbol{\xi}_{\text{pred;j}}, \boldsymbol{s}_{\text{pred;j}}), \boldsymbol{\Sigma}_{\text{pred;j}}\right) \tag{5.4}$$

Where, in practice, the computation involves several Kalman filter propagation step, including first order propagation of uncertainty. Then the difference between the state $(\boldsymbol{\xi}_{\text{pred}}; j, \boldsymbol{s}_{\text{pred;}}j)$ and the current estimate $(\hat{\boldsymbol{\xi}}_j, \hat{\boldsymbol{s}}_j)$ – the one we are optimizing on – can be used to build a residual term between time t_i and t_j:

$$(\hat{\boldsymbol{\xi}}_{\text{pred;j}}, \hat{\boldsymbol{s}}_{\text{pred;j}}) \boxminus (\hat{\boldsymbol{\xi}}_{\text{j}}, \hat{\boldsymbol{s}}_{\text{j}}) \propto \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{\text{pred;j}}}\right) \tag{5.5}$$

Where we used an abstract $\boxminus$ operator to form the residual, which, to the first order, behaves as a centered Gaussian vector which covariance is known from the propagation step. The Jacobian - needed in an gradient based optimization – could be computed by the chain rule applied through the propagation steps (5.4).

One problem with this approach though, is that both predicted j mean value and covariance depend on the initial estimate, through the propagation process. However, this dependence is not made explicit in closed-form. This means that the propagation has to be recomputed each time the i^{th} pose/imu state changes. This would be inefficient to do in an iterative solver, for which estimate changes potentially at each iteration. This is especially true in the case of interest where the preintegrated measurements will be used to create a constraint between images. These images are received at a rate order of magnitude smaller than the IMU sample rate. Thus, numerous inertial measurements are to be integrated to build the residual between successive frames. This technique

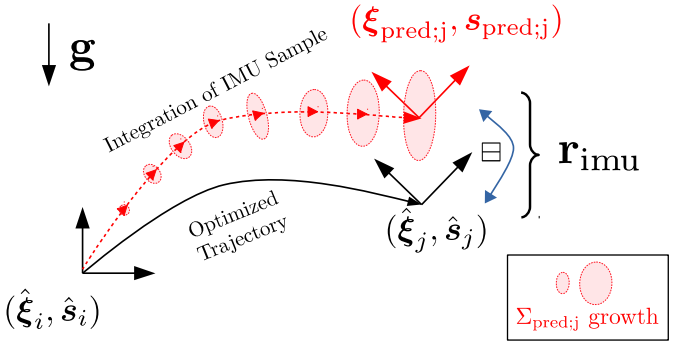


Figure 5.2: IMU residual computed with a propEKF Strategy (Section 5.1.3.1)

can be applied as soon as we have a continuous model for the evolution of the state. We name it "propEKF error term" for convenience. This propEKF strategy is for instance described and used in the context of Visual-inertial bundle adjustment in [Leutenegger et al., 2015].[2] A diagram presenting this technique is depicted on Figure 5.2.

[2] Note that, even if [Leutenegger et al., 2015] describes indeed IMU error term construction with a propEKF strategy, their later open-sourced implementation (https://github.com/ethz-asl/okvis) is based on the preintegrated measurement presented in Section 5.1.3.2

5.1.3.2 Exploiting the structure of inertial propagation

In the case of inertial constraint between subsequent frames or keyframes of a visual-based optimization, it is actually possible to avoid recomputing propagation at each iteration of the nonlinear solver. This involves a slight change of the definition of the residual error that better exploits the structure of the inertial propagation equations. This idea was introduced by [Lupton and Sukkarieh, 2012] and is often used since in optimization based VINS. We describe it in this subsection.

But before that, please note that the following computations assume the knowledge of the sensors biases – the effect of unknown biases will be however discussed at the end of the section. Also, in order to ease the reading of the following, we use a blue color and a tilde accent to distinguish quantities that can be computed directly from received measurements (up to the measurement noise and biases knowledge), without any dependencies on the state estimate. State estimates are hereafter accented with a hat. We also simplify notation $\mathbf{x}(t_i)$ to $\mathbf{x_i}$ when t_i refers to an image timestamp.

Recalling (1.7)-(1.9), Page 15 of Chapter 1, integrating the body evolution model differential equation of strapdown flat-earth body dynamic between t_i and t_j yields:

$$\mathtt{R}_j^w = \mathtt{R}_i^w \widetilde{\Delta\mathtt{R}}_{ij}, \tag{5.6}$$

$$\mathbf{v}_j^w = \mathbf{v}_i^w + \mathbf{g}^w \Delta t + \mathtt{R}_i^w \widetilde{\Delta\mathbf{v}}_{ij}, \tag{5.7}$$

$$\mathbf{p}_j^w = \underbrace{\mathbf{p}_i^w + \mathbf{v}_i^w \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t_{ij}^2}_{\text{Free Fall}} + \mathtt{R}_i^w \widetilde{\Delta\mathbf{p}}_{ij} \tag{5.8}$$

Where Δt is the duration between t_i and t_j and we recall also the expressions of the blue integrals:

$$\widetilde{\Delta\mathtt{R}}_{ij} \overset{\text{def}}{=} \Delta\mathtt{R}_i(t_j) \overset{\text{def}}{=} \mathbf{I}_3 + \int_{t_i}^{t_j} \Delta\mathtt{R}_i(\tau) \left[\boldsymbol{\omega}^b(\tau)\right]_\times \mathtt{d}\tau \tag{5.9}$$

$$\widetilde{\Delta\mathbf{v}}_{ij} \overset{\text{def}}{=} \widetilde{\Delta\mathbf{v}}_i(t_j) \overset{\text{def}}{=} \textstyle\int_{t_i}^{t_j} \Delta\mathtt{R}_i(\tau)\mathbf{a}^b(\tau)\mathtt{d}\tau \tag{5.10}$$

$$\widetilde{\Delta\mathbf{p}}_{ij} \overset{\text{def}}{=} \widetilde{\Delta\mathbf{p}}_i(t_j) \overset{\text{def}}{=} \textstyle\int_{t_i}^{t_j} \widetilde{\Delta\mathbf{v}}_i(\tau)\mathtt{d}\tau \tag{5.11}$$

These quantities are decorated with a tilde to signify they are actually integrated measurements. In fact, the authors of [Lupton and Sukkarieh, 2012] calls these integrals *preintegrated measurements*. They suggest building the inertial error terms between subsequent frames using the following "delta quantities":

$$\Delta\hat{\mathtt{R}}_{ij} \overset{\text{def}}{=} \hat{\mathtt{R}}_i^{w\mathsf{T}} \hat{\mathtt{R}}_j^w \tag{5.12}$$

$$\Delta\hat{\mathbf{v}}_{ij}^{\ b_i} \overset{\text{def}}{=} \Delta\hat{\mathtt{R}}_{ij}^{\mathsf{T}} \hat{\mathbf{v}}_j^{b_j} - \hat{\mathbf{v}}_i^{b_i} - \hat{\mathtt{R}}_i^{w\mathsf{T}} \mathbf{g}^w \Delta t \tag{5.13}$$

$$\Delta\hat{\mathbf{p}}_{ij}^{\ b_i} \overset{\text{def}}{=} \hat{\mathtt{R}}_i^{w\mathsf{T}} \left[\hat{\mathbf{p}}_j^{\ w} - \hat{\mathbf{p}}_i^w - \frac{1}{2}\mathbf{g}^w \Delta t^2\right] - \hat{\mathbf{v}}_i^{b_i} \Delta t \tag{5.14}$$

These quantities can be computed from the current estimate of each state variables and represent actually poses and speed differences – expressed in the body frame at time t_i – relative to a predicted (virtual) position computed as if the body frame was in free fall. The situation is depicted on the diagram 5.3. Using such delta quantities, the IMU residual term of Lupton and Sukkarieh is expressed as:

$$\mathbf{r}_{\text{imu};i} = \begin{bmatrix} \text{Log}_{\text{SO(3)}}\left(\Delta\hat{\mathtt{R}}_{ij} \widetilde{\Delta\mathtt{R}}_{ij}^{\mathsf{T}}\right) \\ \Delta\hat{\mathbf{v}}_{ij} - \widetilde{\Delta\mathbf{v}}_{ij} \\ \Delta\hat{\mathbf{p}}_{ij} - \widetilde{\Delta\mathbf{p}}_{ij} \end{bmatrix} \in \mathbb{R}^9 \tag{5.15}$$

The preintegration approximation consists in assuming that the preintegrated measurements error follows a Gaussian distribution, ie. that the residual (5.15) satisfies:

$$\mathbf{r}_{\text{imu};i} \propto \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\text{imu};i}\right) \tag{5.16}$$

Under this assumption, the residual can be used to derive standard non-linear least squares problem.

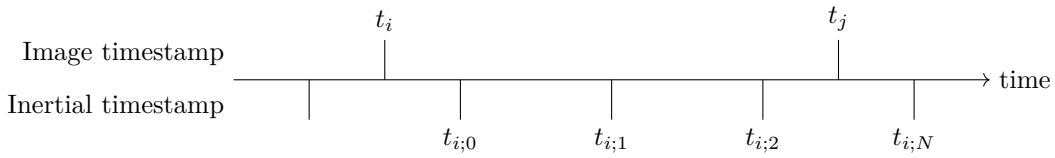
It is expected that the validity of this Gaussianity assumption will degrade with the duration of preintegration, and thus that one should not use this error terms between temporally far instants. How far is too temporally far has to be considered relative to the quality of the IMU sensor.

The integrals along with the covariance $\boldsymbol{\Sigma}_{\text{imu};i}$ are computed thanks to data sample received between t_i and t_j. From a numerical standpoint, we assume that $\boldsymbol{\omega}^b(\tau)$ – the rotational speed in body frame – and $\Delta\mathtt{R}_i^{b_i}(\tau)\mathbf{a}^b(\tau)$ – the acceleration in b_i frame – are constant between two MIMU samples.[3]

This leads to the following approximated expressions:

$$\begin{aligned}
\widetilde{\Delta\mathtt{R}}_{ij} &\simeq \mathrm{Exp}\left(\tilde{\omega}_{i;k}\Delta t_i\right).\mathrm{Exp}\left(\tilde{\omega}_{i;k+1}\Delta t_{i;k+1}\right)\ldots..\mathrm{Exp}\left(\tilde{\omega}_{i;N}\Delta t_{i;N}\right)\\
\widetilde{\Delta\mathbf{v}}_{ij} &\simeq \sum_{k\leq N} \widetilde{\Delta\mathtt{R}}_{ik}\tilde{\mathbf{a}}^b_{i;k}\Delta t_{i;k}\\
\widetilde{\Delta\mathbf{p}}_{ij} &\simeq \sum_{k\leq N} \widetilde{\Delta\mathbf{v}}_{ik} + \frac{1}{2}\widetilde{\Delta\mathtt{R}}_{ik}\tilde{\mathbf{a}}^b_{i;k}\Delta t^2_{i;k},
\end{aligned} \tag{5.17}$$

where we have indexed the time of reception of IMU data in between image the following way:



and where $\Delta t_k = t_{i:k+1} - t_{i:k}$ is the duration between two IMU samples except for $k = 0$; $\Delta t_0 = t_{i:0} - t_i$ and $k = N$; $\Delta t_N = t_{i:N} - t_j$ in order to account for boundaries.

It is possible to use a recursive algorithm to compute the three previous quantities and the covariance $\boldsymbol{\Sigma}_{\text{imu};ij}$ incrementally from the sequence of received measurements. Details of computation are found in [Forster et al., 2015]. In a gradient-based optimization context, the Jacobian of the residual with respect to the state vector can be easily computed from (5.15), (5.12)-(5.14).

Using the error term (5.15) instead of the propEKF one comes with some benefits:

- Efficiency. Compared to Section 5.1.3.2, the integration of measurements can be done only once, at the reception of the IMU data, as the integrals do not depend on the state. This is of interest, because this integration is computationally expensive if the error term summarized many measurements. While if using the propEKF strategy, we would need to recompute the propagation. From a computing architecture point of view, this preintegration can even be computed in a different thread/processing unit than the one running the full optimization process. Preintegrating the measurement does not implicitly assume having an accurate estimation of the current state in contrast to propEKF one.
- Accuracy. As already said relying on the error term of section 5.1.3.1 would require repropagation of the model, otherwise the residual would introduce large linearization error into the optimization. As the preintegrated measurements do not depend on the pose or speed estimates at any time, their expressions are valid if these estimates change, and this reduces linearization errors.

[3] Two remarks about the numerical integration strategy: first, we could use higher order measurement model for computing these integrals: by fitting polynomial function through the data before computing the integral; secondly, the assumption of $\mathbf{a}^b(\tau)$ itself being constant between two samples instead of $\Delta\mathtt{R}_i^{b_i}(\tau)\mathbf{a}^b(\tau)$ could be used. This would come at the cost of slightly more complicated computations, and was already stated in Footnote 7, Page 15

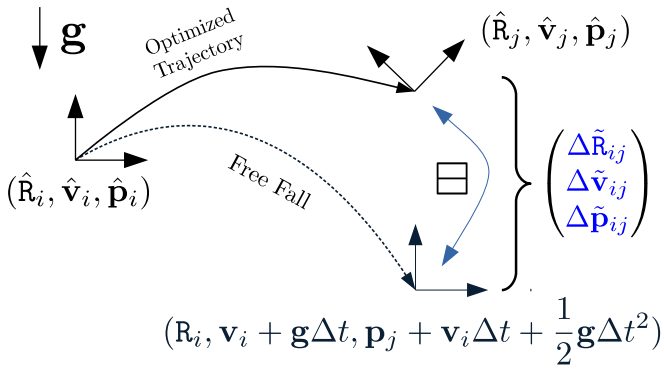


Figure 5.3: IMU residual based on preintegrated quantities of Section 5.1.3.2

A particular case where such preintegrated measurements are of interest is the algorithm initialization phases, where estimates can be far from true values. The formulation allows to still integrate and "summarize" the high frequency received data, even if the state is not totally initialized yet. This case arises typically, when the system is waiting to aggregate enough low-frequency measurement (GPS, visual measurement, etc) to render the state observable. This property was used in the original paper [Lupton and Sukkarieh, 2012] in the case of a stereo-inertial estimator initialization.

Taking the non zero biases into account

As said in the beginning of the section, the pitfall of the previous presentation is that it does not deal with the biases, yet the preintegrated measurements $\widetilde{\Delta\mathrm{R}}_{ij}$, $\widetilde{\Delta\mathbf{v}}_{ij}$ and $\widetilde{\Delta\mathbf{p}}_{ij}$ and the covariance $\boldsymbol{\Sigma}_{\mathrm{imu};i}$ actually depend on the bias estimate used to compensate the measurements as discussed in (1.3) of Chapter 1, Page 13. In the case where the biases are unknown and should be estimated along with the trajectory, Lupton and Sukkarieh suggest using a first-order perturbation technique on $\widetilde{\Delta\mathrm{R}}_{ij}$, $\widetilde{\Delta\mathbf{v}}_{ij}$ and $\widetilde{\Delta\mathbf{p}}_{ij}$ to correct them for biases estimate evolution since their integration. They also propose to ignore the dependency of $\boldsymbol{\Sigma}_{\mathrm{imu};i}$ with respect to the bias. In this case, the complete IMU residual, including bias perturbations, can be written as:

$$\mathbf{r}_{\mathrm{imu};i} = \begin{bmatrix} \mathrm{Log}\left(\Delta\mathrm{R}_{ij}\left(\widetilde{\Delta\mathrm{R}}_{ij}\mathrm{Exp}\left(\partial_{\mathbf{b}_{g_i}}\widetilde{\Delta\mathrm{R}}_{ij}(\hat{\mathbf{b}_{g_i}} - \mathbf{b}_{g_{\mathrm{int};i}})\right)\right)^{-1}\right) \\ \Delta\mathbf{v}_{ij} - \widetilde{\Delta\mathbf{v}}_{ij} + \partial_{\mathbf{b}_{g_i}}\widetilde{\Delta\mathbf{v}}_{ij}(\hat{\mathbf{b}_{g_i}} - \mathbf{b}_{g_{\mathrm{int};i}}) + \partial_{\mathbf{b}_{a_i}}\widetilde{\Delta\mathbf{v}}_{ij}(\hat{\mathbf{b}_{a_i}} - \mathbf{b}_{a\mathrm{int};i}) \\ \Delta\mathbf{p}_{ij} - \widetilde{\Delta\mathbf{p}}_{ij} + \partial_{\mathbf{b}_{g_i}}\widetilde{\Delta\mathbf{p}}_{ij}(\hat{\mathbf{b}_{g_i}} - \mathbf{b}_{g_{\mathrm{int};i}}) + \partial_{\mathbf{b}_{a_i}}\widetilde{\Delta\mathbf{p}}_{ij}(\hat{\mathbf{b}_{a_i}} - \mathbf{b}_{a\mathrm{int};i}) \end{bmatrix} \tag{5.18}$$

Where we used the shortcut notations for partial derivatives $\partial_{\mathbf{b}}\mathbf{A} = \frac{\partial\mathbf{A}}{\partial\mathbf{b}}$ to makes a Taylor expansion appears and where $\mathbf{b}_{a\mathrm{int};i}$ and $\mathbf{b}_{g_{\mathrm{int};i}}$ are the reference constant bias value used during the preintegration step. These first order approximation can in practice be assumed sufficient if the IMU is reasonably calibrated. In this case, the perturbations$(\hat{\mathbf{b}_{\times i}} - \mathbf{b}_{\times\mathrm{int};i})$ are small, and we thus leverage the gains of efficiency from preintegrated measurements while dealing with bias variations.

5.2 Addition of Magneto-inertial Constraint

We now aim at including into the cost function the constraint on the poses related to the evolution of the measured magnetic field

One can imagine at least two paths to extend the VINS cost function (5.3) with this information. The first one would be to estimate a magnetic map as a secondary variable; the second one would be to use magnetic evolution and gradient to build a constraint on subsequent frame only, similarly to how MI-DR filter of Chapter 2 works. The cost function structure in the first case is represented in Figure 5.1c while the one used in the second case is depicted on Figure 5.1d.

In the first case, assuming we can parameterize the magnetic environment with a vector of parameters, $\mathbf{m}^w_{\text{magn}}$, we could try to minimize a cost function with the following form:

$$\begin{aligned}\mathcal{E}(\boldsymbol{X}) = \sum_{(i,l)\in\mathcal{O}} \|\mathbf{r}_{\text{proj};il}(\boldsymbol{\xi}_i, \mathbf{l}_l)\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} + \sum_{i=0}^{N-1} \|\mathbf{r}_{\text{imu};i}(\boldsymbol{\xi}_i, \boldsymbol{s}_i, \boldsymbol{\xi}_{i+1}, \boldsymbol{s}_{i+1})\|^2_{\boldsymbol{\Sigma}_{\text{imu};i}} \\ + \sum_{i=0}^{N-1} \|\mathbf{r}_{\tilde{\mathbf{B}}}(\boldsymbol{\xi}_i, \mathbf{m}^w_{\text{magn}})\|^2_{\boldsymbol{\Sigma}_{\tilde{\mathbf{B}}}}\end{aligned} \tag{5.19}$$

This formulation is appealing and would be natural, specially from people used to SLAM formulation and bundle adjustment. In this formulation, $\mathbf{m}^w_{\text{magn}}$ is a map of the *world* magnetic field, and $\mathbf{r}_{\tilde{\mathbf{B}}}$ is the residual between the predicted measurement knowing the map and the magnetic field actually measured by the sensor: a problem very similar to the classical bundle adjustment one with $\mathbf{m}^w_{\text{magn}}$ playing the role of landmark positions variables.

However, we argue that this formulation is hardly practical in our case. It is not very clear how the magnetic field should be parameterized (i.e. what would be the definition of $\mathbf{m}^w_{\text{magn}}$) for odometry or large scale mapping applications. Simple parametrization as constant magnetic field or affine magnetic field would fail. One could think relying on a volumetric description of the magnetic field, but this would require a lot of memory space, and practical uses (for instance for volume occupancy map) would need to discretize the space coarsely. Such a description would be too rough to exploit the local perturbation of magnetic field that MIMU is designed to measure. Moreover, it would be wrong to assume this magnetic map is static for extended duration: for instance, any movement of metal in a room since the map creation would lead to an obsolete magnetic map with potentially large errors, that would in return corrupt the position estimate. Designing a system robust enough to deal with these kinds of long-term non-stationarities would be a challenge in our opinion.

Instead, we use measurements of the magnetic field and its gradient between two instants to extract information on the corresponding relative pose. We do not add the map of magnetic field $\mathbf{m}^w_{\text{magn}}$ into the state as secondary variable, but instead we add, for each frame, a variable representing the magnetic field at the current location: $\boldsymbol{s}_i = \left[\mathbf{B}^b_i, \mathbf{v}^w_i, \mathbf{b}_{g_i}, \mathbf{b}_{ai}\right]$

The cost function we seek to minimize would hence writes:

$$\begin{aligned}\mathcal{E}(\boldsymbol{X}) = & \sum_{(i,j)\in\mathcal{O}} \|\mathbf{r}_{\text{proj};il}(\boldsymbol{\xi}_i, \mathbf{l}_j)\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} && \text{(landmark reprojection error)} \\ & + \sum_{i=0}^{N-1} \|\mathbf{r}_{\text{mimu};i}(\boldsymbol{\xi}_i, \boldsymbol{s}_i, \boldsymbol{\xi}_{i+1}, \boldsymbol{s}_{i+1})\|^2_{\boldsymbol{\Sigma}_{\text{mimu};i}} && \text{(relative constraint on subsequent MIMU states)} \\ & + \sum_{i=0}^{N-1} \|\mathbf{r}_{\tilde{\mathbf{B}}}(\mathbf{B}_i)\|^2_{\boldsymbol{\Sigma}_{\tilde{\mathbf{B}}}} && \text{(direct magnetic variable measurement)}\end{aligned} \tag{5.20}$$

We will call this cost function the MVINS (Magneto-Visual-Inertial Navigation System) cost function. There are two differences with respect to the cost 5.3 used for VINS. First the direct measurement of the magnetic component of the state leads to the addition of a residual $\mathbf{r}_{\tilde{\mathbf{B}}}$. It writes simply as the difference between the instate magnetic field at time i and the measured field at time i:

$$\begin{aligned}\mathbb{R}^3 &\to \mathbb{R}^3 \\ \mathbf{r}_{\tilde{\mathbf{B}}} : \mathbf{B}^b_i &\mapsto \mathbf{B}^b_i - \tilde{\mathbf{B}}^b_i\end{aligned} \tag{5.21}$$

Where $\mathbf{B}^b_i$ is the current estimated value of the magnetic field at time t_i and $\tilde{\mathbf{B}}^b_i$ stems from the magnetometers reading. The covariance $\boldsymbol{\Sigma}_{\tilde{\mathbf{B}}}$ simply derives from the known characteristics of the magnetometers noise and is a tuning variable.

The second term of (5.20) is also different from the $\mathbf{r}_{\text{imu};i}$ residual of Section 5.1.3.2. It is a constraint between subsequent poses, speeds, *magnetic fields* and biases. Note that we have to express the constraint on subsequent magnetic fields in a residual term depending jointly on IMU data. Indeed, as the magnetic constraint arises from inertial measurement coupled with gradient measurement, its associated error will be correlated with the IMU constraints error between poses.

This MIMU contraint $\mathbf{r}_{\text{mimu};i}$, as we choose to call it in the following, is derived in the next section.

5.2.1 Applying Preintegrated Measurement Technique to MIMU Measurement

In a first approach to build $\mathbf{r}_{\text{mimu};i}$ residual, it would be possible to use the propEKF strategy for the MI-DR continuous model that was presented in Chapter 2, Page 23 and that we write here again:

$$\dot{\mathtt{R}}^w(t) = \mathtt{R}^w(t)[\boldsymbol{\omega}^b(t)]_\times \tag{5.22}$$

$$\dot{\mathbf{v}}^w(t) = \mathtt{R}^w(t)\mathbf{a}_s^b(t) + \mathbf{g} \tag{5.23}$$

$$\dot{\mathbf{p}}^w(t) = \mathbf{v}^w(t) \tag{5.24}$$

$$\dot{\mathbf{B}}^w(t) = \mathtt{R}^w(t)\nabla\mathbf{B}^b(t)\mathtt{R}^w(t)^\mathsf{T}\mathbf{v}^w(t) \tag{5.25}$$

$$\dot{\mathbf{b}}_g(t) = -\frac{1}{\tau_g}\mathbf{b}_g + \eta_{bg} \tag{5.26}$$

$$\dot{\mathbf{b}}_a(t) = -\frac{1}{\tau_a}\mathbf{b}_a + \eta_{ba}. \tag{5.27}$$

We show in this section that the structure of these equations of propagation can again be leveraged to define an error between two instants t_i and t_i that is statistically consistent and does not require full recomputation of the propagation each time the estimate at time t_i changes. While preintegrated quantities have been derived for the inertial part of the model, extending the approach to handle additional magnetic equations is not straightforward. In the following, we introduce three new preintegrated quantities that we call magneto-inertial preintegrated measurements.

As done with the inertial model, we assume constant and known biases to derive the expression of the residuals, and we discuss bias evolution afterwards.

Following the path of the preintegrated IMU measurement, the main idea is to integrate the magnetic field prediction equation by separating contribution of initial speed, gravity, and specific acceleration. The process is depicted graphically on Figure 5.4.

We thus pursue the integration of the continuous equation (5.25) between time t_i and time t_j.

One has:

$$\int_{t_i}^{t_j} \dot{\mathbf{B}}^w(\tau)d\tau = \int_{t_i}^{t_j} \mathtt{R}^w(\tau)\nabla\mathbf{B}^b(\tau)\mathtt{R}^{w\mathsf{T}}(\tau)\mathbf{v}^w(\tau)\mathtt{d}\tau \tag{5.28}$$

splitting $\mathbf{v}^w(\tau)$ thanks to its closed form expression (5.7) one get:

$$\mathbf{B}_j^w - \mathbf{B}_i^w = \int_{t_i}^{t_j} \mathtt{R}^w(\tau)\nabla\mathbf{B}^b(\tau)\mathtt{R}^{w\mathsf{T}}(\tau)\left[\mathbf{v}_i^w + (\tau - t_i)\mathbf{g} + \mathtt{R}_i^w\widetilde{\Delta\mathbf{v}}_i(\tau)\right]\mathtt{d}\tau \tag{5.29}$$

and factorizing by $\mathtt{R}_i$ the integral and using notation $\Delta\mathtt{R}_i(\tau)$ of (5.9):

$$\begin{aligned}\mathbf{B}_j^w - \mathbf{B}_i^w = \mathtt{R}_i^w &\int_{t_i}^{t_j} \Delta\mathtt{R}_i(\tau)\nabla\mathbf{B}^b(\tau)\Delta\mathtt{R}_i(\tau)^\mathsf{T}\mathtt{d}\tau\ \mathtt{R}_i^{w\mathsf{T}}\mathbf{v}_i^w \\ + \mathtt{R}_i^w &\int_{t_i}^{t_j} \Delta\mathtt{R}_i(\tau)\nabla\mathbf{B}^b(\tau)\Delta\mathtt{R}_i(\tau)^\mathsf{T}(\tau - t_i)\mathtt{d}\tau\ \mathtt{R}_i^{w\mathsf{T}}\mathbf{g} \\ + \mathtt{R}_i^w &\int_{t_i}^{t_j} \Delta\mathtt{R}_i(\tau)\nabla\mathbf{B}^b(\tau)\Delta\mathtt{R}_i(\tau)^\mathsf{T}\widetilde{\Delta\mathbf{v}}_i(\tau)\ \mathtt{d}\tau\end{aligned} \tag{5.30}$$

This decomposition shows three new integrals that can be computed only from measurements received from gyrometers, accelerometers and gradient of magnetic field : as a consequence, we will write them hereafter in blue and with a tilde. These integrals have a clear physical interpretation: the first one is the variation of the magnetic field associated with the uniform velocity movement of the device between t_i and t_j. The second one is the one stems from gravity effects on the movement. Both integrals combined represent the magnetic field variation that would have been perceived if the device was in free fall. Both are 3×3 symmetric matrices with zero trace. The last integral corresponds to the variation due to the integration of the specific acceleration between t_i and t_j and it is a vector of dimension 3. For convenience, we will name these integrals with the following symbols:

$$\text{(initial speed)} \qquad \widetilde{\mathbf{\Delta B}}_{\mathbf{v};ij} \overset{\text{def}}{=} \int_{t_i}^{t_j} \widetilde{\Delta \mathtt{R}}_i(\tau) \nabla \tilde{\mathbf{B}}^b(\tau) \widetilde{\Delta \mathtt{R}}_i(\tau)^\mathsf{T} \mathtt{d}\tau, \tag{5.31}$$

$$\text{(gravity)} \qquad \widetilde{\mathbf{\Delta B}}_{\mathbf{g};ij} \overset{\text{def}}{=} \int_{t_i}^{t_j} \widetilde{\Delta \mathtt{R}}_i(\tau) \nabla \tilde{\mathbf{B}}^b(\tau) \widetilde{\Delta \mathtt{R}}_i(\tau)^\mathsf{T} [\tau - t_i] \, \mathtt{d}\tau, \tag{5.32}$$

$$\text{(specific acceleration)} \qquad \widetilde{\mathbf{\Delta B}}_{\mathbf{a};ij} \overset{\text{def}}{=} \int_{t_i}^{t_j} \widetilde{\Delta \mathtt{R}}_i(\tau) \nabla \tilde{\mathbf{B}}^b(\tau) \widetilde{\Delta \mathtt{R}}_i(\tau)^\mathsf{T} \widetilde{\Delta \mathbf{v}}_i(\tau) \mathtt{d}\tau, \tag{5.33}$$

with $\widetilde{\Delta \mathtt{R}}_i(\tau)$ and $\widetilde{\Delta \mathbf{v}}_i(\tau)$ defined as in (5.9) and (5.11).

Expressing the magnetic field in the body frame one has:

$$\mathbf{B}_j^b = \Delta \mathtt{R}_{ij} \Big[\mathbf{B}_i^b + \underbrace{\underbrace{\widetilde{\mathbf{\Delta B}}_{\mathbf{v};ij} \mathtt{R}_i^{w\mathsf{T}} \mathbf{v}_i^w}_{\text{initial speed contribution}} + \underbrace{\widetilde{\mathbf{\Delta B}}_{\mathbf{g};ij} \mathtt{R}_i^{w\mathsf{T}} \mathbf{g}^w}_{\text{gravity contribution}}}_{\text{free fall contribution}} \tag{5.34}$$

$$+ \underbrace{\widetilde{\mathbf{\Delta B}}_{\mathbf{a};ij}}_{\text{specific acceleration contribution}} \Big]. \tag{5.35}$$

This decomposition separates nicely what depends on the measured input and what depends on the state we are optimizing on. We name the three new integrals *preintegrated magneto-inertial measurements*, and we exploit this structure to express an efficient magneto-inertial error term $\mathbf{r}_{\text{mimu};i}$, similarly to what was done for $\mathbf{r}_{\text{imu};i}$.

$\mathbf{r}_{\text{mimu};i}$ is defined as the following function of the state and of the preintegrated measurement:

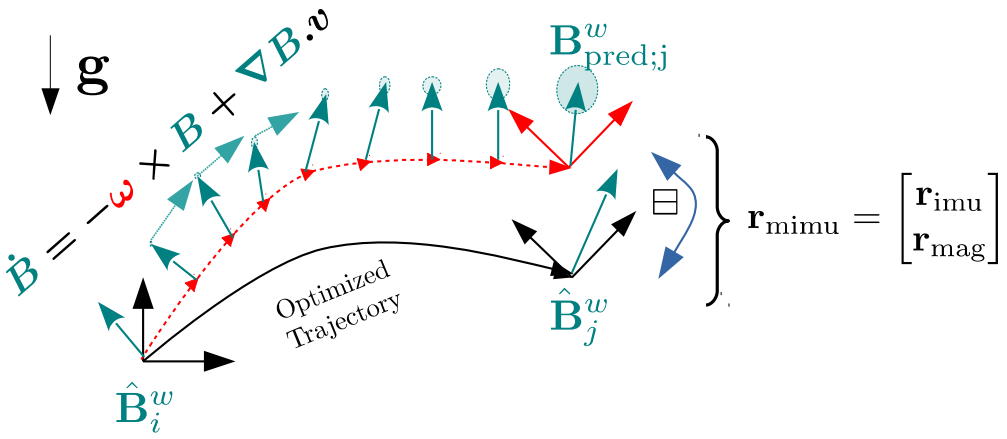
$$\mathbf{r}_{\text{mimu};i} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{r}_{\text{imu};i} \\ \mathbf{r}_{\text{mag};i} \end{bmatrix} \in \mathbb{R}^{12} \tag{5.36}$$

$$\mathbf{r}_{\text{mag};i} \overset{\text{def}}{=} \mathbf{B}_i^{b_i} - \Delta \mathtt{R}_j \left[\mathbf{B}_j^{b_j} - \widetilde{\mathbf{\Delta B}}_{\mathbf{v};ij} \mathbf{v}_i^b - \widetilde{\mathbf{\Delta B}}_{\mathbf{g};ij} \mathtt{R}_i^\mathsf{T} \mathbf{g}^w - \widetilde{\mathbf{\Delta B}}_{\mathbf{a};ij} \right]. \tag{5.37}$$

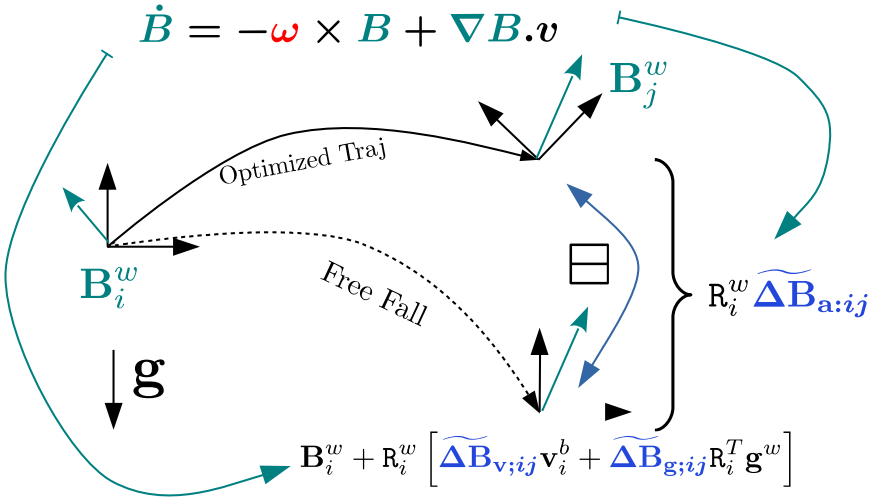
and where $\mathbf{r}_{\text{imu};i}$ is the residual (5.15).

This residual is – similarly to $\mathbf{r}_{\text{imu};i}$ – efficient to compute and so are Jacobians with respect to state parameters. Indeed, the preintegrated terms can be computed only once, at measurement reception; only the sum of the 5 terms in (5.37) needs to be computed at each iteration.

Bias estimation and preintegrated magneto-inertial measurement If inertial sensors biases are estimated online, the bias estimates can change after the computation of the preintegrated measurement. In order to cope with this, the preintegration step also computes the derivatives of the magnetic residual with respect to inertial biases, as done for IMU preintegration. During optimization, preintegrated measurements are corrected to the first order with biases evolution since the time of their computation.



(a)



(b)

Figure 5.4: (**a**) Residual computed from propEKF strategy; (**b**) Decomposition of magnetic prediction with free fall and specific acceleration terms.

5.2.1.1 Computing the covariance of the MIMU error term

The covariance $\mathbf{\Sigma}_{\text{mimu};i}$ we would like to write in (5.19) actually would depend on the state. It is, for instance, evident from the expression of $\mathbf{r}_{\text{mag};i}$ that the noise polluting the preintegrated measurements will have a stronger impact on $\mathbf{r}_{\text{mag};i}$ at high speed. This section addresses practically this issue.

Similarly to what is done for IMU preintegration, we approximate the errors in magneto-inertial preintegrated quantities by assuming that they are related to the "true ones"– that would be computed with perfect sensors – by the following relation:

$$\begin{bmatrix} \widetilde{\Delta \mathtt{R}_{ij}} \\ \widetilde{\Delta \mathbf{v}_{ij}} \\ \widetilde{\Delta \mathbf{p}_{ij}} \\ \text{Vec}\left(\widetilde{\Delta \mathbf{B}_{\mathbf{v};ij}}\right) \\ \text{Vec}\left(\widetilde{\Delta \mathbf{B}_{\mathbf{g};ij}}\right) \\ \widetilde{\Delta \mathbf{B}_{\mathbf{a};ij}} \end{bmatrix} = \begin{bmatrix} \text{Exp}_{\text{SO(3)}}\left(\epsilon_{1:3}\right) \Delta \mathtt{R}_{ij} \\ \Delta \mathbf{v}_{ij} + \epsilon_{4:6} \\ \Delta \mathbf{p}_{ij} + \epsilon_{7:9} \\ \text{Vec}\left(\Delta \mathbf{B}_{\mathbf{v};ij}\right) + \mathcal{P}_{\nabla \mathbf{B}}(\epsilon_{10:14}) \\ \text{Vec}\left(\Delta \mathbf{B}_{\mathbf{g};ij}\right) + \mathcal{P}_{\nabla \mathbf{B}}(\epsilon_{15:19}) \\ \Delta \mathbf{B}_{\mathbf{a};ij} + \epsilon_{20:23} \end{bmatrix}, \tag{5.38}$$

with ϵ a centered Gaussian vector:

$$\epsilon \propto \mathcal{N}(\mathbf{0}_{23\times 1}, \mathbf{\Sigma}_{\text{preint};ij}). \tag{5.39}$$

$\mathbf{\Sigma}_{\text{preint};ij}$ is the covariance matrix of the preintegrated measurements. This covariance is computed iteratively during the reception of the MIMU data alongside the integrals. $\mathcal{P}_{\nabla \mathbf{B}}$ is the (constant) 9×5 gradient operator defined in (2.13), Page 21. The top left corner of $\mathbf{\Sigma}_{\text{preint};ij}$ is exactly the 9×9 covariance of the preintegrated IMU measurements (without magnetometer) $\mathbf{\Sigma}_{\text{imu};i}$ that is derived in [Forster et al., 2015] (and that we do not recall in this thesis):

$$\mathbf{\Sigma}_{\text{preint};ij} = \begin{bmatrix} \mathbf{\Sigma}_{\text{imu};i} & * \\ * & * \end{bmatrix} \tag{5.40}$$

Under this approximation, the probability density of the MIMU residual (conditioned on the state) is also a Gaussian, but whose covariance depends on the value of the state, as already noted:

$$\mathbf{r}_{\text{mimu};i} \propto \mathcal{N}(\mathbf{0}_{12\times 1}, \mathbf{A}_{\mathbf{X}} \mathbf{\Sigma}_{\text{preint};ij} \mathbf{A}_{\mathbf{X}}^{\mathsf{T}}) \tag{5.41}$$

$$\mathbf{A}_{\mathbf{X}} = \begin{bmatrix} \mathbf{I}_9 & \mathbf{0}_{9\times 5} & \mathbf{0}_{9\times 5} & \mathbf{0}_{9\times 3} \\ \mathbf{0}_{3\times 9} & \mathbf{v}_i \otimes \mathbf{I}_3 \mathcal{P}_{\nabla \mathbf{B}} & \mathbf{g}^{\mathsf{T}} \mathtt{R}_i^w \otimes \mathbf{I}_3 \mathcal{P}_{\nabla \mathbf{B}} & \mathbf{I}_3 \end{bmatrix} \tag{5.42}$$

and we write the corresponding non-linear error term function:

$$\begin{bmatrix} \mathbf{r}_{\text{imu};i}^{\mathsf{T}} & \mathbf{r}_{\text{mag};i}^{\mathsf{T}} \end{bmatrix} \left(\mathbf{A}_{\mathbf{X}} \mathbf{\Sigma}_{\text{preint};ij} \mathbf{A}_{\mathbf{X}}^{\mathsf{T}}\right)^{-1} \begin{bmatrix} \mathbf{r}_{\text{imu};i} \\ \mathbf{r}_{\text{mag};i} \end{bmatrix}. \tag{5.43}$$

It yields the $\mathbf{r}_{\text{mimu};i}$ terms in the cost (5.20) with $\mathbf{\Sigma}_{\text{mimu};i} = \mathbf{A}_{\mathbf{X}} \mathbf{\Sigma}_{\text{preint};ij} \mathbf{A}_{\mathbf{X}}^{\mathsf{T}}$. The dependence of the covariance with respect to the state implies that (5.20) does not take the form of a classical nonlinear least squares – that expects known and constant covariance. In order to circumvent this, and to get a pure sum of squared residuals, we can use the "whitened" residual vector $(\mathbf{A}_{\mathbf{X}} \mathbf{\Sigma}_{\text{preint};ij} \mathbf{A}_{\mathbf{X}}^{\mathsf{T}})^{-\frac{1}{2}} \begin{bmatrix} \mathbf{r}_{\text{imu};i} \\ \mathbf{r}_{\text{mag};i} \end{bmatrix}$ whose squared value is indeed exactly equal to (5.43). Doing so has at least three practical drawbacks though:

- First, the computation of $(\mathbf{A}_{\mathbf{X}} \mathbf{\Sigma}_{\text{preint};ij} \mathbf{A}_{\mathbf{X}}^{\mathsf{T}})^{-\frac{1}{2}}$ has to occur each time the estimate changes (basically at each iteration), which is thus way less efficient than the computation needed for the IMU residual, where the covariance inverse could be computed once. Moreover, this matrix has a size of 12×12, which is not negligible for real-time with a lot of residual.

- Secondly, the correlation between the magnetic residuals and the IMU ones prevents applying a robust loss function solely to the additional constraints arising from magnetic information.
- Finally, the Jacobian of the whitened residual required for optimization are non-trivial to compute analytically. This is because the complexity of the dependence of the residual with respect to the state: a matrix inverse square root has to be computed!

The next section solves the two first issues and discusses the last one.

5.2.1.2 Robust Weighting and fast MIMU residual computation

The matrix $\mathbf{\Sigma}_{\text{mimu};i}$ of the previous section was computed assuming the magneto-inertial preintegrated measurements error ϵ was a random variable following a centered Gaussian density. This uncertainty was then propagated to the residual expression.

We stress that these computations only considered the MIMU sensor noise as sources of error and entirely disregarded errors arising in the modeling of the magnetic evolution. Indeed, the assumptions of non-curved or non-stationary magnetic field are in practice another source of error, that is not modeled. Such errors lead to outliers in the magnetic prediction (in the same way outliers affect visual predictions), that can be tackled using a robust loss function. On the other hand, IMU errors are correctly modeled as Gaussian and the corresponding residuals should not be down-weighted by the robust estimation process. In this section, we rework a bit the expression on the MIMU residual to allow the use of a robust loss function on the magnetic part of the MIMU residual, without affecting the pure IMU residual.

The idea is to leverage the Schur complement on the matrix $(\mathbf{A}_{\mathbf{X}}\mathbf{\Sigma}_{\text{preint};ij}\mathbf{A}_{\mathbf{X}}^{\mathsf{T}})$ in the equation (5.43), in order to split the error term in two parts (see Appendix A, Page 169):

$$\mathbf{r}_{\text{imu};i}^{\mathsf{T}}\mathbf{\Sigma}_{\text{imu};i}^{-1}\mathbf{r}_{\text{imu};i} + (\mathbf{r}_{\text{mag};i} - \mathbf{B}_{\mathbf{X}}\mathbf{r}_{\text{imu};i})^{\mathsf{T}}\mathbf{C}_{\mathbf{X}}^{-1}(\mathbf{r}_{\text{mag};i} - \mathbf{B}_{\mathbf{X}}\mathbf{r}_{\text{imu};i}) \tag{5.44}$$

Where $\mathbf{B}_{\mathbf{X}} \in \mathbb{R}^{3\times 9}$ and $\mathbf{C}_{\mathbf{X}} \in \mathbb{R}^{3\times 3}$ are matrices that depends on the state whose expressions are not given here but can be retrieved easily by applying the Schur complement technique.

Doing so reveals the fact – hidden until now – that the MIMU total cost function can actually be expressed as the sum of two squared terms, one corresponding to $\mathbf{r}_{\text{imu};i}$ of Section 5.1.3.2 and an additional one. Because of the correlation between $\mathbf{r}_{\text{mag};i}$ and $\mathbf{r}_{\text{imu};i}$ this additional one is not directly a weighted version of $\mathbf{r}_{\text{mag};i}$, but a combination of $\mathbf{r}_{\text{mag};i}$ and $\mathbf{r}_{\text{imu};i}$. If the form of the second term of (5.44) can be seen as a least squares term with covariance $\mathbf{C}_{\mathbf{X}}$ – similarly to the full residual of the last section – this covariance still depends explicitly on the estimate, which is not the canonical least squares cost function. We can however, once again, whiten the residual and use $\mathbf{C}_{\mathbf{X}}^{-\frac{1}{2}}(\mathbf{r}_{\text{mag};i} - \mathbf{B}_{\mathbf{X}}\mathbf{r}_{\text{imu};i})$ as a non-linear residual with Identity covariance.

This decomposition in two additive terms suits the initial motivation of applying the robust loss function only to the additional constraint that magnetometers involve; we are now able to robustify the cost function (5.44) the following way:

$$\|\mathbf{r}_{\text{imu};i}\|^2_{\mathbf{\Sigma}_{\text{imu};i}} + \rho\left(\mathbf{C}_{\mathbf{X}}^{-\frac{1}{2}}(\mathbf{r}_{\text{mag};i} - \mathbf{B}_{\mathbf{X}}\mathbf{r}_{\text{imu};i})\right) + \text{cst} \qquad \text{with } \rho \text{ a robust norm}$$

Which, Moreover, as $\mathbf{C}_{\mathbf{X}}$ is a small 3×3 matrix, this computation is way more efficient than the one of $(\mathbf{A}_{\mathbf{X}}\mathbf{\Sigma}_{\text{preint};ij}\mathbf{A}_{\mathbf{X}}^{\mathsf{T}})^{-\frac{1}{2}}\begin{bmatrix}\mathbf{r}_{\text{imu};i}\\ \mathbf{r}_{\text{mag};i}\end{bmatrix}$ raised in previous section.

Unfortunately, the Jacobian matrix of this residual is still not straightforward to compute, because of the $\mathbf{C}_{\mathbf{X}}^{-\frac{1}{2}}$ term. We thought about different alternatives to compute it:

1. Use the residual $\mathbf{C}_{\mathbf{X}}^{-\frac{1}{2}}(\mathbf{r}_{\text{mag};i} - \mathbf{B}_{\mathbf{X}}\mathbf{r}_{\text{imu};i})$ as a non-linear error term and compute its Jacobian with respect to the state by leveraging a code differentiation library; thus avoiding painful analytical chain rule through the square root inverse operation.

2. Recompute at each iteration the weighting matrix $\mathbf{C}_\mathbf{X}$ in an IRLS fashion along with the weight implied by the use of a robust loss function – and thus ignore its dependence with respect the state when computing the Jacobian. This approximation could slow down convergence though.

3. Use the residual $(\mathbf{r}_{\text{mag};i} - \mathbf{B}_\mathbf{X}\mathbf{r}_{\text{imu};i})$ and compute the matrix $\mathbf{C}_\mathbf{X}^{-\frac{1}{2}}$ once, at the first estimate of the body speed and attitude. This method is the cheapest one but would introduce significant errors during initialization.

In contrast to the algorithm presented in [Caruso et al., 2017b] where the third solution was implemented, we use the first, more exact, solution in the rest of this chapter.

5.3 Gradient-based Optimization on Manifold

This section recalls how to do gradient based optimization with a state that belongs to a manifold. It essentially borrows notions presented for instance in a robotic context in [Wagner et al., 2011] and [Hertzberg et al., 2013].

5.3.1 State Manifold and Local parametrization

The state $\boldsymbol{X}$ we optimize on belongs to a compound manifold $\mathcal{X}$ that is not a vector space because of the rotations variables $\mathtt{R}^w$. We use the local tangent space of the manifold to express a perturbation around a state, and we write $\delta\boldsymbol{X}(\hat{\boldsymbol{X}})$ the element of the tangent plane at the current linearization point $\hat{\boldsymbol{X}}$. This element is decomposed as:

$$\delta\boldsymbol{X}(\hat{\boldsymbol{X}}) = [\underbrace{\delta\boldsymbol{\xi}_1 \cdots \delta\boldsymbol{\xi}_N}_{\text{frame pose}}, \underbrace{\delta\boldsymbol{s}_1 \cdots \delta\boldsymbol{s}_N}_{\text{mimu state}}, \underbrace{\delta\mathbf{l}_{p_1} \cdots \delta\mathbf{l}_{p_M}}_{\text{landmarks state}}]^\mathsf{T} \tag{5.45}$$

For the optimization over $\mathcal{X}$, we need to define a *retraction operator*. This operator expresses the link between a perturbation in tangent space and the perturbed states on the manifold. We use the $\boxplus$ operator symbol for the retraction operator, so that $\mathbf{X} \boxplus \delta\boldsymbol{X}(\mathbf{X})$ computes a new state in $\mathcal{X}$ from a perturbation around some previous state $\mathbf{X}$.

In this chapter, we define a retraction operator as regular addition operation for all components of the state except for keyframe pose states where we use:

$$\boldsymbol{\xi} \boxplus \delta\boldsymbol{\xi} = [\mathbf{p} + \delta\boldsymbol{\xi}_{1:3}, \mathtt{R}\mathrm{Exp}\left(\delta\boldsymbol{\xi}_{4:6}\right)] \tag{5.46}$$

We define the inverse operator $\boxminus$ as the binary operator giving the perturbation element between two elements of $\mathcal{X}$. It is defined as regular minus operation except for keyframe pose states for which it is:

$$\delta\boldsymbol{\xi}_{12}(\boldsymbol{\xi}_1) = \boldsymbol{\xi}_2 \boxminus \boldsymbol{\xi}_1 = \left[\mathbf{p}_2 - \mathbf{p}_1, \mathrm{Log}(\mathtt{R}_1^{-1}\mathtt{R}_2)\right] \tag{5.47}$$

This operator is of importance in a gradient descent based method as it defines how the Jacobian should be computed and how the state should be updated.

5.3.2 Levenberg-Marquard Algorithm on Manifold

In this chapter, we minimize the cost $\mathcal{E}(\boldsymbol{X})$ iteratively through a Levenberg-Marquardt algorithm adapted to optimization on the state manifold using the retraction operator of previous sections. We note $\hat{\mathbf{X}}$ the current estimate and $\mathbf{r}\left(\hat{\mathbf{X}} \boxplus \delta\mathbf{X}\right)$ the concatenation of all residual error terms, already weighted by their covariance.

LM is a Gauss-Newton descent algorithm which relies on a approximation of the norm of the residual vector for a perturbation along the linearization point tangent space. This approximation writes:

$$\|\mathbf{r}\left(\hat{\mathbf{X}} \boxplus \delta\mathbf{X}\right)\|^2 \simeq \|(\mathbf{r}(\hat{\mathbf{X}}) + \mathtt{J}_{\mathbf{r}}|_{\hat{\mathbf{X}}}\delta\mathbf{X}\|^2, \tag{5.48}$$

where $\mathtt{J}_{\mathbf{r}}|_{\hat{\mathbf{X}}}$ is the Jacobian of the residual function around $\hat{\mathbf{X}}$. A damped minimizer of (5.48) is found by inverting the modified normal equation:

$$(\mathtt{J}_{\mathbf{r}}|_{\hat{\mathbf{X}}}^{\mathsf{T}}\mathtt{J}_{\mathbf{r}}|_{\hat{\mathbf{X}}} + \lambda\mathbf{D})\delta\mathbf{X} = \mathtt{J}_{\mathbf{r}}|_{\hat{\mathbf{X}}}^{\mathsf{T}}\mathbf{r}(\hat{\mathbf{X}}) \tag{5.49}$$

$\lambda \in \mathbb{R}^+$ is called the damping factor of the Levenberg-Marquardt method and $\mathbf{D}$ is a diagonal matrix – we choose the diagonal of $\mathtt{J}_{\mathbf{r}}|_{\hat{\mathbf{X}}}^{\mathsf{T}}\mathtt{J}_{\mathbf{r}}|_{\hat{\mathbf{X}}}$ matrix, which is a classical choice in bundle adjustment. After each iteration, the increment $\delta\hat{\mathbf{X}}$ is applied with the following update rule:

$$\hat{\mathbf{X}} \leftarrow \hat{\mathbf{X}} \boxplus \delta\mathbf{X}. \tag{5.50}$$

In the Levenberg-Marquardt scheme, the parameter lambda is not constant from one iteration to another; its absolute value is increased if the last iteration actually reduced the error, and decreased otherwise.

Note that, in practice, we will take into account robust cost function through an IRLS scheme as described in Section 3.4.3, Page 39.

5.4 Testing the MIMU Preintegrated Residual

We first tested if the minimization of the cost function (5.20) without any visual observation was able to reconstruct a trajectory effectively. To do so, we built an optimization problem with only the magneto-inertial part of the cost function. More precisely, on a data sequence from a real MIMU system, we selected keyframes regularly, and we preintegrated magneto-inertial measurements in between these keyframes. Then we built the error term between MIMU and pose states at the chosen keyframes. The resulting cost function is optimized incrementally with the algorithm of Section 5.3.2.

In order to initialize the first poses and biases variables correctly, we use for the first 10 seconds high-frequency keyframes and initialize the very first attitude estimate from accelerometer reading, assuming the device is at rest. The subsequent variables are initialized from magneto-inertial propagation from previous estimates.

This test has two goals:

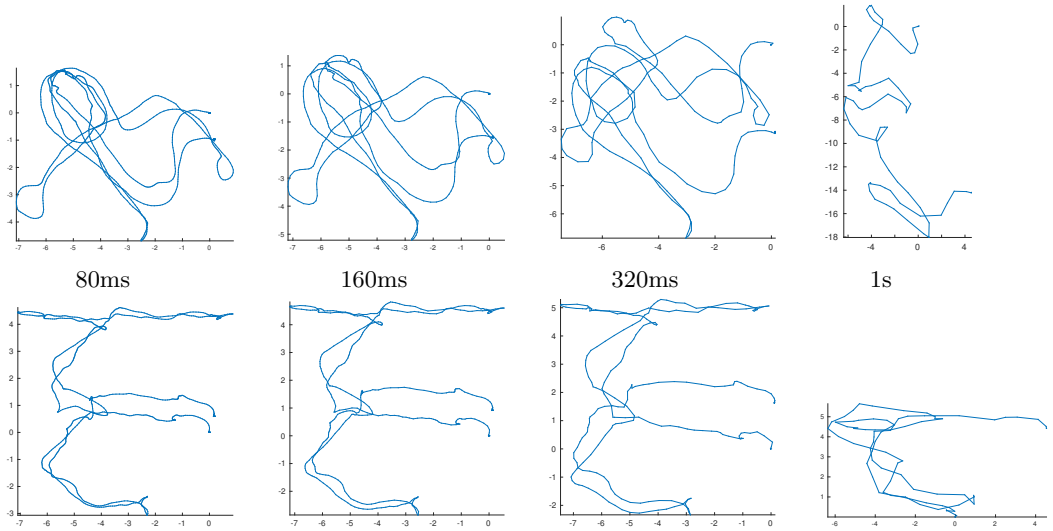
- Unitary test. Verify that the implementation of the MIMU residual was correct.
- Proof of concept. Verify that we could build a meaningful magnetic prediction over duration large enough with respect to typical inter-keyframe duration in a SLAM or bundle adjustment system – roughly one second.

The second goal is actually important: typically the duration between two keyframes is two or three orders of magnitude higher than the MIMU sample period. The MI-DR filter of Chapter 2, even if based on the same equation prediction, predicts and corrects the magnetic field state at high frequency (325 Hz). Such a frequency is not compatible with optimization based inference, in which we only predict states, including magnetic field state, at frame or keyframe rate.

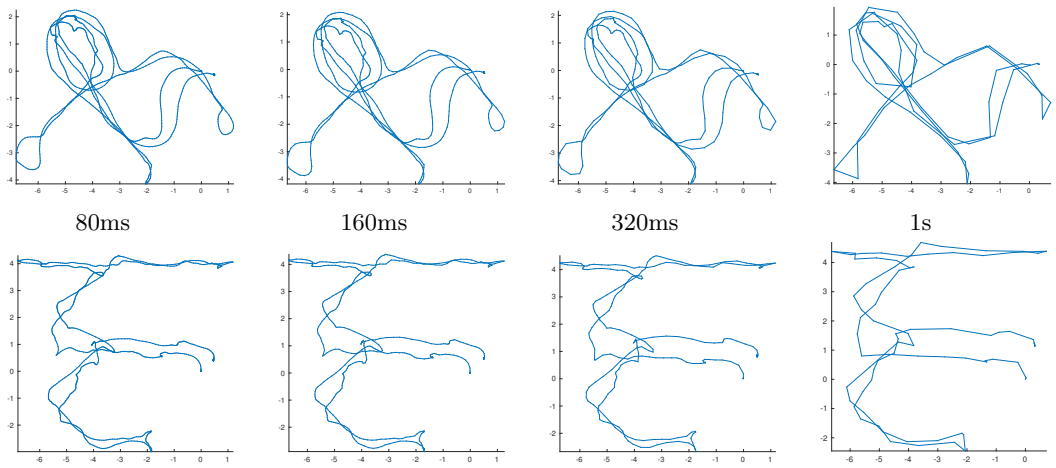
The Figure 5.5 shows a trajectory reconstruction experiment, using only the magnetic and inertial residual, with varying durations between keyframes. We compare our preintegrated measurement residual with a residual obtained by merely assuming that the magnetic field *gradient* is constant in the inertial frame during the entire integration duration – estimating its value alongside the state. Top graphics depict results of the "constant gradient" residual, while bottom ones depict results obtained with our preintegrated strategy. From the results, we draw the following conclusion:

- Both residuals allow building a consistent trajectory when used over a small duration. Indeed, we observe that the three floors of the house are well reconstructed for a duration between keyframes of 80ms, and the trajectory of different floors superimpose roughly.

- The preintegrated magneto-inertial measurements allow reconstructing a trajectory even for moderate durations between keyframe. Indeed, if the bottom plot of Figure 5.5 shows trajectories becoming piecewise linear when reducing the number of keyframes, the global shape of trajectory stays the same. This is in contrast with what we observe on the top graphics: with residuals derived under a constant gradient assumption, the estimated trajectory degrades quickly with longer durations between keyframes.



(a)



(b)

Figure 5.5: Magnetic prediction experiment. A Trajectory across three floors of a house is reconstructed by a batch optimization with only magneto-inertial error terms. We test for different duration of preintegration between pose states (in subcaption). Top graphics assume a uniform magnetic gradient over all the duration of integration, bottom ones stem from the use of the preintegrated residuals $\mathbf{r}_{\text{mimu};i}$ and $\mathbf{r}_{\text{mag};i}$ defined in Section 5.2.1. With the constant gradient assumption the quality of error terms degrades quickly with increasing integration duration, contrarily to the presented approach.

5.5 Application: a Sliding Window Smoother

Solving the full bundle adjustment problem is generally of interest for offline reconstruction, but does not fit real-time purposes, because it is too computationally expensive to solve for large problems. Practical algorithms rely on either conditioning on or marginalization of past states.

Based on the full MVINS cost function (5.20), Page 78, we implemented a sliding-window smoother compatible with real-time, that relies on marginalization of landmark, past pose and past MIMU states.

5.5.1 Algorithm Overview

The cost function is maintained internally using the factor graph formalism through the use of the GTSAM[4] library.

The next sections will detail the algorithm steps. A general outline is given hereafter, and a diagram of the algorithm is depicted on Figure 5.6. When an image is acquired:

1. Corner features that are already present in the previous frame are tracked by optical flow, and – if the new frame is a keyframe – new corners are detected, using a visual corner detection algorithm. In parallel, we preintegrate all magneto-inertial measurement having a timestamp between the last and the current image instant to form a preintegrated error term between the two frames.
2. A selection of new landmarks are triangulated from previous and new observations.
3. New variables are added to the cost function: current pose state, MIMU state, and newly activated landmarks. Error terms are also added: preintegrated constraints, new observations of already activated landmarks, observations of newly activated landmarks.
4. We select the variables to marginalize and drop some errors terms. Marginalization is pursued to create a new prior error term.
5. The new state is found minimizing the cost function.
6. Finally, the next image is waited for.

5.5.2 Marginalization of States

In order to bound the computational time, we forget past states and modify the cost function accordingly through marginalization. Next section explains the transformation of the cost that is involved by the marginalization process, while the Section 5.5.2.2 describes the marginalization scheme we use, I.e., how we select the variables to be marginalized at each time step.

5.5.2.1 Marginalization of Variables in a SLAM problem

Assume we want to remove the variables $\mathbf{X}_{\text{k;marg}}$ from the cost function while keeping the variables $\mathbf{X}_{\text{k;keep}}$. We can partition the total cost function in two terms as:

$$\mathcal{E}(\mathbf{X}_k) = \mathcal{E}_{\text{marg}}(\mathbf{X}_{\text{k;keep}}, \mathbf{X}_{\text{k;marg}}) + \mathcal{E}_{\text{keep}}(\mathbf{X}_{\text{k;keep}}). \tag{5.51}$$

$\mathcal{E}_{\text{marg}}(\mathbf{X}_{\text{k;keep}}, \mathbf{X}_{\text{k;marg}})$ contains squared residuals implying the state to be marginalized, while $\mathcal{E}_{\text{keep}}(\mathbf{X}_{\text{k;keep}})$ contains all other residuals. The marginalization of $\mathbf{X}_{\text{k;marg}}$ is based on the Gauss-Newton approximation of the cost function. Without any loss of generality[5] we can write the approximation the following way:

$$\mathcal{E}_{\text{marg}}(\mathbf{X}_k \boxplus \delta\boldsymbol{X}) = \|\mathbf{r}_{\text{marg}}(\mathbf{X}_k) + \mathbf{J}\delta\boldsymbol{X}\|^2_{\boldsymbol{\Sigma}_{\text{marg}}} \tag{5.52}$$

[4] https://bitbucket.org/gtborg/gtsam

[5] Note that this is indeed also true if a robust loss function is used for some residual. In this case $\boldsymbol{\Sigma}_{\text{marg}}$ will reflect the weighting factor induced by IRLS technique.

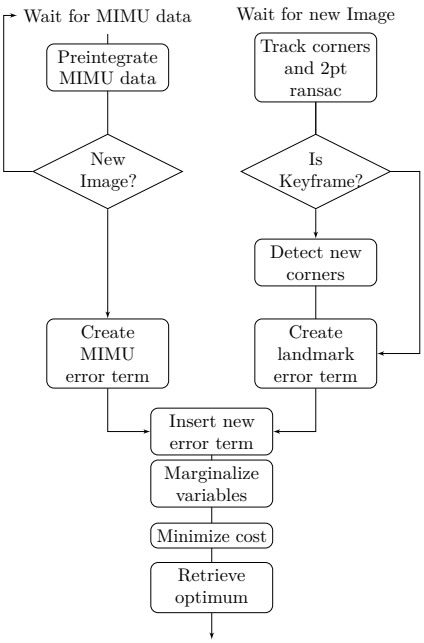


Figure 5.6: Diagram of the pipeline of the sliding window estimator as implemented with GTSAM library.

This cost function can be separated in two residuals with the help of a QR decomposition of $\boldsymbol{\Sigma}_{\text{marg}}^{-\frac{1}{2}}\mathbf{J}$:

$$\mathcal{E}_{\text{marg}}(\mathbf{X}_k \boxplus \delta\boldsymbol{X}) = \|\boldsymbol{\Sigma}_{\text{marg}}^{\frac{1}{2}}\mathbf{r}_{\text{marg}}(\mathbf{X}_k) + \boldsymbol{\Sigma}_{\text{marg}}^{\frac{1}{2}}\mathbf{J}\delta\boldsymbol{X}\|^2 \tag{5.53}$$

(Doing the QR decomposition) (5.54)

$$= \left\| \boldsymbol{\Sigma}_{\text{marg}}^{-\frac{1}{2}}\mathbf{r}_{\text{marg}}(\mathbf{X}_k) + \begin{bmatrix}\mathbf{O}_1 & \mathbf{O}_2\end{bmatrix} \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_{12} \\ 0 & \mathbf{R}_2\end{bmatrix} \begin{bmatrix}\delta\boldsymbol{X}_{\text{k;marg}} \\ \delta\boldsymbol{X}_{\text{k;keep}}\end{bmatrix} \right\|^2 \tag{5.55}$$

(Using L_2 norm invariance to orthogonal application) (5.56)

$$= \left\| \begin{bmatrix}\mathbf{O}_1^\mathsf{T} \\ \mathbf{O}_2^\mathsf{T}\end{bmatrix} \boldsymbol{\Sigma}_{\text{marg}}^{-\frac{1}{2}}\mathbf{r}_{\text{marg}}(\mathbf{X}_k) + \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_{12} \\ 0 & \mathbf{R}_2\end{bmatrix} \begin{bmatrix}\delta\boldsymbol{X}_{\text{k;marg}} \\ \delta\boldsymbol{X}_{\text{k;keep}}\end{bmatrix} \right\|^2 \tag{5.57}$$

$$\begin{aligned} &= \left\| \mathbf{O}_1^\mathsf{T}\boldsymbol{\Sigma}_{\text{marg}}^{-\frac{1}{2}}\mathbf{r}_{\text{marg}}(\mathbf{X}_k) + \mathbf{R}_1\delta\boldsymbol{X}_{\text{k;marg}} + \mathbf{R}_{12}\delta\boldsymbol{X}_{\text{k;keep}} \right\|^2 \\ &\quad + \left\| \mathbf{O}_2^\mathsf{T}\boldsymbol{\Sigma}_{\text{marg}}^{-\frac{1}{2}}\mathbf{r}_{\text{marg}}(\mathbf{X}_k) + \mathbf{R}_2\delta\boldsymbol{X}_{\text{k;keep}} \right\|^2 \end{aligned} \tag{5.58}$$

The first part of (5.58) is the only term in the linearized total cost $\mathcal{E}$ that depends on $\delta\boldsymbol{X}_{\text{k;marg}}$. Assuming that $\mathbf{R}_1$ is invertible, this can always be set to zero by choosing $\delta\boldsymbol{X}_{\text{k;marg}}$ accordingly. Hence, we can just approximate the cost function $\mathcal{E}_{\text{marg}}(\mathbf{X}_k \boxplus \delta\boldsymbol{X})$ with:

$$\mathcal{E}_{\text{marg}}(\mathbf{X}_k \boxplus \delta\boldsymbol{X}) \simeq \|\mathbf{O}_2^\mathsf{T}\boldsymbol{\Sigma}_{\text{marg}}^{\frac{1}{2}}\mathbf{r}_{\text{marg}}(\mathbf{X}_k) + \mathbf{R}_2\delta\boldsymbol{X}_{\text{k;keep}}\|^2. \tag{5.59}$$

We will call this new residual $\mathbf{r}_{\text{prior;k}}$. Marginalizing the variables $\mathbf{X}_{\text{k,marg}}$ is exactly replacing $\mathcal{E}_{\text{marg}}(\mathbf{X}_k \boxplus \delta\boldsymbol{X})$ by $\|\mathbf{r}_{\text{prior;k}}\|^2$ in the cost function for all subsequent time steps.[6]

It is important to note the consequence of such a transformation of the global cost: by using this marginalization technique, we actually fix the linearization points for the marginalized variables and

[6] This process is simply called "reduction" by the authors of [Triggs et al., 2000]. Note that this is indeed exact marginalization – in probability theoretical sense – if error distributions are Gaussian and residual linear.

give up the possibility to relinearize associated residuals. Notably, this means that the presented algorithm will accumulate linearization errors exactly as an EKF does. Compared to the EKF however, one possible benefit could be to use the full non-linear cost function and do several solver iterations before marginalization to get a better linearization point.

Besides, one important and deep drawback of marginalizing is that long loop-closure cannot be handled consistently easily, as variables are removed from the state a soon as they leave the sliding window. We stress here that marginalization is a solution to a purely computational cost problem: from the point-of-view of the quality of the estimation, it would always have been better to optimize the full cost function.

5.5.2.2 Marginalization Strategy: Double Sliding Windows

It is known that some marginalization choices break the sparsity of the SLAM cost function and impacts negatively the performance. [Sibley et al., 2010] shows that Hessian fill-in occurs when the prior residual affects the landmark geometry in the SLAM problem. To avoid this, we adopt here the marginalization strategy of [Leutenegger et al., 2015]. This strategy drops, before marginalization, reprojection error terms associated either with non-keyframe or to landmarks that are still visible in some recent keyframes to avoid a joint prior on keyframe and landmarks. In particular, we retain the "double window" aspect of their strategy: we keep a reduced set of very recent MIMU and pose states and a set of older and more temporally separated keyframe pose states. The first set involves mainly short-term MIMU prediction through $\mathbf{r}_{\text{mimu};i}$ between each frame, while the second set involves medium-term information carried from landmark observations only.

The cost to be optimized at time step k is then of the form:

$$\begin{aligned}\mathcal{E}(\boldsymbol{X}_k) = &\sum_{(i,j)\in\mathcal{O}_k} \|\mathbf{r}_{\text{proj};il}(\boldsymbol{\xi}_i, \mathbf{l}_j)\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} + \sum_{i\in\mathcal{S}_k} \|\mathbf{r}_{\text{meas}}(\mathbf{B}_i)\|^2_{\boldsymbol{\Sigma}_{\text{mag}}} \\ &+ \sum_{i\in\mathcal{S}_k} \|\mathbf{r}_{\text{mimu}}(\boldsymbol{\xi}_i, \boldsymbol{s}_i, \boldsymbol{\xi}_{i+1}, \boldsymbol{s}_{i+1})\|^2_{\boldsymbol{\Sigma}_{\text{imu};i}} + \|\mathbf{r}_{\text{prior;k}}(\{\boldsymbol{\xi}_i\}_{i\in\Upsilon_k}, \boldsymbol{\xi}_{k-|\mathcal{S}_k|}, \boldsymbol{s}_{k-|\mathcal{S}_k|})\|^2. \quad (5.60)\end{aligned}$$

where $\mathcal{O}_k$ is the set of pairs time step/landmark leading to reprojection error still in the cost function at time k, $\mathcal{S}_k$ is the set of time steps for which the pose and MIMU states are still in the visual-magneto-inertial sliding window at time k and Υ_k the set of time steps for which the pose variables are still in the visual window but whose MIMU tates have been already discarded. A diagram summarizing the different time windows and variables of the problem is presented in Figure 5.7.

Care should also be taken to avoid inconsistencies related to the use of two different linearization points for the same variable (as in the EKF approach of [Li and Mourikis, 2013] or [Engel et al., 2018]). This happens if the estimate of some variable changes after having been included in the prior – because the prior still carry information linearized around the estimate at the time of marginalization. We use here a fixed Jacobian approach, which definitively fixes the linearization point of all variables already included in the prior, which is equivalent to the modified Levenberg-Marquardt algorithm described in the next section.

5.5.3 Handling the Linearization Point of the Prior Term within a Levenberg-Marquardt Algorithm

As already said, maintaining a prior linear with respect to some component of the state amounts to choose a fixed linearization point and to restrict the estimation to the tangent space around that component at the time of marginalization. In order to avoid introducing two different linearization points and tangent spaces for one variable, the Levenberg-Marquardt algorithm is modified in a way that all variables are not linearized around the most recent estimate in the Jacobian computation, but sometimes around an earlier estimate.

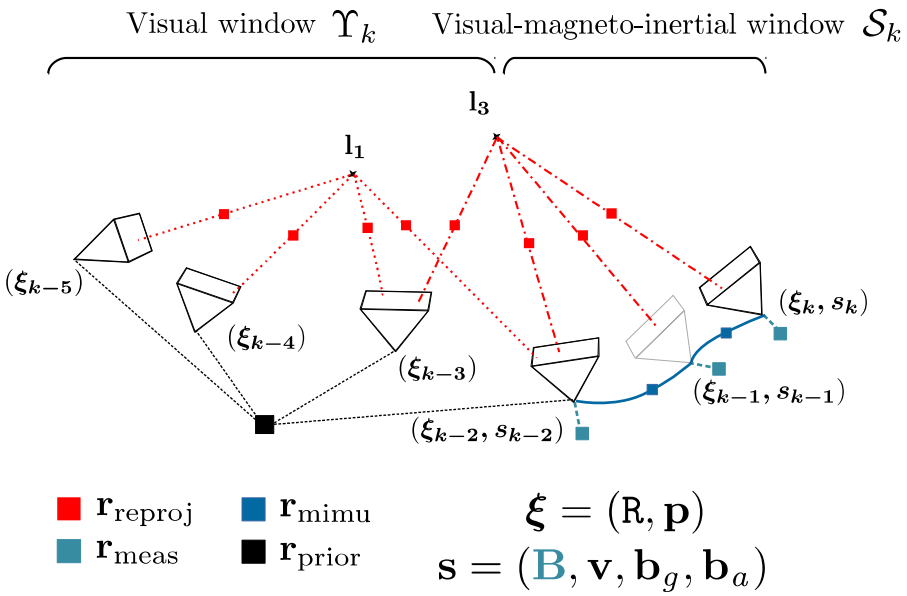


Figure 5.7: State and cost function minimized at each time step by the sliding window smoother.

This trick is popular in the SLAM community, as a workaround to numerical issues and some problems of false observability. It is for instance used by [Engel et al., 2018] or [Leutenegger et al., 2015].[7]

Following these references, we define the current estimate by a perturbation $\boldsymbol{\Delta}\mathbf{X}$ of the linearization point along the tangent space: $\mathbf{X}_{\text{lin}} \boxplus \boldsymbol{\Delta}\mathbf{X}$. We seek at each iteration to minimize a first order approximation of the norm of the residual vector with respect to a perturbation along the linearization point tangent space:

$$\|\mathbf{r}\left(\mathbf{X}_{\text{lin}} \boxplus (\boldsymbol{\Delta}\mathbf{X} + \delta\mathbf{X})\right)\|^2 \simeq \|(\mathbf{r}(\mathbf{X}_{\text{lin}} \boxplus \boldsymbol{\Delta}\mathbf{X}) + \mathtt{J}_{\mathbf{r}}|_{\mathbf{X}_{\text{lin}}} \delta\mathbf{X}\|^2. \tag{5.61}$$

Where $\mathtt{J}_{\mathbf{r}}|_{\mathbf{X}_{\text{lin}}}$ is indeed the Jacobian of the residual function around $\mathbf{X}_{\text{lin}}$ (and not around $\mathbf{X}_{\text{lin}} \boxplus \boldsymbol{\Delta}\mathbf{X}$). A damped solution is found by inverting:

$$(\mathtt{J}_{\mathbf{r}}|_{\mathbf{X}_{\text{lin}}}^{\mathsf{T}} \mathtt{J}_{\mathbf{r}}|_{\mathbf{X}_{\text{lin}}} + \lambda\mathbf{D})\delta\mathbf{X} = \mathtt{J}_{\mathbf{r}}|_{\mathbf{X}_{\text{lin}}}^{\mathsf{T}} (\mathbf{r}(\mathbf{X}_{\text{lin}} \boxplus \boldsymbol{\Delta}\mathbf{X})) \tag{5.62}$$

With $\lambda \in \mathbb{R}^+$ the damping factor of the Levenberg-Marquardt method and $\mathbf{D}$ a diagonal matrix; we choose the diagonal of $\mathtt{J}_{\mathbf{r}}|_{\mathbf{X}_{\text{lin}}}^{\mathsf{T}} \mathtt{J}_{\mathbf{r}}|_{\mathbf{X}_{\text{lin}}}$ matrix. After each iteration, the increment $\delta\mathbf{X}$ is applied with the following update rule:

$$\boldsymbol{\Delta}\mathbf{X} \leftarrow \boldsymbol{\Delta}\mathbf{X} + \delta\mathbf{X}. \tag{5.63}$$

At the end of each iteration, but only for states for which the linearization point has not been fixed yet, we update the linearization point through:

$$\mathbf{X}_{\text{lin}} \leftarrow \mathbf{X}_{\text{lin}} \boxplus \boldsymbol{\Delta}\mathbf{X}. \tag{5.64}$$

This trick is, all-in-all, a small variation of the Levenberg-Marquardt algorithm that is easy to implement.

[7] According to the authors of [Engel et al., 2018] this strategy for fixing the linearization point has strong beneficial effects on quality of the estimate, specially for very long trajectories. We have not evaluated this point in the present work.

5.5.4 System Initialization

The cost function we optimize is not convex. The iterative solver thus requires a good initial estimate of the state. As we build and solve the cost function incrementally, estimates for new poses, speed, and magnetic field can be deduced from MIMU propagation and previous estimates, and estimate for new landmark can be computed from triangulation.

For the very first estimate, this is not possible though, as we do not have any starting point to integrate or poses to triangulate from. Instead, we use the MI-DR filter of [Chesneau et al., 2016] to initialize a first set of keyframes poses and MIMU estimates. From these known poses, we triangulate a first set of landmarks to initialize a local map and start the non-linear optimization.[8] This method, admittedly, only provides a useful initialization in the working domain of the magneto-inertial dead reckoning system. However, contrarily to other commonly used methods, it does not assume that the device is at rest during initialization. Works describing alternatives methods for initialization were given Section 4.7.

5.5.5 Gauge Fixing

Four degrees of freedom are known to be unobservable: the heading and the global position. If not handled carefully, the (Gauss-Newton approximation of) Hessian of the optimization process will be rank deficient and numerical errors will arise. We fix the gauge by setting an error term on the first keyframe translation and heading. This indeed fixes the general absolute orientation around gravity and absolute position that are otherwise unobservable and avoid optimization failures.

5.5.6 Features Tracking and Keyframe Selection

We intentionally use a simple method for corner association from image to images. Bucketed detection of Harris corner is done in each keyframe; corners are then tracked with OpenCV[9] pyramidal KLT algorithm in all subsequent frames. Tracking is done between successive frame: the template is reinitialized at each frame; thus one can restrain to an efficient translational KLT formulation which assuming small movement between successive frames. The tracking keeps going on until either: (i) the feature goes out of scope; (ii) the feature tracking failed; (iii) the feature is classified as an outlier.

This is also well known that corner tracked by translational KLT with reinitialization of templates will drift on image space. A common strategy is to register regularly the currently tracked patch with the one at the time of corner first detection. The obtained registration error magnitude is used to assess the drift: if larger than a threshold the features are classified as outliers. This registration is generally done through an affine transform of the patch [Bouguet, 2000], which is quite expensive to compute. In our implementation, we keep the idea of monitoring regularly point appearance change with respect to the first detection but, instead of affine registration, we compare ORB descriptor [Rublee et al., 2011] between first corner detection and current features location in image space. This descriptor is designed to be roughly invariant to point-of-view changes and is a cheap alternative to affine registration. In practice, we observed that this strategy is particularly useful to remove early points lying on the border of occluding contours.

For grossly non-static features rejection, we also run a 2-point RANSAC algorithm ([Bazin et al., 2010],[Troiani et al., 2014]) from the last keyframe to the current frame, using the relative orientation from the unbiased integrated gyroscope, corrected with the current bias estimate. We do it before descriptor computation. These two outlier rejection tests render our features tracking strategy

[8] The MI-DR filter of [Chesneau et al., 2016] being an EKF, it also needs a good initial estimate of the attitude and speed, which may make the reader think that we just moved the problem away. However, we empirically observe that the MI-DR filter converges very quickly, with a broad convergence basin, and is not that sensitive to initialization error. Furthermore, the linearization error that the EKF induces during initialization phase does not propagate to the non-linear estimator, as we only use estimated values as an initialization point.

[9] http://opencv.org

somewhat conservative: we prune many features. This is not such an issue in our case, as our estimator can handle situations with few or even zero features thanks to the tight fusion.

We triangulate a landmark only if the rotation-corrected pixel disparity of its observations is larger than a threshold, and integrated into the cost function only if the initial triangulation gave reprojection error lower than a threshold. This threshold is chosen loose in order to make visual information enter the cost function, even if the poses used for their triangulation were not accurate. This is of particular relevance when the estimate drifted in the absence of visual information for a significant duration.

For keyframe selection, we use the strategy of [Engel et al., 2018]: the new keyframe decision is triggered by a threshold on a linear combination of average disparity and rotation corrected disparity since the last keyframe.

5.6 Experiment on Real Data

5.6.1 Hardware and Dataset

5.6.1.1 Hardware

Two hardware prototypes were used during the second part of this thesis. They are both depicted on Figure 5.8. The first hardware (a) featured strong synchronization capabilities, with camera triggered by the MIMU hardware, while the second hardware (b) did not provide such a feature: resynchronization had to be handled by software.

If hardware (a) was tried at first for monocular/MIMU experiment, this unit showed mediocre results. We suspect the ferric connectors of the camera were perturbing too much the magnetic field. On hardware (b), a large distance between the MIMU hardware and the camera was introduced to minimize as much as possible the effect of the camera ferric parts on the field measured by the MIMU. Note that this distance is necessary here only because the camera is a commercial camera, that was not explicitly designed to reduce its magnetic footprint. Nothing prevents *a priori* having a magnetically neutral camera. In wait for a fully integrated prototype, we used montage (b) for the dataset recording.

As in hardware (b) MIMU and images are not synchronized, and as the presented sliding window algorithm expects a synchronized stream of data, we run our algorithm offline, after a synchronization preprocessing of the data. The synchronization preprocessing consists in measuring the delay between IMU sample timestamp and images timestamp in the beginning and the end of the trajectory. We use here the Kalibr toolbox of [Furgale et al., 2013], and compensate clock drift and offset between the two sensors.[10]

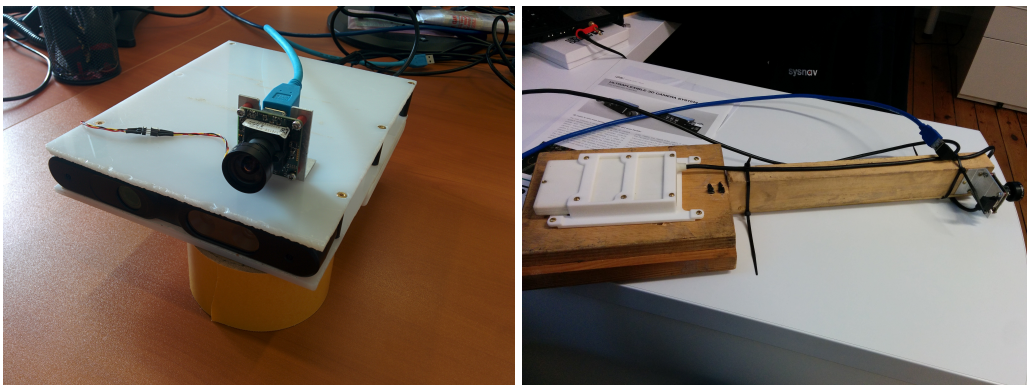
5.6.1.2 Datasets Used for Experiment

In order to test our algorithm, we recorded MIMU and camera data while carrying the hardware and walking around. The environment of the dataset mainly consists of a building, its no lit basement, and its outdoor neighborhood. The movement results from the fast walk of a pedestrian and the camera is mostly looking forward (i.e., with z-axis in the direction of the movement). The main characteristics of the dataset are summarized in Figure 5.9.

Unfortunately, as the scene traveled is mainly indoor, it was not possible to obtain a ground truth by GNSS. We furthermore stress that the trajectories depicted on the satellite map of Figure 5.9, are from our estimator pose output. In order to assess the quality of our estimate, and particularly its scale, we provide superpositions of estimated trajectories with an orthoimage from the IGN[11] having half-meter pixel resolution. Each trajectory was aligned using the detection of a fixed

[10] Note that we documented in Appendix C, the detrimental effect of a slight missynchronization on the position estimate.

[11] http://ign.fr/



(a) (b)

Figure 5.8: Hardware used for the dataset capture. The prototype (a) featured strong synchronization capabilities, with camera trigger controlled by the MIMU ardware, while the prototype (b) did not provide such a features: resynchronization must be handled by software. We used montage (b) for the experiment because the large distance between MIMU sensor and camera permits to remove any effect related to ferric parts of the camera on the field measured by the MIMU.

checkerboard marker at the starting position. Position and heading of the checkerboard are set only once for all trajectory (and not once per trajectory).

5.6.2 Implementation Details and Parameters Choice

We use the MATLAB[12] wrapper of the factor-graph based GTSAM[13] library for the cost function construction, maintenance and minimization. We build the magneto-inertial preintegrated measurement on top of the pure IMU one already included in the library and whose implementation follows closely the description of [Forster et al., 2015].

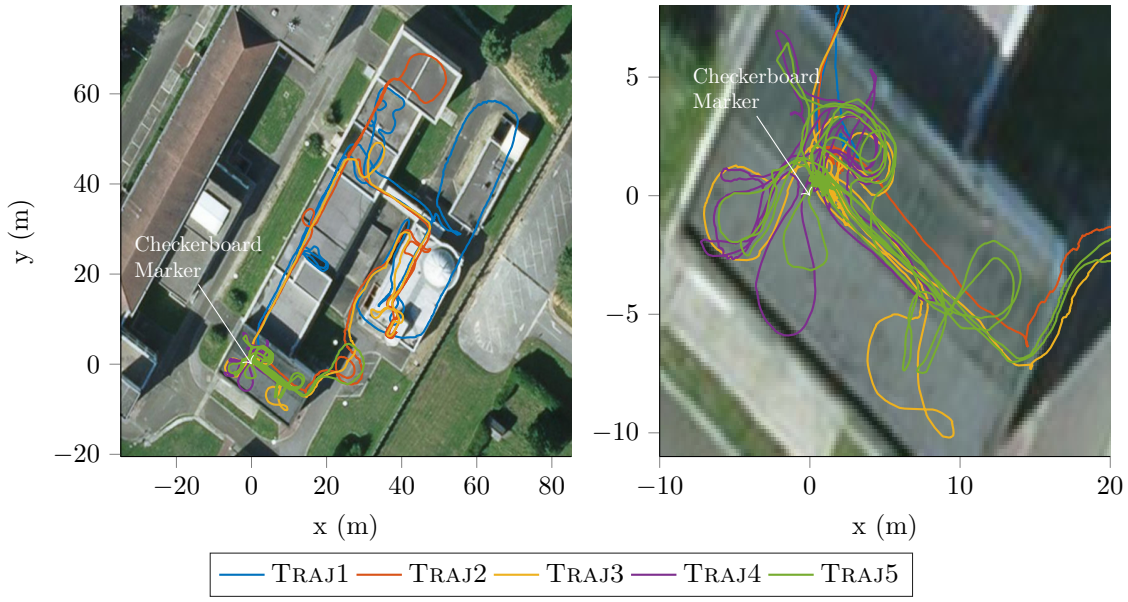
For robustifying the magnetic residual, we use the Tukey M-Estimator loss function depicted in Figure 5.10.[14]

Compared to classical alternatives choice (for instance Huber or Cauchy norm) Tukey loss is pretty aggressive: the flat extreme regions do indeed discard totally the influence of outliers during the algorithm iterations.

In order to improve further the robustness of the solution, we sometimes avoid inserting the magneto-inertial residual between frames and prefer to insert a pure inertial residual. Indeed, in the case of very low gradients of the magnetic field, the added information is mainly noise and tends to corrupt the estimate. We use a fixed threshold on the spectral norm of the gradient matrix to decide if the full MIMU residual should be used.

The following table summarizes the main parameters we had to set along with how reasonable default values have been chosen, and their values in our experiments when we could disclose them.

[12] http://mathworks.com
[13] https://bitbucket.org/gtborg/gtsam/
[14] If we were to define the problem as a Maximum A Posteriori estimation, this loss should match some expected residual probability density. However, we do not actually seek to justify our cost function as a MAP estimator. The robust loss is here considered as a parameter of the algorithm which is tuned empirically.



(a)

MIMU sample frequency	325 Hz
Image sample frequency	20 Hz
Typical keyframe frequency	10 Hz - 1 Hz
Image resolution	640×512 pixels
Image field of view	95 deg
Location	LRBA building K
Position ground truth available	No
Metrics used	Final drift using a fixed checkerboard in percentage of trajectory length

(b)

Figure 5.9: Dataset characteristics (a) All trajectories of the dataset superimposed with satellite image of the environment. (b) Main characteristics of the Indoor/Outdoor dataset.

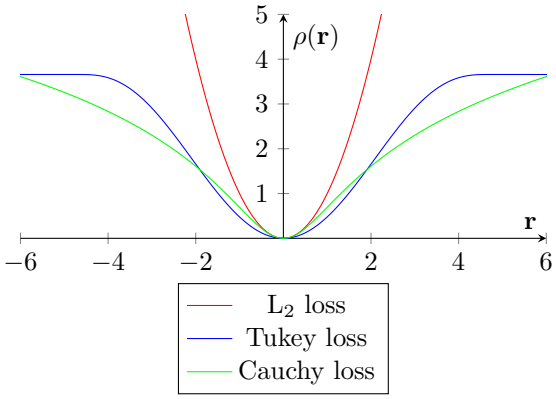


Figure 5.10: Tukey biweight loss function used on the magnetic part of the residual in $\mathbf{r}_{\text{mimu};i}$ compared to the L_2 loss and the Cauchy loss used for landmark observation error.

Parameters	**Tuning cue**	**Value**
Covariance of gyro white noise	from Allan Variance	8e-5 rad.s^{-1}.$\sqrt{\text{Hz}}^{-1}$
Covariance of acc white noise	from Allan Variance	3e-3 m.s^{-2}.$\sqrt{\text{Hz}}^{-1}$
Gyro random walk	from Allan Variance	1e-5 rad.s^{-2}.$\sqrt{\text{Hz}}^{-1}$
Acc random walk	from Allan Variance	1e-4 m.s^{-3}.$\sqrt{\text{Hz}}^{-1}$
Gyro random walk time constant	from Allan Variance	1800s
Acc random walk time constant	from Allan Variance	1800s
Covariance of magnetometers white noise	peak to peak noise	n.c.
Uncertainty of magnetic gradient and prediction	magnetometers noise through gradient computation (amplified)	n.c.
Visual window size	[Leutenegger et al., 2015]	7
Visual-magneto-inertial window size	[Leutenegger et al., 2015]	3
Uncertainty of observation point	roughly one pixel	$\sqrt{2}$ pixels
Robust norm on visual observation	arbitrary	Huber with default scale
Robust norm on magnetic residual	arbitrary	Tukey with default scale
Max initial reprojection error	arbitrary	15

Table 5.1: List of parameters tuned for the sliding window smoother. A majority of them are tuned based on sensors characteristics. Few of them are chosen empirically.

5.6.3 Results discussion on an Indoor/Outdoor/Dark dataset

In the following results we will call our full system MVINS, while VINS will denote our systems without magnetic error term. MI-DR *filter* will denote the filter presented in Chapter 2 (EKF on MIMU). We show in this section comparative results of each of these algorithms, demonstrating that the magnetic error terms render MVINS systems more robust to bad illumination situation.

5.6.3.1 Results of MVINS Compared to VINS

Figure 5.11 illustrates a typical case where the magneto-inertial residual improves robustness of the estimator in low light area. Figure 5.12 shows the Z estimated for each method on the same dataset. In dark areas, the VINS estimate starts to drift because too few corner points are detected for an extended duration. Then, the VINS estimate becomes very discontinuous before going eventually back near to the trajectory of MVINS estimate when new corners can again be detected. The observations presented are actually not specific to this dataset but quite general over all trajectories tested.

This sudden return to a correct position is best seen on the following Z-profile of the trajectory.

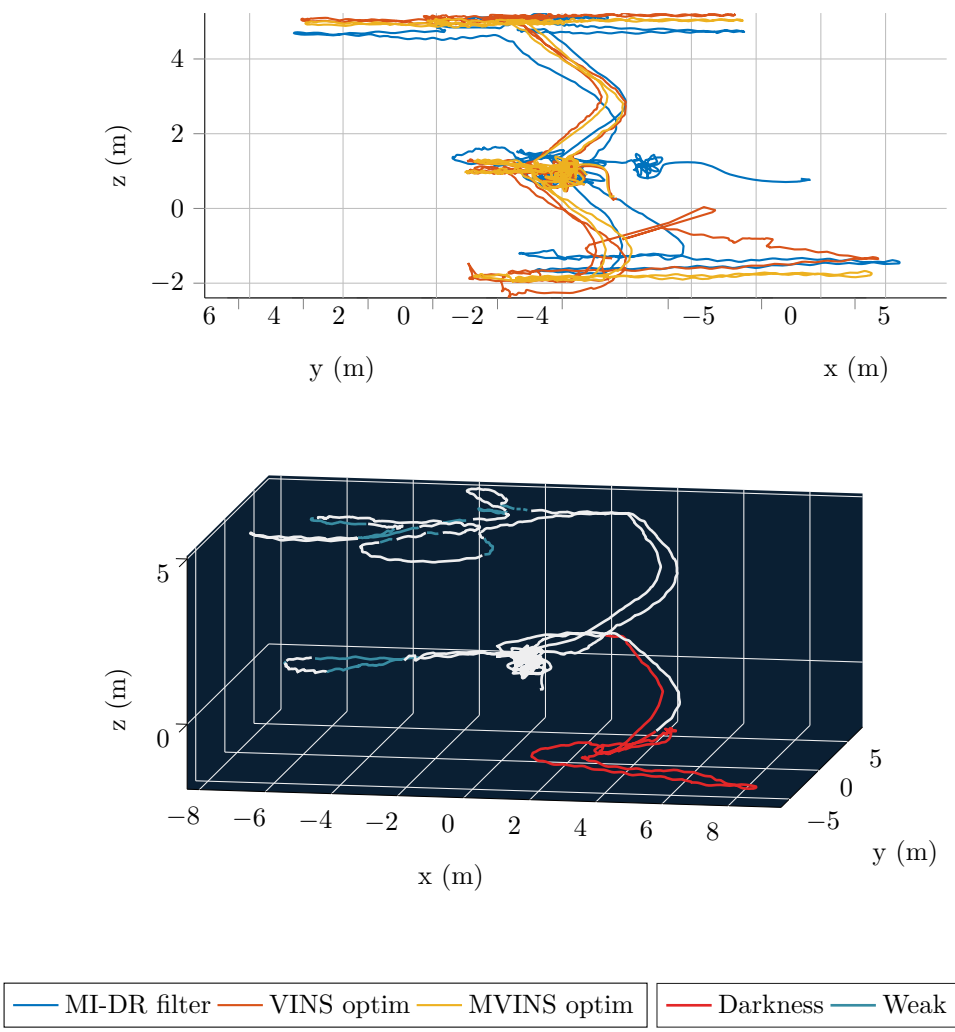


Figure 5.11: Typical improvement of our system compared to pure VINS in dark area. TRAJ4 dataset.

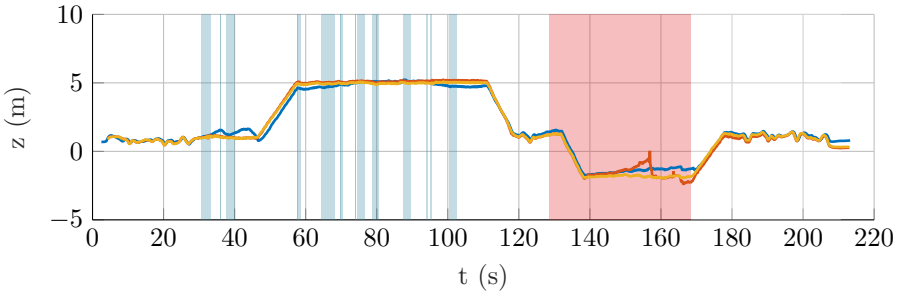


Figure 5.12: Z-profile on the TRAJ4 dataset. Line and background color are the same as in 5.11

This effect could seem surprising for people used to pure visual odometry. It actually arises thanks to the correlation lying into the prior residual. These are able to correct the trajectory when the local speed becomes locally observable again. Note that, on this graphics, this effect appears in the part of the trajectory still annotated in *darkness*. The *darkness* part of the trajectory (in red) is defined here and hereafter as all instant where the received image intensity average is below a threshold; it does not imply that no vision corners could be used at all. That is why the VINS trajectory seems to correct itself toward the MVINS trajectory right in the middle of a dark area. In fact, the correction appears at a time step where visual features are detected on a small part of the image, while most of it is totally black.

5.6.3.2 Results of MVINS Compared to MI-DR

The Figure 5.13 shows the results on three trajectories, each one doing loops through outdoor and indoor scenes. The right part of the figure shows MVINS trajectories with color-coded information on the environment to report parts of the trajectory which are in dark or outdoor conditions (the latter ones being detected by a weak magnetic gradient).

We observe the following general trend. MI-DR filter exhibits a significant drift and even an erratic trajectory in the weak gradient part of the trajectory (in blue). These phenomena appear on all trajectories and always lead to a higher final drift. In contrast, VINS does not exhibit significant drift in outdoor scenes, and MVINS method closely follows the VINS estimate in the areas which are difficult for MI-DR filter.

5.6.3.3 Comparison of Least Squares and Robust Magnetic Error

The previous results could not have been obtained with a standard L_2 loss function. In fact, we observe that the robust loss function drastically improves robustness when magnetic field does not follow MI-DR hypotheses. One example of this behavior can be illustrated on TRAJ3.

On the Figure 5.14, we draw in pale yellow the trajectory obtained without the robust loss on magnetic prediction residual. As can be seen, in the outdoor area of the trajectory, the non-robust estimator output becomes suddenly erratic for some time. This corresponds to a moment when the measured gradient is above the threshold – so that magnetic prediction residuals are not discarded – but the magnetic field perceived is not totally stationary for some reasons. In contrast to the non-robust version, the robust version stays smooth all along the trajectory and also drifts slower.

On the Figure 5.16, we can assess precisely the effect of the robust norm in the case of Traj3. The figure shows curves that describe the following quantities across times: the magnetic residual, the robust norm induced weight of this residual at the end of iterations and the vision reprojection error distribution. (each colored line is a quantile of the distribution.) The bad behavior of the non-robust estimator occurs at 160 seconds. The middle curve in gray shows that indeed, at that timestamp, the robust estimator downweight the magnetic residual drastically (up to utterly annihilating it), while the value of the residual becomes big in absolute value. This means that they are non-modeled effect in the magnetic field that robust loss succeeds in coping with.

(a) Traj1 Length: ∼ 530 m

(b) Traj2 Length: ∼ 380 m

(c) Traj3 Length: ∼ 368 m

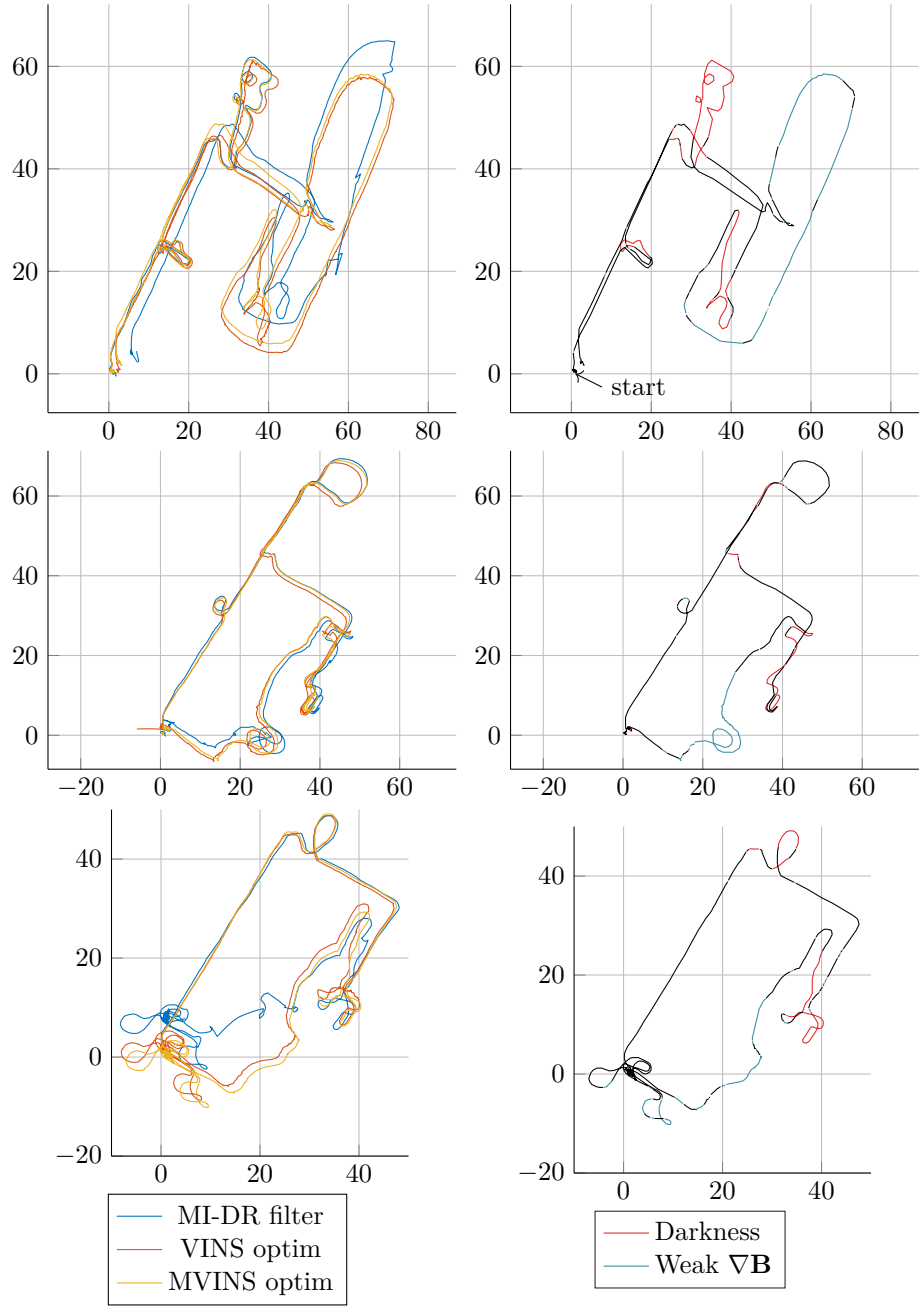


Figure 5.13: Overview of Traj1, Traj2, and Traj3 trajectories as reconstructed by the MI-DR filter, our sliding-window smoother without magnetic residual, and our full sliding window smoother. The right part of the plot is a color-coded description of the visual and magnetic environment.

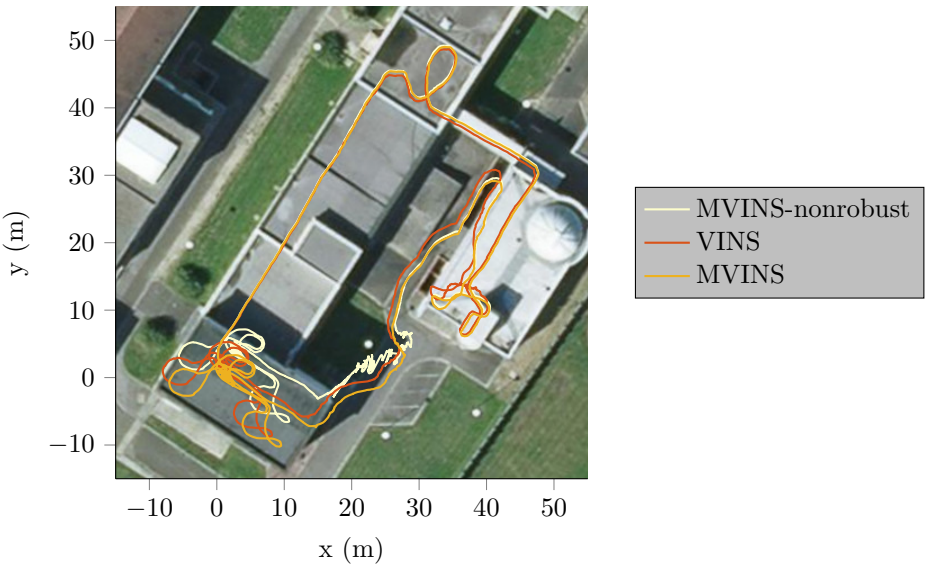


Figure 5.14: Influence of the robust norm on TRAJ3 dataset. MVINS-nonrobust is the sliding window estimator with full magneto-inertial residual but where robust norm is not employed on the magnetic part. Final drift: MVINS-nonrobust: 3.62m, VINS: 1.173m, MVINS:1.14m

The beneficial effect of the robust loss was also significant on TRAJ5 dataset as shown in Figure 5.15. In this trajectory, strong non-stationarities of the magnetic field completely corrupts MI-DR filter estimate, the three floors of the trajectory are not even recognizable. The trajectory of the non-robust version of our sliding window estimator tends to drift erratically in the same direction as MI-DR estimate while MVINS stays close to the better performing VINS optimization method. MVINS even improve the VINS estimate in the basement area (in red), which is better handled (note the smoothness of the yellow curves with compared to the red one). Final drift is also improved as shown in the table on Figure 5.15.

5.6.4 A Word about Runtime Performance

The implementation used for the experiment was a MATLAB/GTSAM prototype not designed to be run in real-time. However, the method should be able to ultimately run in real-time on a modern embedded processor with a proper implementation. Indeed, [Qin et al., 2017], presents a very similar algorithm – without magnetometers error term – that run in real-time on a smartphone.

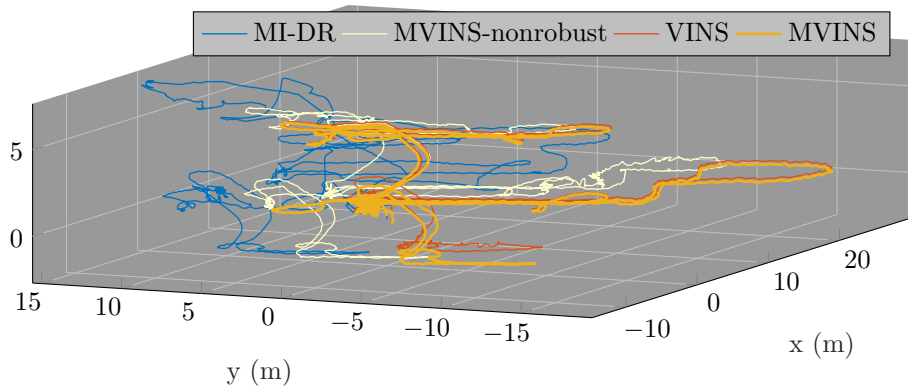
The difference between our algorithm and theirs potentially impacting negatively the performance are the following:

- ours adds a vector of dimension three by timestep to the state. In results, the Jacobian and Hessian matrices are slightly bigger.
- ours add some computation overhead during propagation to propagate the preintegrated magneto-inertial measurement and their derivative with respect to biases.
- ours add two new residual term of size three that depends exclusively on the MIMU states and pose (not on landmark position). In results, Jacobian are bigger.

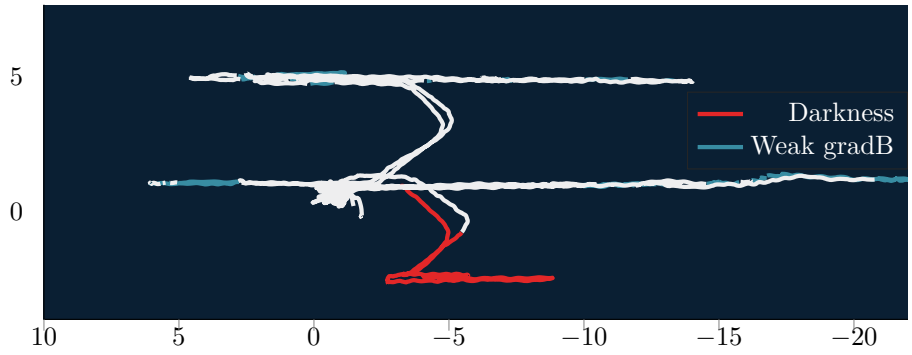
Note that in contrast, the following performance critical aspects of the algorithm are not modified:

- integration of constraint inter-keyframe is done only once in the preintegration process
- global sparsity structure of the optimization is not modified

We argue that the runtime performance will not be drastically reduced compared to [Qin et al., 2017]. We, however, admit these minor overheads could lead to a slightly different trade-off between performance and accuracy. The main parameters which can be tuned to lower computational



(a)



(b)

Algo.	final translational drift
MI-DR	2.94%
VINS	0.25%
MVINS-nonrobust	1.94%
MVINS	0.18%

(c)

Figure 5.15: Influence of the robust loss on Traj5 dataset.(a) estimate trajectory (b) environmental condition along the trajectory (c) final translational drift (% of trajectory length).

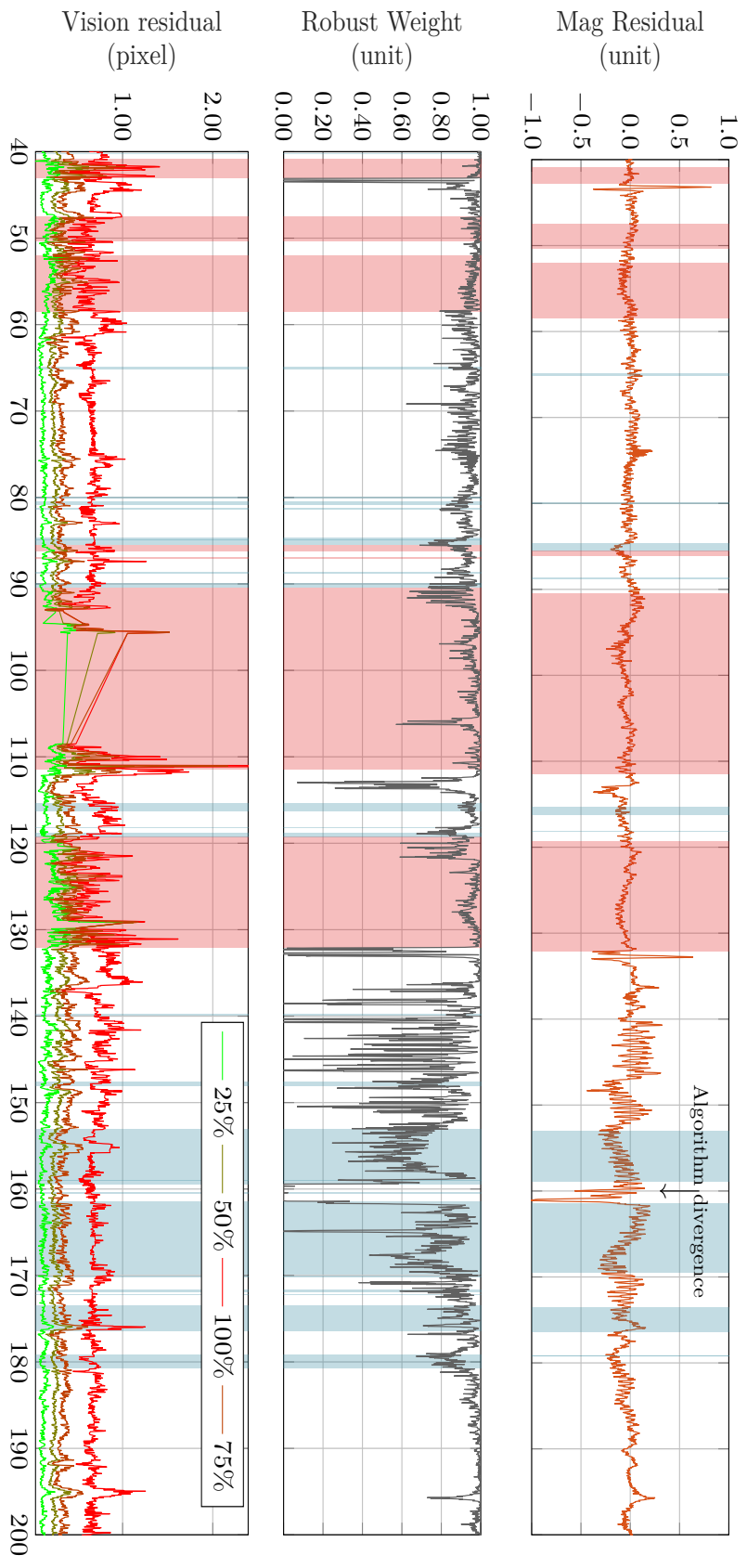


Figure 5.16: Residual, mag weight and vision residual quantiles for Traj3. Transparent red background is for time of bad illumination. Transparent blue background if for time of weak gradient. (in blue areas, the magnetic prediction residual is deactivated, while in red areas, some opportunistic features can rarely still be tracked.)

burden, if required, are the number of features tracked or the size of the sliding windows.

5.7 Trajectory Quality After a Long Run of the Estimator

This section documents a problematic phenomenon we observe on our estimate that is not fully understood and corrected yet. We still describe here for exhaustivity.

5.7.1 Corruption of Local Consistency Trajectory Estimate

If the addition of magnetometer information renders the trajectory smoother indeed when vision information is temporarily unavailable, we still observed on our real data discontinuous trajectories, especially after a long run duration of the algorithm. This effect is generally significant and visible on our dataset at the end of trajectories where the pedestrian comes back in front of the checkerboard marking the starting point. An example is shown in the left side of Figure 5.17. On the left part of the figure, we draw the trajectory obtained by the MVINS estimator presented in Section 5.5, along with the reconstructed position of landmark used in the estimator. We also show a zoom over the part of the trajectory corresponding to the closure of the main loop. TRAJ3 dataset starts at the position indicated on the figure and follow the loop clockwise. The zoomed version shows the trajectory when the system comes back to the start position, which is marked by a checkerboard. The landmarks in the transparent blue areas are actually detected on the checkerboard.

Before analyzing it, let us stress that these colored point-clouds represent the estimated landmark positions. We underline these point-clouds are *not* a map of the environment, as the landmarks are instantaneously marginalized when the tracking of the corresponding image features is lost. Consequently, (i) the colored dots represent the position of the landmark at their time of marginalization; (ii) several dots can represent the same physical corner, as we do not try to detect old points again once their features track has ended; (iii) the position drift transmits to the point cloud, making potentially objects appear or several time.

As can be seen on the bottom *left* drawing, the checkerboard does not appear solely twice, but many times. Its position actually drifts from bottom left to top right of the figure, before stabilizing. Simultaneously the trajectory – mainly doing circle looking at the checkerboard, seems to be very discontinuous (and not as smooth as the movement really is). We conclude that the trajectory actually drifts for roughly one meter in a few seconds; this is totally inconsistent with the relative low drifts since the beginning of the trajectory.

What we observe here is actually strong corrections of the position of *all* keyframes currently in the windows. This correction occurs through the linear prior error term. If one is interested in the local movement, (for instance if using this estimate for control or of augmented reality applications) these kinds of erratic correction would be dramatic: they arise from estimator mechanics and do not correspond to physical movement!

It seems such corrections arise from long-term correlation – accumulated into the prior – between biases, speed, position, and heading.

5.7.1.1 Workaround

This effect is not totally described in literature and research papers, however, by looking at different open-source implementations of similar systems, we note that some of them present step that would indeed workaround such an issue. These "workarounds" are rarely explicitly documented. We list hereafter three of these strategies found in [Leutenegger et al., 2015], [Qin et al., 2017] and [Engel et al., 2018].

Constrained the gauge at each step by fixing oldest pose in window This strategy is used in [Leutenegger et al., 2015]. Even if not advertised in their paper, their open source implementation constrains the optimization problem to have the oldest pose of the visual window fixed.

They actually find at each timestep the minimum of the optimization problem:

$$\mathcal{E}(\boldsymbol{X}_k) = \sum_{(i,j)\in\mathcal{O}_k} \|\mathbf{r}_{\text{proj};il}(\boldsymbol{\xi}_i, \mathbf{l}_j)\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} + \sum_{i\in\mathcal{S}_k} \|\mathbf{r}_{\text{imu}}(\boldsymbol{\xi}_i, \boldsymbol{s}_i, \boldsymbol{\xi}_{i+1}, \boldsymbol{s}_{i+1})\|^2_{\boldsymbol{\Sigma}_{\text{imu};i}} + \|\mathbf{r}_{\text{prior;k}}(\{\boldsymbol{\xi}_i\}_{i\in\Upsilon_k}, \boldsymbol{\xi}_{k-|\mathcal{S}_k|}, \boldsymbol{s}_{k-|\mathcal{S}_k|})\|^2 + \|\mathbf{r}_{\text{fixed}}(\boldsymbol{\xi}_{\text{oldest}})\|^2_{\frac{1}{\epsilon}\mathbf{I}_4} \quad (5.65)$$

Where $\mathbf{r}_{\text{fixed}}$ is the difference between yaw angle, global position of current estimate and yaw angle and global position of the estimate of the oldest frame in the window at last timestep. ϵ is around the machine epsilon.

This error term is only added for computing iterations, and not during the marginalization phase of the algorithm. We argue however that the use of this artificial error term to fix the Gauge is not ideal. Indeed, during optimization of the cost, it plays the role of a pose measurement that forbids movement of absolute position. However, this position is correlated with the biases estimate. Fixing the position, also imposes a strong constraint on the bias estimate, and could slow down its convergence.

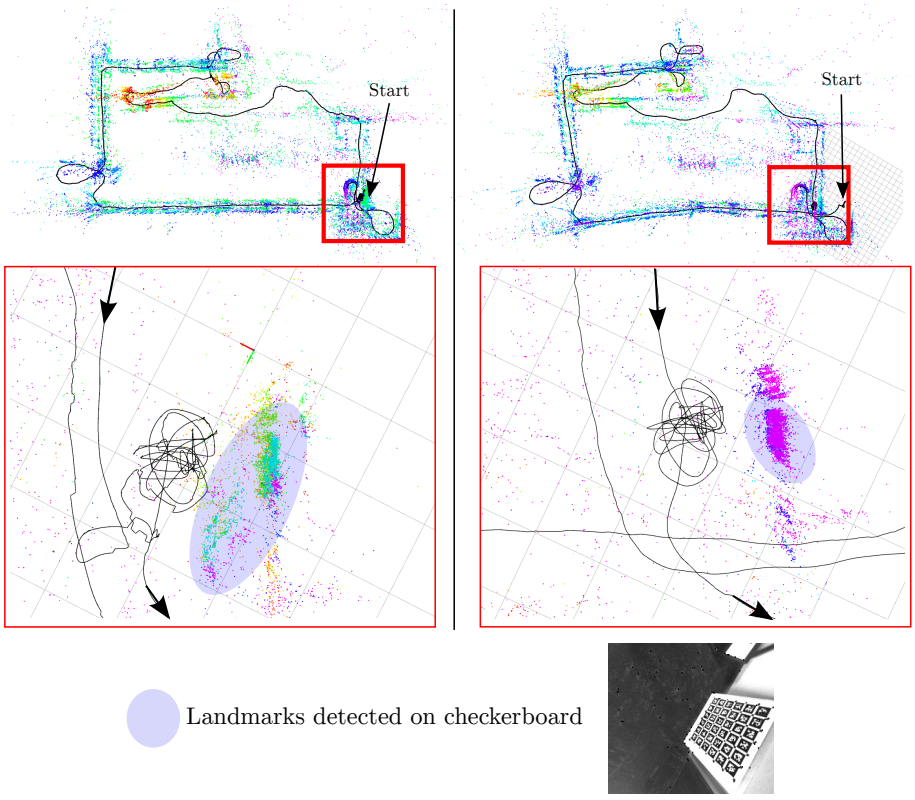


Figure 5.17: **Left:** Overview of the output trajectory and used 3D landmark on Traj3 of the sliding window estimator. **Right:** Overview of the output trajectory and used 3D landmark on Traj3 of the sliding window estimator modified in order to fix the first keyframe pose of the window at each instant. See text for explanation.

Free Gauge Optimization and State Reset In contrast, [Qin et al., 2017], does not fix anything during the optimization and optimize directly the cost (5.60). However, in order to smooth the estimate, they transform (thanks to a rigid transform) the found minimizer so that the yaw angle and the absolute position of the first keyframe are exactly the same as before the optimization. They hence enforce continuity of the estimate.

Projection of Iteration Increment on known observable Subspace Another strategy is to project at each iteration the linear increment in a way that applying it does not change the value of some combination of variables. This approach seems to be used in the code of [Engel et al., 2018].

We implemented a variant of the first workaround. We observed a trajectory that is displayed on the right of Figure 5.17. The trajectory obtained is indeed smoother over the entire duration, but also exhibits stronger drift compared to when not using the workaround. This error stems mainly from heading drift.

We are thus facing a trade-off between two desirable properties for our *dead-reckoning* systems: low drift versus local continuity of the estimate.

5.8 Discussion and Conclusion of this chapter

5.8.1 Chapter Summary

In this chapter, we presented a way to use magnetometers information into an optimization framework.

Inspired by recent work on preintegration technique for IMU data, we formalized magneto-inertial preintegration measurements. These preintegrated measurements were used to build magneto-inertial error terms that actually retain the computational benefits of the pure IMU preintegrated measurements and can thus be used in an optimization-based framework with small overhead. We validated that these MIMU error terms can be used alone to reconstruct a trajectory.

As an application of these error terms, we build a sliding window smoother based on the developed cost function that combines constraint coming from visual and magnetic information. We implemented a marginalization strategy in order to keep computational cost bounded.

We showed on real and realistic datasets that the resulting cost function was indeed adequate for fusing the data from a monocular camera and the MIMU system advantageously, taking benefits from both sensor strength.

We also assessed the critical requirement of using a robust loss on magneto-inertial residual in order to filter out the effect of non modeled magnetic field effect.

5.8.2 Limitations and Critics

Here are some limitations and some criticisms one might object with respect to the work presented here, of which we are fully aware.

- the evaluation method is not perfect. Its main weak point is the absence of ground truth trajectory along the *entire* trajectory. If we were careful not to overclaim result, it would be necessary to have a dataset where a ground truth is available all along the trajectory in order to compute a more significant metric (for instance the root-mean-square error of the position vector). This kind of ground truth can be given by a motion capture room for indoor parts and GNSS for outdoor parts of the trajectory. Note that in both cases, these ground truth acquisition technique constraint the environment to either indoor or either outdoor scenes, which is not the scenario we were aiming for.
- in order to implement practically the cost function, we had to rely on early marginalizations of variables, thus fixing linearization point early in the process. Actually, in our implementation, only the first three poses, MIMU states, and landmarks (the visual magneto-inertial window) are not yet linearized around a point. Realizing that, the reader might wonder if there are really any benefits from using the full non-linear error term on such a few states, compared to a pure filtering framework for instance. We actually bring some performance comparison between a filter and the sliding window smoother presented here in Section 6.7.
- our system does not exploit visual information at their full potential: it does not detect nor correct for loop closures
- our system shows local trajectory discontinuities, we admit that it is not clear yet, if there arises from hardware issues, remaining implementation wrongness or estimator properties, and should be investigated further. Section 6.7 will come back briefly on this issues that also appears in pure filtering context.

5.8.3 Possible Extensions of the work

A fully real-time implementation would be a requirement, as already remarked. This should only require careful programming, without strong changes to the algorithm. Having such an implementation would enable easier and deeper experiments. For instance, coupled with a larger database of datasets, it would enable practical benchmarking capabilities and parameters grid

search, which we think is a requirement for developing such an algorithm for real application, at least from an engineering point of view.

We also think at this point that the development of a proper hardware, combining the two sensors through a real co-conception approach, is another substantial requirement for further performance improvements. This would require skills and resource well beyond the ones used in this work though.

But the most interesting extension of this work would be without any doubt inclusion of this new dead-reckoning approach into a full SLAM framework featuring:

- batch optimization along trajectories through either a pose graph or a full bundle adjustment optimization
- loop closure handling in the past portion of one trajectory
- relocalization capabilities and loop closure handling in a known map.

Such systems are for instance described in [McDonald et al., 2013] or more recently in [Schneider et al., 2017]. The development of such a complete system would require a lot of work and was considered as out of reach for the remaining of the thesis. Furthermore, it was unclear if this work would answer to our central question: "are MI-DR and MIMU useful alongside VINS system?". Indeed, we had the intuition that the magneto-inertial ideas would not bring a lot of improvement to the listed SLAM features: in our understanding, magnetic-aided loop closure would actually require mapping the magnetic field at a more or less coarse level. That could be a challenge for reasons we already explained in Section 5.2. We guess that images, as a more abundant source of information, are way more suited for loop closure detection and relocalization pose computation.

In between the time this work was done and the writing of the thesis, two new open-source software were released that could serve as a basis of such a SLAM system: [Schneider et al., 2017] and [Qin et al., 2017]. Notably, the first one is designed to be modular, and research oriented. It was used as the basis for successful experiments: [Burri et al., 2015; Bürki et al., 2016; Fehr et al., 2016] If these systems had been released earlier, we would surely have taken the time to integrate our dead-reckoning system into their framework. Instead, we focused on using a cheaper estimator for solving the dead-reckoning problem.

Some work is also needed to include the last progress of MI-DR technique, into our optimization framework. Such as rejection of power line magnetic interferences described in [Chesneau et al., 2016].

We also rejected rather quickly, at the beginning of this chapter, the possibility of mapping the magnetic field and gave some challenges associated. These challenges are not easy to handle in our opinion and could be addressed: (through magnetic fingerprinting method) we considered the subject as out of our scope for this thesis.

5.8.4 The Remaining of the Dissertation

The remaining of the thesis focus of a filtering approach solving the same sensor fusion problem. Such filtering approach, because being very lightweight, could be implemented on embedded hardware more efficiently than the approach presented in this chapter and would be of strong interest in real applications.

Recalling that "filtering is merely the first half-iteration of a nonlinear optimization procedure" [Triggs et al., 2000], we also wanted to verify that the overhead induces by optimization was actually bringing some accuracy gain to our solution, for instance, compared to state-of-the-art filter, as often claimed (e.g., in [Leutenegger et al., 2015]).

The Chapter 6 describes the implementation of an inverse square root filter to solve for the fusion problem, while Chapter 7 build on theoretical properties linked with the parametrization of the filter.

Chapter 6

Why (not) filter?

In this chapter, we study an estimator based on a filtering approach designed to run in bounded time and more efficiently than the sliding window smoother presented in Chapter 5. If the approach of Chapter 5 could be seen as trying to use magneto-inertial techniques in methods developed by the vision-based navigation community, this present chapter does the opposite: using visual information into the preferred estimator used for MI-DR *technology, namely the EKF. This work was described thoroughly in the article [Caruso et al., 2017c], which is here reproduced entirely along with contextualizing comments and a comparison with the estimator of previous chapter.*

6.0 Chapter Introduction

As noted in the introductory chapter of this second part of the thesis, two paradigms have competed for solving the VINS sensor fusion problem: optimization and filtering. The work [Strasdat et al., 2012] argued very solidly in 2012 that, regarding visual SLAM, there were little reasons to use a filtering scheme compared to *sparse* optimization, as this was more computationally expensive *for the same precision*. Their analysis was done in the context of pure visual SLAM, where the 3D map is as of interests as the position. But in fact, if one is more interested in a robust position estimate, as in the dead-reckoning problem we want to solve, a lot of usable estimators cannot be described as pure bundle adjustment or pure filtering: the two extremes studied in [Strasdat et al., 2012]

For instance, their specific definition of filtering, excluded the MSCKF (Multi-State Constraint Filter) filter formulation of [Mourikis and Roumeliotis, 2007] or the sliding window smoother of Chapter 5. These kinds of algorithms belong actually to a middle ground between the filter and bundle adjustment definition of [Strasdat et al., 2012]. Often, algorithms of the literature are said filter- or optimization-based depending on with which "extreme" they shared the most characteristics.

All these "hybrid" estimator (e.g., full bundle adjustment, EKF-SLAM (EKF-SLAM) sliding window smoother based on optimization and marginalization, iterative extended Kalman filter, exactly sparse information filter, etc.) can actually be described in a unified manner in a Bayesian network of factor graph framework. From this point of view, these inference processes differ mainly – and often merely – from the marginalization (or conditioning) strategy employed.

In this chapter, we describe an estimator that is best described as an inverse square root filter. The approach is to extend the MI-DR EKF presented in Chapter 2 with visual information. To do so, we borrow ideas of the MSCKF filter of [Mourikis and Roumeliotis, 2007]: inertial data are used to feed a dynamic model to propagate the filter, while landmark visual observations are used to create a measurement equation linking subsequent poses; following MSCKF methodology this measurement equation is used without having to augment the filter state with proper landmark estimate; a very interesting property for computational reasons. Also, the presented filter shares with EKF the fact that variables are marginalized as soon as possible. (earlier than in the sliding windows smoother of Section 5.5).

Reading notes This work was described thoroughly in the article [Caruso et al., 2017c], which is reproduced entirely starting from next page. The reader that have already read Chapter 5 could safely skip the first few sections of this chapter and goes directly through Section 6.4, Page 117 and Section 6.5 Page 123. These sections present the specificity of the filter estimator and its result on our dataset. Also, notes that the speed state is here expressed in body frames instead of the world frame as done in Chapter 5. The Section 6.7 is an addition compared to the original article and concludes the chapter by an explicit comparison between the filter described and the sliding window estimator of the previous chapter.

6.1 Introduction

6.1.1 Motivation

Infrastructure-less navigation and positioning in indoor location is a technical prerequisite for numerous industrial and consumer applications: ranging from lone worker safety in industrial facilities, to augmented reality. Still, it remains an open challenge to efficiently and reliably combine embedded sensors to reconstruct a position or a trajectory. In the present work, we address this challenge restricting ourselves to the following sensors: MEMS gyroscopes, accelerometers, magnetometers and a standard industrial vision camera. We motivate this choice by the fact these sensors are cheap and can easily be embedded in a wearable form factor, which makes this combination appealing for pedestrian applications. Moreover, VINS (Visual-Inertial Navigation Systems) literature, showed recently tremendous progress in the past few years.

6.1.2 State of the Art and Contribution

Indeed, if a wide range of embedded visual sensors were previously presented to solve the problem, such as rotating LIDARs [Zhang et al., 2014] or depth sensors [Guo and Roumeliotis, 2013], much of the recent efforts focused on conventional cameras, as they are cheap, and already present in a wide range of lightweight devices, such as smart-phones. While authors of [Konolige et al., 2010; Leutenegger et al., 2015; Usenko et al., 2016] rely on multiple embedded cameras; single cameras solutions have been shown to provide good results. For instance, authors of [Li and Mourikis, 2013; Hernandez et al., 2015] present efficient filtering methods for monocular VINS. Nonetheless, monocular VINS remains highly sensitive to degenerate motions scenarios. For instance, the scale factor is weakly constrained in case of a steady motion and pure rotation motions must be tackled with special care in the estimation process. Moreover, current VINS implementations rely heavily on high-frequency visual corrections (10–20 Hz), and break when the visual environment is not adequate for more than a few seconds (bad illumination, presence of smoke, motion blur, etc.).

Compared to the previous literature, we explore a sensor suite alternative to VINS and able to provide precise and robust navigation information. We combine the vision sensor with a MIMU (Magneto-Inertial Measurement Unit), i.e., an IMU sensor augmented with *an array of* magnetometers. Within a stationary and non-uniform magnetic environment, the magnetic measurements render the body speed observable, which has been shown to improve motion prediction significantly compared to IMU alone [Dorveaux et al., 2011; Chesneau et al., 2016]. This technique, named Magneto-Inertial Dead-Reckoning (MI-DR) hereafter, is perfectly fitted for indoor navigation as human-made environment, wall, floor, and pieces of furniture all perturb the magnetic field in a significant way. As a result, the open-loop position error of this method is often around a few percents of the trajectory length in indoor environments (see [Chesneau et al., 2017]). MI-DR fails however in places where the gradient of the magnetic field vanishes, (commonly outdoor) and lacks robustness when the magnetic environment is not stationary.

We have already presented various approaches to fuse the information from MIMU sensor with vision sensors for dead-reckoning estimation. In [Caruso et al., 2016], we proposed a semi-tight fusion scheme combining a depth sensor with a MIMU to increase robustness and availability of the position/orientation informations. Yet, we finally turned towards conventional monocular cameras, mainly because the limited range of depth sensors makes them unable to improve the MI-DR estimation in large rooms or outdoor. In order to still be able to estimate the scale accurately, we also turned towards a fully tight fusion scheme, which includes estimation of camera pose, current speed, magnetic field and inertial sensor biases. We investigated an optimization-based solution in [Caruso et al., 2017b] and a filter-based solution in [Caruso et al., 2017a]. Here we present an extension of the work presented in the conference paper [Caruso et al., 2017a] with new results and a slightly different implementation of the filter. The used estimator is still inspired by the pure VINS method presented in [Wu et al., 2015] for the visual measurements and by the magnetic prediction and measurement process of [Chesneau et al., 2016] for the MIMU handling process. We will call it MI-MSCKF for Magneto-inertial Multi-State Constraint Kalman Filter. As in [Caruso

et al., 2017a], we choose a square-root implementation which leads to better overall conditioning of matrices operations involved in the filtering process. The inverse form is also computationally interesting for high-dimensionality measurements [Anderson and Moore, 1979, p.141].

We show on experimental data that the MIMU provides robustness in situations where vision fails, and that, reciprocally, vision allows the system to handle cases where the magnetic gradient is to low for the MIMU to work correctly, typically in outdoor situations.

6.1.3 Paper Organization

After introducing general notations and conventions in Section 6.2, we describe the dynamic model our filter is built on in Section 6.3, with a focus on the magnetic prediction equation, less known in the visual-inertial literature. This section also presents the model discretization and error model of the sensors. The Section 6.4 describes the filter design: chosen state and equations for propagation and measurement update steps. The last section (Section 6.5) presents comparative results of trajectory estimation obtained on real datasets.

6.2 Notations [Same as in this thesis]

6.2.1 General Conventions

Bold capital letters $\mathbf{X}$ denote matrices or elements of manifold. Parenthesis are used to denote the Cartesian product of two elements $a \in \mathcal{A}, b \in \mathcal{B} \mapsto (a, b) \in \mathcal{A} \times \mathcal{B}$ and brackets for the concatenation of two row vectors. For a vector $\mathbf{x} = [x_1, x_2, x_3]^\mathsf{T}$, $\mathbf{x}_2$ denotes its second component x_2 and $\mathbf{x}_{2:3}$ is the sub-vector $[x_2, x_3]^\mathsf{T}$. For matrices, we define the vectorization operation $\mathrm{Vec}\,()$ so that:

$$\mathrm{Vec}\left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}\right) = \begin{bmatrix} x_{11}, x_{21}, x_{12}, x_{22} \end{bmatrix}^\mathsf{T}. \tag{6.1}$$

$\mathrm{A} \otimes \mathrm{B}$ will denote the Kronecker product of matrices A and B. ∂_A will be a shorthand for the derivative with respect to the coefficients of $\mathrm{Vec}\,(\mathrm{A})$. The notation $\|\mathbf{x}\|_{\mathbf{\Sigma}}$ is the Mahalanobis norm of invertible covariance $\mathbf{\Sigma}$: $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \mathbf{x}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{x}$. $\mathbf{I}_n$ is the identity matrix of size n and $\mathbf{0}_{n\times m}$ the zero matrix of size $n \times m$. $\mathbf{I}_n$ is the identity matrix of size $n \times n$ or corresponding application. $\boldsymbol{\mathcal{O}}(n)$ is the orthogonal matrix group of size n.

6.2.2 Reserved Symbols

Generally and except stated otherwise we use the following symbols: $\mathbf{p}$ for the translational part of the body pose, $\mathtt{R}$ for its rotational part. $\mathbf{v}$ for the velocity, $\mathbf{b}_a$ and $\mathbf{b}_g$ for inertial sensor bias, $\mathbf{B}$ for the magnetic field, $\nabla\mathbf{B}$ for its 3×3 gradient matrix. $\boldsymbol{\omega}$ is used for the rotational speed and $\mathbf{a}$ for the specific acceleration. 3D landmark positions are noted with the letter $\mathbf{l}$ and these observations into an image with the letter $\boldsymbol{o}$. We use a tilde symbol for measured quantities or generally quantities that can be derived from sensor reading. We use a hat for estimated versions of a physical quantity. $\mathbf{g}$ symbol is kept for the gravity vector in inertial coordinates. The world coordinates are defined such that the gravity vector writes $\mathbf{g} \simeq [0, 0, -9.81]^\mathsf{T}$. When ambiguous, the reference frame in which a quantity is expressed will be noted in exponent: w stands for the world gravity-aligned reference frame, b for the current body reference frame and c for the camera frame. The Figure 6.1 summarizes the chosen notations.

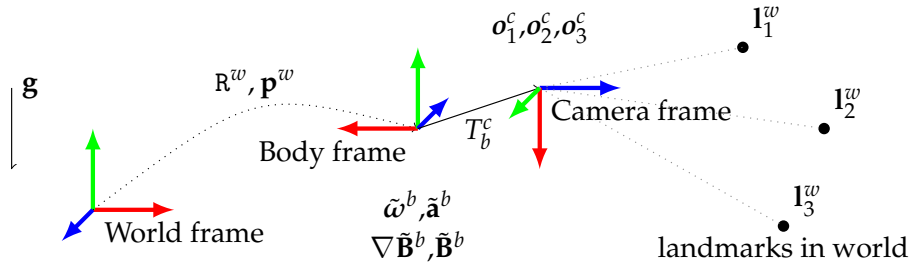


Figure 6.1: Reference coordinate frames at play in the problem, with associated typical measurements.

6.2.3 Rotation Parametrization

For rotations, we use the convention that $\mathtt{R}^w$ transforms a vector from body frame to world frame by left-multiplying it. For the sake of clarity of the developments, we represent the attitude of the sensor as a rotation matrix. The Special Orthogonal Group is denoted SO(3)and its associated Lie algebra $\mathfrak{so}(3)$– the set of skew symmetric matrices. Any element of $\mathfrak{so}(3)$can be identified with a vector of $\mathbb{R}^3$: $[x]_\times \in \mathfrak{so}(3)$ with $x \in \mathbb{R}^3$ and vex so that $vex([x]_\times) = x$. exp and log are the standard exponential map and logarithm on SO(3). As in [Forster et al., 2015], we use vectorized versions of exp and log:

$$\text{Exp}: \begin{array}{ccc} \mathbb{R}^3 & \to & \text{SO}(3) \\ \boldsymbol{\delta\theta} & \mapsto & \exp([\boldsymbol{\delta\theta}]_\times) \end{array} \tag{6.2}$$

and $\text{Log} : \text{SO}(3) \to \mathbb{R}^3$ the inverse function. With these conventions, $\text{Log}(\text{Exp}(x)) = x$.

6.3 On-Board Sensors and Evolution Model

6.3.1 Sensing Hardware

The MIMU sensors (see Figure 6.2) provide raw measurements of biased proper acceleration $\tilde{\mathbf{a}}^b$, biased angular velocity $\tilde{\boldsymbol{\omega}}^b$, magnetic field $\tilde{\mathbf{B}}^b$ and its gradients $\nabla\tilde{\mathbf{B}}^b$. The latter is a 3×3 matrix which elements are estimated by finite differences between signals recorded on an array of magnetometers. These sensors are carefully calibrated offline and registered in the same spatial coordinate frame with a method similar to [Dorveaux et al., 2009]. The resulting MIMU coordinate frame, centered around the magnetometer and accelerometer (which are assumed colocalized here), will be used as the *body* frame in all subsequent derivations. We use a global shutter camera modeled as a pinhole camera with instantaneous exposure. In practice, recorded real images are undistorted using intrinsic calibration parameters. Intrinsic parameters (focal length in pixels (f_x, f_y), principal point coordinates (c_x, c_y) and distortion coefficients) are assumed to be known from a preliminary calibration. The pinhole camera projection function π maps a landmark 3D coordinates $\mathbf{l}^c$ expressed in the camera frame to the pixel coordinates of its projection onto the image:

$$\pi: \begin{array}{ccc} \mathbb{R}^3 & \to & \mathbb{R}^2 \\ \mathbf{l}^c & \mapsto & \begin{bmatrix} f_x \frac{\mathbf{l}_1^c}{\mathbf{l}_3^c} + c_x \\ f_y \frac{\mathbf{l}_2^c}{\mathbf{l}_3^c} + c_y \end{bmatrix} \end{array}. \tag{6.3}$$

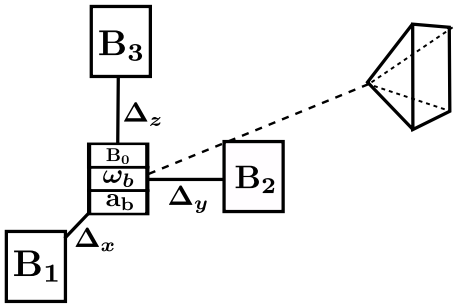


Figure 6.2: Schematic view of on-board sensors. In addition to accelerometers and gyrometers, the MIMU includes several magnetometers: a central one and at least three peripheral ones in order to compute the full 3×3 matrix of magnetic field gradients. The camera is rigidly attached to the MIMU sensor.

The camera is rigidly attached to the MIMU sensor board. The transformation $(\mathtt{R}_c^b, \mathbf{p}_c^b)$ between the camera frame and the body frame is assumed to be known. We could alternatively include this transform into the state filter as done in [Hesch et al., 2014] for instance.

Since hardware synchronization was not possible with the components used here, we use a datation approach. In the online estimation process we make the simplifying assumption that images are captured simultaneously with the MIMU sample closest in time. More precisely, the image information at time t_{image_p} is processed using the state estimate at time t_{mimu_k} with k such as:

$$k = \arg\min_{k, t_{mimu_k} < t_{image_p}} |t_{mimu_k} - t_{image_p}|.$$

The temporal error done with this approach is always smaller than the sampling period of the MIMU sensors, which is often below the exposure time of the camera. Besides, this approximation significantly simplifies time management as all measurements are indexed by the MIMU time sampling.

6.3.2 Evolution Model

In order to estimate the position and orientation of the body coordinate frame in the world frame, one has to track an estimate of its speed and of the magnetic field at the current position of its center. We model the evolution of these quantities with the following differential equations:

$$\dot{\mathtt{R}}^w(t) = \mathtt{R}^w(t)[\boldsymbol{\omega}^b(t)]_\times, \tag{6.4}$$

$$\dot{\mathbf{p}}^w(t) = \mathtt{R}^w(t)\mathbf{v}^b(t), \tag{6.5}$$

$$\dot{\mathbf{v}}^b(t) = -[\boldsymbol{\omega}^b(t)]_\times \mathbf{v}^b(t) + \mathtt{R}^{w\mathsf{T}}(t)\mathbf{g}^w + \mathbf{a}^b(t), \tag{6.6}$$

$$\dot{\mathbf{B}}^b(t) = -[\boldsymbol{\omega}^b(t)]_\times \mathbf{B}^b(t) + \nabla\mathbf{B}^b(t)\mathbf{v}^b(t). \tag{6.7}$$

This model relies on the following assumptions on the environment:

- **Flat-earth approximation.** We assume that the ENU (East-North-Up) earth frame at filter initialization is an inertial frame.
- **Stationary magnetic field** in the world frame—although possibly spatially non-uniform, leading to the spatial gradient in (6.7).

The first assumption is, in practice, often used in VINS literature, in which gyroscopes are not precise enough to measure earth rotational speed (roughly 7×10^{-5} rad/s). However, if high-end gyroscopes had been used, it is likely that an estimate based on the simple model (6.4)–(6.7) would introduce some error, confusing earth rotational velocity with biases estimates. In our opinion though, even in the case of high-end hardware, it is not clear that the visual pipeline used in this work would be able to provide information reliable enough to estimate biases with sufficient accuracy to be influenced by earth rotational speed. This would be a great challenge to address.

Note also that the world frame differs from the ENU frame defined at filter initialization: they both have their origin at the position of the center of the body frame at initial time, but they can differ by a rotation around the gravity direction, as the heading is not observable at initialization.

Equation (6.7) is the key equation for MI-DR. It relates the evolution of the magnetic field with kinematics quantities and local magnetic gradient. It actually renders the body velocity observable provided the matrix $\nabla\mathbf{B}^b$ is invertible. However, it fails to give useful translational information if the magnetic field is uniform. This happens outdoor, where the magnetic field is uniformly equal to the earth magnetic field. In the latter situation, magnetic gradients can vanish a few meters away from a wall, ceiling or floor.

Also, the stationarity assumption can be challenged in some environments, or punctually if pieces of metal are moving in the vicinity of the magnetometers even if some nonstationarities can be modeled—as the case of power-line interference [Chesneau et al., 2016].

6.3.3 Model Discretization

The model will be used in a discrete extended Kalman filtering framework described in Section 6.4 below. We thus discretize it the following way:

$$\mathtt{R}^w_{k+1} = \mathtt{R}^w_k \widetilde{\Delta\mathtt{R}}_{kk+1}, \tag{6.8}$$

$$\mathbf{p}^w_{k+1} = \mathbf{p}^w_k + \mathtt{R}^w_k \mathbf{v}^b_k \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t_{ij}^2 + \mathtt{R}^w_k \widetilde{\Delta\mathbf{p}}_{kk+1}, \tag{6.9}$$

$$\mathbf{v}^b_{k+1} = \widetilde{\Delta\mathtt{R}}^{\mathsf{T}}_{kk+1} \left(\mathbf{v}^b_k + \mathtt{R}^{w\mathsf{T}}_k \mathbf{g}^w \Delta t + \widetilde{\Delta\mathbf{v}}_{kk+1} \right), \tag{6.10}$$

$$\mathbf{B}^b_{k+1} = \widetilde{\Delta\mathtt{R}}^{\mathsf{T}}_{kk+1} \left(\mathbf{B}^b_k + \widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{v};kk+1}\mathbf{v}^b_k + \widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{g};kk+1}\mathtt{R}^{w\mathsf{T}}_k \mathbf{g}^w + \widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{a};kk+1} \right). \tag{6.11}$$

where we introduced the following notation corresponding to continuous integrals:

$$\widetilde{\Delta\mathtt{R}}_{kk+1} \overset{\text{def}}{=} \Delta\mathtt{R}_k(t_{k+1}), \tag{6.12}$$

$$\widetilde{\Delta\mathbf{v}}_{kk+1} \overset{\text{def}}{=} \widetilde{\Delta\mathbf{v}}_k(t_{k+1}), \tag{6.13}$$

$$\widetilde{\Delta\mathbf{p}}_{kk+1} \overset{\text{def}}{=} \int_{t_k}^{t_{k+1}} \widetilde{\Delta\mathbf{v}}_k(\tau)\mathtt{d}\tau, \tag{6.14}$$

$$\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{v};kk+1} \overset{\text{def}}{=} \int_{t_k}^{t_{k+1}} \Delta\mathtt{R}_k(\tau)\nabla\mathbf{B}^b(\tau)\Delta\mathtt{R}_k(\tau)^{\mathsf{T}}\mathtt{d}\tau, \tag{6.15}$$

$$\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{g};kk+1} \overset{\text{def}}{=} \int_{t_k}^{t_{k+1}} \Delta\mathtt{R}_k(\tau)\nabla\mathbf{B}^b(\tau)\Delta\mathtt{R}_k(\tau)^{\mathsf{T}}\left[\tau - t_k\right]\mathtt{d}\tau, \tag{6.16}$$

$$\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{a};kk+1} \overset{\text{def}}{=} \int_{t_k}^{t_{k+1}} \Delta\mathtt{R}_k(\tau)\nabla\mathbf{B}^b(\tau)\Delta\mathtt{R}_k(\tau)^{\mathsf{T}}\Delta\tilde{\mathbf{v}}_k(\tau)\mathtt{d}\tau, \tag{6.17}$$

$$\text{with } \Delta\mathtt{R}_k(\tau) \overset{\text{def}}{=} \mathbf{I}_3 + \int_{t_k}^{\tau} \Delta\mathtt{R}_k(s)\left[\boldsymbol{\omega}^b(s)\right]_\times \mathtt{d}s \quad \text{and} \quad \widetilde{\Delta\mathbf{v}}_k(\tau) \overset{\text{def}}{=} \int_{t_k}^{\tau} \Delta\mathtt{R}_k(s)\mathbf{a}^b(s)\mathtt{d}s. \tag{6.18}$$

The notation $\Delta\mathtt{R}_k(\tau)$ is the rotation matrix that transforms point in body frame at time τ to the corresponding point in body frame at time t_k, that can be deduced from gyroscopes integration.

These integrals can be computed from unbiased MIMU measurements. We thus estimate the biases of the accelerometers and gyrometers along with previously quantities. These are assumed to follow a first-order Gauss-Markov stochastic evolution:

$$\dot{\mathbf{b}}_g(t) = -\frac{1}{\tau_g}\mathbf{b}_g + \eta_{bg}, \tag{6.19}$$

$$\dot{\mathbf{b}}_a(t) = -\frac{1}{\tau_a}\mathbf{b}_a + \eta_{ba}. \tag{6.20}$$

where generating noises η_{bg} and η_{ba} satisfy:

$$\mathbb{E}\left([\eta_{bg}(t), \eta_{ba}(t)]\right) = \mathbf{0}_{6\times 1}, \tag{6.21}$$

$$\mathbb{E}\left([\eta_{bg}(t_1), \eta_{ba}(t_1)].[\eta_{bg}(t_2), \eta_{ba}(t_2)]^{\mathsf{T}}\right) = \boldsymbol{\mathcal{W}}_c \delta(t_2 - t_1), \tag{6.22}$$

$$\boldsymbol{\mathcal{W}}_c = \mathrm{diag}(\sigma^2_{bg;c}\mathbf{I}_3, \sigma^2_{ba;c}\mathbf{I}_3). \tag{6.23}$$

τ_b, τ_a, $\sigma_{bg;c}$ and $\sigma_{ba;c}$ are expressed in s, s, $\frac{\mathrm{rad}}{s^2}\frac{1}{\sqrt{Hz}}$ and $\frac{\mathrm{m}}{s^3}\frac{1}{\sqrt{Hz}}$ respectively and are characteristics of the IMU. Discretization of the evolution of biases leads to:

$$\mathbf{b}_{g\,k+1} = -\exp\left(\tfrac{\Delta t_{ij}}{\tau_g}\right)\mathbf{b}_{g\,k} + \boldsymbol{\eta}_{bg} \tag{6.24}$$

$$\mathbf{b}_{a\,k+1} = -\exp\left(\tfrac{\Delta t_{ij}}{\tau_a}\right)\mathbf{b}_{a\,k} + \boldsymbol{\eta}_{ba} \tag{6.25}$$

The $\boldsymbol{\eta}_{\mathbf{x}}$ appearing in (6.24) and (6.25) will be then modeled as discrete random variables with Gaussian density $\mathcal{N}(0, \boldsymbol{W})$, $\boldsymbol{W}$ being computed as $(t_{k+1} - t_k)\boldsymbol{\mathcal{W}}_c$ [Simon, 2010] (p. 231).

The presented models discretization exhibits integrals that all have to be computed numerically in order to build the filter on. Normally, the choice of the numerical integration method depends on the required accuracy. In the current implementation though, we take a very simple approach: we assume that the biases, acceleration and magnetic field gradient—the last two are in world frame—are constant between two MIMU sample and use the following identity:

$$\widetilde{\Delta\mathtt{R}}_{kk+1} \simeq \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}^b_k - \mathbf{b}_{g\,k}\right)\Delta t_k\right)\ , \tag{6.26}$$

$$\widetilde{\Delta\mathbf{v}}_{kk+1} \simeq \left(\tilde{\mathbf{a}}^b_k - \mathbf{b}_{a\,k}\right)\Delta t_k\ , \tag{6.27}$$

$$\widetilde{\Delta\mathbf{p}}_{kk+1} \simeq \frac{1}{2}\left(\tilde{\mathbf{a}}^b_k - \mathbf{b}_{a\,k}\right)\Delta t_k^2\ , \tag{6.28}$$

$$\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{v};kk+1} \simeq \nabla\tilde{\mathbf{B}}^b_k \Delta t_k\ , \tag{6.29}$$

$$\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{g};kk+1} \simeq \frac{1}{2}\nabla\tilde{\mathbf{B}}^b_k \Delta t_k^2\ , \tag{6.30}$$

$$\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{a};kk+1} \simeq \nabla\tilde{\mathbf{B}}^b_k \frac{1}{2}\left(\tilde{\mathbf{a}}^b_k - \mathbf{b}_{a\,k}\right)\Delta t_k^2. \tag{6.31}$$

The error induced by this simple integration scheme is limited by the relatively high frequency of MIMU sample (325 Hz).

6.3.4 Sensors Error Model

Sensors errors propagate through the discrete model when used for filtering. We assume that the noisy sensor reading at time k (the *input vector* of the filter) can be written:

$$\tilde{\mathbf{u}}_k = \underbrace{\begin{bmatrix}\tilde{\mathbf{a}}^b_k \\ \tilde{\boldsymbol{\omega}}^b_k \\ \mathrm{Vec}\left(\nabla\tilde{\mathbf{B}}^b_k\right)\end{bmatrix}}_{\text{measurement}} = \underbrace{\begin{bmatrix}\mathbf{a}_k \\ \boldsymbol{\omega}_k \\ \mathrm{Vec}\left(\nabla\mathbf{B}_k\right)\end{bmatrix}}_{\text{real values}} + \begin{bmatrix}\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 5} \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 5} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathcal{P}_{\nabla\mathbf{B}}\end{bmatrix} \underbrace{\begin{bmatrix}\boldsymbol{\eta}_{\mathbf{a}^b_k} \\ \boldsymbol{\eta}_{\boldsymbol{\omega}^b_k} \\ \boldsymbol{\eta}_{\nabla\mathbf{B}^b_k}\end{bmatrix}}_{\text{input noise } \delta\mathbf{u}_k \in \mathbb{R}^{11}} \tag{6.32}$$

The noise $\delta\mathbf{u}_k$ is assumed to be Gaussian $\delta\mathbf{u}_k \propto \mathcal{N}(0, \boldsymbol{\Sigma}_{\mathbf{u}})$ and is a sensor characteristic. We use here:

$$\boldsymbol{\Sigma}_{\mathbf{u}} = \begin{bmatrix}\sigma^2_{\boldsymbol{\omega}}\mathbf{I}_3 & 0_{3\times 3} & 0_{3\times 5} \\ 0_{3\times 3} & \sigma^2_{\mathbf{a}}\mathbf{I}_3 & 0_{3\times 5} \\ 0_{5\times 3} & 0_{5\times 3} & \boldsymbol{\Sigma}_{\nabla\mathbf{B}}\end{bmatrix} \text{ with } \sigma_{\boldsymbol{\omega}} \text{ in } \frac{rad}{s},\ \sigma_{\mathbf{a}} \text{ in } \frac{m}{s^2} \text{ and } \boldsymbol{\Sigma}_{\nabla\mathbf{B}} \text{ in } \frac{\text{Gauss}}{m} \in \mathbb{R}^{5\times 5}. \tag{6.33}$$

Note that $\mathcal{P}_{\nabla\mathbf{B}}$ reflects that we explicitly exploit the symmetry in the magnetic field from Maxwell equation. They implies that the gradient should be a symmetric matrix and of zero trace and thus

has 5 degrees of freedom, instead of the nine coefficient of the full matrix. $\mathcal{P}_{\nabla \mathbf{B}} \in \mathbb{R}^{9\times 5}$ is thus the matrix of the application generating the vectorized gradient matrix from minimal gradient coordinates:

$$\begin{array}{ccc} \mathbb{R}^5 & \rightarrow & \mathbb{R}^9 \\ \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{pmatrix} & \mapsto & \mathrm{Vec}\left(\begin{pmatrix} g_1 & g_2 & g_3 \\ g_2 & g_4 & g_5 \\ g_3 & g_5 & -g_1-g_4 \end{pmatrix}\right) \end{array} \tag{6.34}$$

6.4 Tight Fusion Filter

This section describe thoroughly the MI-MSCKF filter that is used in the experimental Section 6.5. Section 6.4.1 describes the parametrization of the state of the filter. Section 6.4.2 is dedicated to the propagation step while Section 6.4.3 details the magnetic and visual measurement equations.

6.4.1 State and Error State

We define the state space at time index k, $\mathcal{X}_k$ as a manifold which is compound of :

- the *keyframe poses* state space $\mathcal{K}_k$, which elements are the poses of a set of N past frames at time indexes $\{i_1, ...i_N\}$ not necessarily temporally successive but close in time. With poses written $\boldsymbol{\xi}_i = (\mathtt{R}_i^w, \mathbf{p}_i^w)$, this part of the state have the following form:

$$\left(\boldsymbol{\xi}_{i_1}, \cdots, \boldsymbol{\xi}_{i_N}\right) \in \mathcal{K}_k = \left(\mathrm{SO}(3) \times \mathbb{R}^3\right)^N \tag{6.35}$$

- the *current mimu state* $\mathcal{S}$ space which elements have the following form:

$$\boldsymbol{s}_k = \left(\boldsymbol{\xi}_k, \mathbf{v}_k^b, \mathbf{B}_k^b, \mathbf{b}_{a\,k}, \mathbf{b}_{g\,k}\right)^\mathsf{T} \in \mathrm{SO}(3) \times \mathbb{R}^3 \times \mathbb{R}^{12} \tag{6.36}$$

A complete state space element at time index k is thus noted:

$$\mathbf{X}_k = \left(\boldsymbol{\xi}_{i_1}, \cdots, \boldsymbol{\xi}_{i_N}, \boldsymbol{s}_k\right) \in \mathcal{X}_k = \mathcal{K}_k \times \mathcal{S} \tag{6.37}$$

We use the Lie group structure of this manifold and its tangent space to (i) define the error tracked by the filter and (ii) define the Jacobian of the measurement process. This derives from the fact that a perturbation around an element can be expressed as an element of its Lie algebra. We use the $\boxplus$ operator symbol, so that $\mathbf{X}_k \boxplus \delta\boldsymbol{X}_k$ computes a new state in $\mathcal{X}_k$ from a tangent perturbation $\delta\boldsymbol{X}_k$ around $\mathbf{X}_k$. We define it as regular addition operation for all components of the state except for pose states where we use:

$$\boldsymbol{\xi} \boxplus \delta\boldsymbol{\xi} = \left(\mathrm{Exp}\left(\delta\boldsymbol{\xi}_{4:6}\right)\mathtt{R}, \mathbf{p} + \delta\boldsymbol{\xi}_{1:3}\right) \tag{6.38}$$

Similarly we define the reciprocal operator $\boxminus$ as the binary operator giving the perturbation element between two states of $\mathcal{X}$. It is defined as regular minus operation except for pose states for which it is:

$$\boldsymbol{\xi}_2 \boxminus \boldsymbol{\xi}_1 = \left[\mathrm{Log}(\mathtt{R}_2\mathtt{R}_1^{-1})^T, \mathbf{p}_2^\mathsf{T} - \mathbf{p}_1^\mathsf{T}\right]^\mathsf{T} \tag{6.39}$$

We here define the error state as the application $\boxminus$ operator between the true state and the estimated state, noted hereafter with an hat. It is thus an element of the tangent space at the current estimate.

$$\mathbf{e}_k \overset{\text{def}}{=} \mathbf{X}_k \boxminus \hat{\mathbf{X}}_k \Leftrightarrow \mathbf{X}_k = \hat{\mathbf{X}}_k \boxplus \mathbf{e}_k \tag{6.40}$$

Note that this implies a parametrization of the rotation error in *world* frame, and is different from our previous work [Caruso et al., 2017a] where rotation error was parametrized in the body frame.

The filtering process propagates the estimated mean, along with an estimate of uncertainty. This uncertainty is represented as a Gaussian density of the error state $\mathbf{e}_k \propto \mathcal{N}(0, \mathbf{P}_k)$ in order to take advantage of the Lie group structure defined previously, i.e., a minimal parametrization and a locally Euclidean structure in tangent space.

For numerical reasons, the covariance $\mathbf{P}_k$ will be tracked by the filter in an square-root information form, such that we have the relationship $\mathbf{P}_k = (\hat{\boldsymbol{\mathcal{S}}}_k^\mathsf{T} \hat{\boldsymbol{\mathcal{S}}}_k)^{-1}$ with $\hat{\boldsymbol{\mathcal{S}}}_k$ an upper triangular matrix. The next section describes how this quantity evolves through the different steps of the filter: propagation and update.

6.4.2 Propagation/Augmentation/Marginalization

At the time of arrival of a new IMU data $k+1$, $\mathbf{X}_{k|k}$ and $\hat{\boldsymbol{\mathcal{S}}}_{k|k}$ are propagated. This is summarized by Figure 6.3 and splitted in three steps. First, the mimu state $\boldsymbol{s}_k$ is propagated with discretized model (Section 6.4.2.1), then the full state is augmented with the resulting new mimu state $\boldsymbol{s}_{k+1}$ (Section 6.4.2.2). Finally, some part of this augmented state are marginalized before the update step (Section 6.4.2.3).

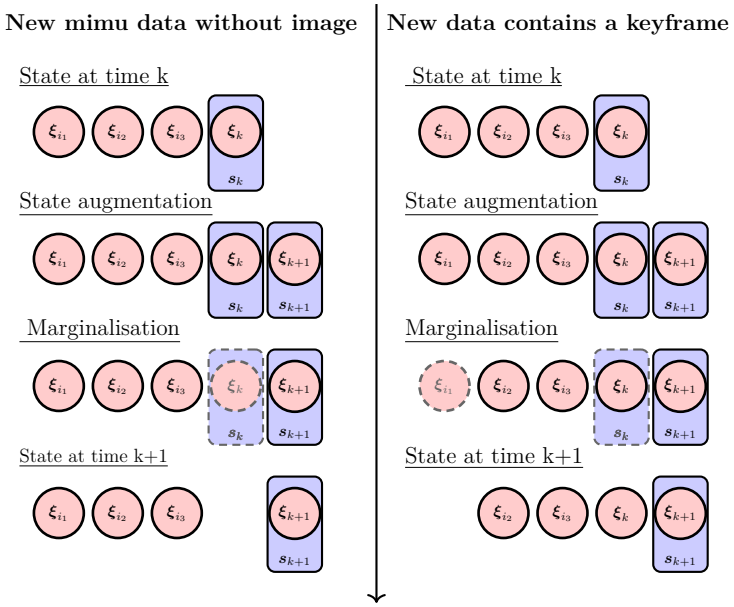


Figure 6.3: Illustration of state augmentation and marginalization across time as described in Section 6.4.2.

6.4.2.1 Propagation

Keyframe poses states are estimation of physical quantities blocked at fixed instant in time, their error do not evolve with time and thus they are propagated with an identity function. Besides, the current mimu state error is propagated according to

$$\boldsymbol{s}_{k+1} = \mathbf{f}_{\text{mimu}}(\boldsymbol{s}_k, \tilde{\mathbf{u}}_k, \boldsymbol{\eta}_k) \tag{6.41}$$

where the *discrete mimu process function* $\mathbf{f}_{\text{mimu}}$ summarizes (6.8)–(6.11) and (6.24)–(6.25).

The mimu state error is increased from the three sources of uncertainty: the stochastic model of biases, the measurement noise $\delta\mathbf{u}_k$ on the input vector and the uncertainty of the previous estimate:

$$\mathbf{e}_{\text{mimu},k+1} = \mathbf{f}_{\text{mimu}}(\hat{\boldsymbol{s}}_k \boxplus \mathbf{e}_k^{\text{mimu}}, \tilde{\mathbf{u}}_k - \delta\mathbf{u}_k, \boldsymbol{\eta}_k) \boxminus \mathbf{f}_{\text{mimu}}(\hat{\boldsymbol{s}}_k, \tilde{\mathbf{u}}_k, 0) \tag{6.42}$$

$$\simeq \boldsymbol{\Phi}_{k+1,k}^{\text{mimu}} \mathbf{e}_k^{\text{mimu}} + \mathbf{G}_{k+1,k}^{\text{mimu}} \delta\mathbf{u}_k + \mathbf{C}_{k+1,k}^{\text{mimu}} \boldsymbol{\eta}_k + O\left(\mathbf{e}_k^{\text{mimu}}, \delta\mathbf{u}_k, \boldsymbol{\eta}_k\right) \tag{6.43}$$

The expression of matrices $\boldsymbol{\Phi}_{k+1,k}^{\text{mimu}}$, $\mathbf{G}_{k+1,k}^{\text{mimu}}$, $\mathbf{C}_{k+1,k}^{\text{mimu}}$ are derived by 1st order development of (6.42):

$$\boldsymbol{\Phi}_{k+1,k}^{\text{mimu}} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{R}\mathbf{b}_g}} & \mathbf{0}_3 \\ \left[-\mathtt{R}_k(\mathbf{v}_k^b \Delta t + \widetilde{\Delta\mathbf{p}}_{kk+1})\right]_\times & \mathbf{I}_3 & \mathtt{R}_k \Delta t & \mathbf{0}_3 & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_g}} & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_g}} \\ \mathtt{R}_{k+1}^\mathsf{T} [\mathbf{g}^w]_\times \Delta t & \mathbf{0}_3 & \Delta\mathtt{R} & \mathbf{0}_3 & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_g}} & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_a}} \\ \Delta\mathtt{R}^\mathsf{T} \widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{g};kk+1} [\mathbf{g}^w]_\times & \mathbf{0}_3 & \Delta\mathtt{R}^\mathsf{T} \widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{v};kk+1} & \Delta\mathtt{R}^\mathsf{T} & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_g}} & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_a}} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\exp\left(\frac{\Delta t}{\tau_g}\right) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\exp\left(\frac{\Delta t}{\tau_a}\right) \end{bmatrix} \tag{6.44}$$

$$\mathbf{G}_{k+1,k}^{\text{mimu}} = \begin{bmatrix} -\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{R}\mathbf{b}_g}} & 0 & 0 \\ -\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_g}} & -\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_a}} & 0 \\ -\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_g}} & -\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_a}} & 0 \\ -\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_g}} & -\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_a}} & \mathbf{G}^{\text{mimu}}_{k+1,k_{\mathbf{B}\nabla\mathbf{B}}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{C}_{k+1,k}^{\text{mimu}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \mathbf{I}_3 & 0 \\ 0 & \mathbf{I}_3 \end{bmatrix} \tag{6.45}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{R}\mathbf{b}_g}} = -\mathtt{R}_k \Delta t \tag{6.46}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_g}} = -\Delta t\, [\mathbf{v}_{k+1}]_\times + \partial_{\mathbf{b}_{g_k}} \left(\widetilde{\Delta\mathbf{v}}_{kk+1}\right) \tag{6.47}$$

$$\begin{aligned} \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_g}} &= -\Delta t\, [\mathbf{B}_{k+1}]_\times \\ &\quad + \Delta\mathtt{R}^\mathsf{T} \left(\mathbf{v}_k^\mathsf{T} \otimes \mathbf{I}_3\right) \partial_{\mathbf{b}_g} \left(\mathrm{Vec}\left(\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{v};kk+1}\right)\right) \\ &\quad + \Delta\mathtt{R}^\mathsf{T} \left(\mathbf{g}^\mathsf{T}\mathtt{R}_k \otimes \mathbf{I}_3\right) \partial_{\mathbf{b}_g} \left(\mathrm{Vec}\left(\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{g};kk+1}\right)\right) \\ &\quad + \Delta\mathtt{R}^\mathsf{T} \partial_{\mathbf{b}_g} \left(\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{a};kk+1}\right) \end{aligned} \tag{6.48, 6.49}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_g}} = \mathtt{R}_k \partial_{\mathbf{b}_{g_k}} \left(\widetilde{\Delta\mathbf{p}}_{kk+1}\right) \tag{6.50}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_a}} = \Delta\mathtt{R}^\mathsf{T} \partial_{\mathbf{b}_{a_k}} \left(\widetilde{\Delta\mathbf{v}}_{kk+1}\right) \tag{6.51}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_a}} = \mathtt{R}_k \partial_{\mathbf{b}_{a_k}} \left(\widetilde{\Delta\mathbf{p}}_{kk+1}\right) \tag{6.52}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_a}} = \Delta\mathtt{R}^\mathsf{T} \partial_{\mathbf{b}_{a_k}} \left(\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{a};kk+1}\right) \tag{6.53}$$

$$\begin{aligned} \mathbf{G}^{\text{mimu}}_{k+1,k_{\mathbf{B}\nabla\mathbf{B}}} &= \Delta\mathtt{R}^\mathsf{T} \left(\mathbf{v}_k^\mathsf{T} \otimes \mathbf{I}_3\right) \partial_{\nabla\mathbf{B}} \left(\mathrm{Vec}\left(\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{v};kk+1}\right)\right) \\ &\quad + \Delta\mathtt{R}^\mathsf{T} \left(\mathbf{g}^\mathsf{T}\mathtt{R}_k \otimes \mathbf{I}_3\right) \partial_{\nabla\mathbf{B}} \left(\mathrm{Vec}\left(\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{g};kk+1}\right)\right) \mathcal{P}_{\nabla\mathbf{B}} + \Delta\mathtt{R}^\mathsf{T} \partial_{\nabla\mathbf{B}} \left(\widetilde{\boldsymbol{\Delta}\mathbf{B}}_{\mathbf{a};kk+1}\right) \mathcal{P}_{\nabla\mathbf{B}} \end{aligned} \tag{6.54}$$

where $\mathcal{P}_{\nabla\mathbf{B}}$ is defined as in (6.34). Note that we wrote here the transition and noise matrices as a function of the integrals (6.12)–(6.17), independently of the way the are computed, so that these

expressions are still valid if one choose to compute the integrals with a more sophisticated scheme. Derivative of these integrals with respect to the biases are also required: these quantities should be computed simultaneously with the integrals. In our implementation, they are computed easily with the approximations made in (6.26)–(6.31).

6.4.2.2 State Augmentation

When MIMU sample $k+1$ arrives, the mimu propagation function is used to augment the state with the new current mimu state $\hat{\boldsymbol{s}}_{k+1} = \mathbf{f}_{\text{mimu}}(\hat{\boldsymbol{s}}_k, \tilde{\mathbf{u}}_k, 0)$, leading to the augmented state $\hat{\mathbf{X}}^{\oplus}_{k+1|k}$. The error state square root information matrix is augmented accordingly to [Wu et al., 2015], $\hat{\boldsymbol{\mathcal{S}}}^{\oplus}_{k+1|k}$ using the Jacobian derived in previous subsection.

$$\hat{\mathbf{X}}^{\oplus}_{k+1|k} \overset{\text{def}}{=} \left(\mathbf{X}_{k|k}, \boldsymbol{s}_{k+1}\right) \tag{6.55}$$

$$\hat{\boldsymbol{\mathcal{S}}}^{\oplus}_{k+1|k} \overset{\text{def}}{=} \begin{bmatrix} \hat{\boldsymbol{\mathcal{S}}}_{k|k} & 0 \\ \boldsymbol{V_1} & \boldsymbol{V_2} \end{bmatrix} \tag{6.56}$$

with

$$\boldsymbol{V_1} = [\mathbf{0}_{18\times 6}, \cdots, \mathbf{0}_{18\times 6}, \boldsymbol{Q}^{-\frac{1}{2}}_{k+1} \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k\,k+1,k}] \tag{6.57}$$

$$\boldsymbol{V_2} = \boldsymbol{Q}^{-\frac{1}{2}}_{k+1} \tag{6.58}$$

and the discrete model noise at $\boldsymbol{Q}_k$ time k :

$$\boldsymbol{Q}_{k+1} = \left(\mathbf{G}^{\text{mimu}}_{k+1,k} \mathbf{C}^{\text{mimu}}_{k+1,k}\right) \begin{pmatrix} \boldsymbol{\Sigma_u} & 0 \\ 0 & \boldsymbol{W} \end{pmatrix} \begin{pmatrix} \mathbf{G}^{\text{mimu}\,T}_{k+1,k} \\ \mathbf{C}^{\text{mimu}\,T}_{k+1,k} \end{pmatrix}. \tag{6.59}$$

6.4.2.3 Marginalization of Old State

Then, in order to bound the size of $\hat{\mathbf{X}}_{k+1|k}$, some part of it are marginalized. The state elements to be marginalized depend on the type of data available at the current timestamps as depicted in Figure 6.3. If only MIMU data (without a new keyframe image attached) are arriving at time k, $\boldsymbol{s}_k$ is marginalized. If MIMU data and a new keyframe image are available at time k, the oldest keyframe pose is marginalized together with the non-pose element of $\boldsymbol{s}_k$. Note that since the image frame rate is well below MIMU frequency, the first case happens more often than the second case.

Within the square root information form, marginalization is done similarly to [Wu et al., 2015]. With $\boldsymbol{\Pi}_k$ being the matrix permutation putting the *to marginalize* error states at the beginning, a QR decomposition of a square root information matrix of the permuted augmented error state vector writes:

$$\hat{\boldsymbol{\mathcal{S}}}^{\oplus}_{k+1|k} \boldsymbol{\Pi}_k = \mathbf{O}_p \boldsymbol{R}_p, \quad \begin{matrix} \mathbf{O}_p \in \boldsymbol{\mathcal{O}}(n) \\ \boldsymbol{R}_p \in \mathbb{R}^n \end{matrix} \tag{6.60}$$

$$= \mathbf{O}_p \begin{bmatrix} * & * \\ 0 & \hat{\boldsymbol{\mathcal{S}}}_{k+1|k} \end{bmatrix} \tag{6.61}$$

We obtain the predicted state $\hat{\mathbf{X}}_{k+1|k}$ removing marginalized states, and its—upper triangular—square-root information matrix $\hat{\boldsymbol{\mathcal{S}}}_{k+1|k}$. If no measurement update has to be performed at current time step they can be used directly as $\mathbf{X}_{k+1|k+1}$ and $\hat{\boldsymbol{\mathcal{S}}}_{k+1|k+1}$ for the next propagation.[1]

[1] The marginalization of a joint Gaussian distribution with a square-root information form is not often demonstrated in book or lecture but can be deduced from the full information form $\boldsymbol{\Lambda}_{\text{joint}}$:

$$\text{if } \boldsymbol{\Lambda}_{\text{joint}} = \begin{bmatrix} \boldsymbol{\Lambda}_M & \boldsymbol{\Lambda}_{MR} \\ \boldsymbol{\Lambda}^{\mathsf{T}}_{MR} & \boldsymbol{\Lambda}_R \end{bmatrix} = \hat{\boldsymbol{\mathcal{S}}}^{\mathsf{T}}_{\text{joint}} \hat{\boldsymbol{\mathcal{S}}}_{\text{joint}} = \begin{bmatrix} \hat{\boldsymbol{\mathcal{S}}}^{\mathsf{T}}_M \hat{\boldsymbol{\mathcal{S}}}_M & \hat{\boldsymbol{\mathcal{S}}}^{\mathsf{T}}_M \hat{\boldsymbol{\mathcal{S}}}_{MR} \\ \hat{\boldsymbol{\mathcal{S}}}^{\mathsf{T}}_{MR} \hat{\boldsymbol{\mathcal{S}}}_M & \hat{\boldsymbol{\mathcal{S}}}^{\mathsf{T}}_R \hat{\boldsymbol{\mathcal{S}}}_R + \hat{\boldsymbol{\mathcal{S}}}^{\mathsf{T}}_{MR} \hat{\boldsymbol{\mathcal{S}}}_{MR} \end{bmatrix} \quad \text{with } \hat{\boldsymbol{\mathcal{S}}}_{\text{joint}} = \begin{bmatrix} \hat{\boldsymbol{\mathcal{S}}}_M & \hat{\boldsymbol{\mathcal{S}}}_{MR} \\ 0 & \hat{\boldsymbol{\mathcal{S}}}_R \end{bmatrix} \tag{6.62}$$

then the square root information matrix resulting from marginalization of the the M variables is: $\hat{\boldsymbol{\mathcal{S}}}^{\text{marg}}_R = \hat{\boldsymbol{\mathcal{S}}}_R$. This is deduced by calculus from the usually demonstrated fact that: $\boldsymbol{\Lambda}^{\text{marg}}_R = \boldsymbol{\Lambda}_R - \boldsymbol{\Lambda}_{MR} \boldsymbol{\Lambda}^{-1}_M \boldsymbol{\Lambda}^{\mathsf{T}}_{RM}$)

6.4.3 Measurement Update

The filter processes two kinds of measurement: (i) the magnetic one that compares the magnetic field measured at the current timestamp with the magnetic field predicted by the filter; (ii) the visual measurement equation for features for which tracking has just ended.

We first briefly recall the update process of the inverse square root filter on a manifold. Let us suppose that some measurement occurs that can be modeled as:

$$\mathbf{h}(\mathbf{X}_{k+1}) = \tilde{\mathbf{h}}_{k+1} + \boldsymbol{\eta}_{\mathbf{h}} \in \mathbb{R}^{n_m},\ \boldsymbol{\eta}_{\mathbf{h}} \sim \mathcal{N}(0, \Sigma_{\mathbf{h}}) \tag{6.63}$$

with n_m the dimension of the measurement vector. Writing the dimension of the predicted state as n_s, the measurement error $\mathbf{z}_{k+1} = \mathbf{h}(\hat{\mathbf{X}}_{k+1|k}) - \tilde{\mathbf{h}}_{k+1}$ and $\boldsymbol{H}$ the jacobian of the application:

$$\begin{array}{ccc} \mathbb{R}^{n_s} & \rightarrow & \mathbb{R}^{n_m} \\ \delta\boldsymbol{X} & \mapsto & \mathbf{h}(\hat{\mathbf{X}}_{k+1|k} \boxplus \delta\boldsymbol{X}) \end{array}, \tag{6.64}$$

the update step finds the tangent correction $\delta\mathbf{X}^*$ that minimizes the following linearized cost:

$$\begin{aligned} \mathcal{C}(\delta\mathbf{X}) &= \|\delta\mathbf{X}\|^2_{\mathbf{P}_{\mathbf{k+1|k}}} + \|\boldsymbol{H}\delta\mathbf{X} - \mathbf{z}_{k+1}\|^2_{\boldsymbol{\Sigma}_h} \\ &= \left\|\hat{\boldsymbol{\mathcal{S}}}_{k+1|k}.\delta\mathbf{X}\right\|^2 + \left\|\boldsymbol{\Sigma}_{\mathbf{h}}^{-\frac{1}{2}}\left(\boldsymbol{H}\delta\mathbf{X} - \mathbf{z}_{k+1}\right)\right\|^2 \\ &= \left\| \begin{bmatrix} \hat{\boldsymbol{\mathcal{S}}}_{k+1|k} \\ \boldsymbol{\Sigma}_{\mathbf{h}}^{-\frac{1}{2}}\boldsymbol{H} \end{bmatrix} \delta\mathbf{X} - \begin{bmatrix} 0 \\ \boldsymbol{\Sigma}_{\mathbf{h}}^{-\frac{1}{2}}\mathbf{z}_{k+1} \end{bmatrix} \right\|^2. \end{aligned} \tag{6.65}$$

The optimum point can be obtained by of a thin QR decomposition:

$$\begin{bmatrix} \hat{\boldsymbol{\mathcal{S}}}_{k+1|k} \\ \boldsymbol{\Sigma}_h^{-\frac{1}{2}}\boldsymbol{H} \end{bmatrix} = \mathbf{O}_u \boldsymbol{R}_{\boldsymbol{u}}, \quad \begin{array}{l} \mathbf{O}_u \in \boldsymbol{\mathcal{O}}(n_s + n_m) \\ \boldsymbol{R}_{\boldsymbol{u}} \in \mathbb{R}^{n_s+n_m} \end{array} \tag{6.66}$$

$$\mathcal{C}(\delta\mathbf{X}) = \left\| \boldsymbol{R}_{\boldsymbol{u}}\delta\mathbf{X} - \mathbf{O}_u^{\mathsf{T}} \begin{bmatrix} 0 \\ \boldsymbol{\Sigma}_{\mathbf{h}}^{-\frac{1}{2}}\mathbf{z}_{k+1} \end{bmatrix} \right\|^2, \tag{6.67}$$

$\boldsymbol{R}_{\boldsymbol{u}}$ being upper triangular, $\delta\mathbf{X}^*$ is efficiently computed by back-substitution. This optimal correction is finally applied to the predicted state with the retraction operator :

$$\mathbf{X}_{k+1|k+1} = \hat{\mathbf{X}}_{k+1|k} \boxplus \delta\mathbf{X}^* \tag{6.68}$$

$$\hat{\boldsymbol{\mathcal{S}}}_{k+1|k} = \boldsymbol{R}_{\boldsymbol{u}}\mathtt{J}_r^{-1}. \tag{6.69}$$

With $\mathtt{J}_r$ being the Jacobian at e = 0 of:

$$\begin{aligned} &\mathbb{R}^{n_s} \rightarrow \mathbb{R}^{n_s} \\ &\quad e \mapsto (\hat{\mathbf{X}}_{k+1|k} \boxplus (\delta\mathbf{X}^* + e)) \boxminus (\hat{\mathbf{X}}_{k+1|k} \boxplus \delta\mathbf{X}^*). \end{aligned} \tag{6.70}$$

Intuitively, this Jacobian transforms the square-root information matrix from the tangent space of predicted state to the tangent space of the updated state. Note that, in the current parametrization choice (cf. (6.38)), this Jacobian is the identity matrix; this was not the case in our previous work [Caruso et al., 2017a] where the rotation error was expressed in body frame.

6.4.3.1 Magnetic Measurement Update

The magnetic measurement update is the simplest and uses at MIMU frequency the direct magnetic field measurement which writes:

$$h_{\mathbf{B}}(\mathbf{X}_k) = \boldsymbol{P}_{\mathbf{B}_b}\mathbf{X}_k \tag{6.71}$$

$$\Sigma_{h_{\mathbf{B}}} = \sigma^2_{\mathbf{B}}\mathbf{I}_3 \tag{6.72}$$

with $\boldsymbol{P}_{\mathbf{B}_b}$ the projection operator from the state space to the coordinates of the state corresponding to $\mathbf{B}_b$. $\sigma_{\mathbf{B}}$ is the noise of the magnetometers reading.

6.4.3.2 Opportunistic Feature Tracks Measurement Update

We use the feature tracks in the same way as proposed by [Mourikis and Roumeliotis, 2007]. We derive here the equation for completeness.

When a feature track ends or when the frame in which the feature was detected is about to be marginalized, we process the entire feature track as a measurement constraining the pose of each in-state frame where the feature was detected.

The predicted reprojection of feature i in frame at time t is written:

$$\mathbf{pr}^i_t(\boldsymbol{\xi}_t, \mathbf{l}^w_i) = \pi\left(\mathtt{R}^{\mathrm{b}}_{\mathrm{c}}{}^{\mathsf{T}}\left(\mathtt{R}^{w}_{t}{}^{\mathsf{T}}\left(\mathbf{l}^w_i - \mathbf{p}^w_t\right) - \mathbf{p}^{\mathrm{b}}_{\mathrm{c}}\right)\right) \in \mathbb{R}^2 \tag{6.73}$$

with $\mathbf{l}^w_i \in \mathbb{R}^3$ the 3D-position of the features i in inertial coordinates and $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ the projection function of the camera. We recall that $(\mathtt{R}^{\mathrm{b}}_{\mathrm{c}}, \mathbf{p}^{\mathrm{b}}_{\mathrm{c}})$ is the known transform between the body frame and the camera frame. $\mathbf{l}^w_i$ is computed by a fast triangulation function from known in-state poses and the measured observation, noted $\boldsymbol{o}_{it}$.

By stacking all these reprojections, we can write the non linear measurement function:

$$\mathbf{h}_{fi}(\mathbf{X}, \mathbf{l}^w_i) = \begin{bmatrix} \cdot \\ \mathbf{pr}^i_t(\boldsymbol{\xi}_t, \mathbf{l}^w_i) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \boldsymbol{o}_{it} \\ \cdot \end{bmatrix} + \boldsymbol{\eta}_{fi}\ , \tag{6.74}$$

with $\boldsymbol{\eta}_{fi}$ is assumed to be an additive Gaussian white noise : $\boldsymbol{\eta}_{fi} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_C)$, $\boldsymbol{\Sigma}_C = \sigma_c I_{2m}$. We note subsequently $\boldsymbol{o}_i$ the vector resulting from stacking of 2-vector observation $\boldsymbol{o}_{it}$.

The attentive reader may note that this measurement equation is not of the form of (6.63), because $\mathbf{l}^w_i$ is not part of our state: for this reason, we can not directly use (6.65) and (6.67). Worse, the computed $\mathbf{l}^w$ estimate is correlated with the current state error, so that we can not just use it as a fixed constant. The solution used here aims at expressing a projection of the residual that depends only on the poses, up to the first order.

We start by linearizing the residual then proceeds the minimization in two steps to extract the used measurement equation. Linearization of $\mathbf{r}_i : \mathbf{X}, \mathbf{l}^w_i \mapsto (\mathbf{h}_{fi}(\mathbf{X}, \mathbf{l}^w_i) - \boldsymbol{o}_{\mathbf{i}})$ yields:

$$\begin{aligned} &\mathbf{r}_i(\mathbf{X}_{k+1|k} \boxplus \delta\boldsymbol{X}, \mathbf{l}^w_i + \delta\mathbf{l}^w_i) \\ &\quad\simeq \boldsymbol{F}_i\delta\boldsymbol{X} + \boldsymbol{E}_f\delta\mathbf{l}^w_i + \left(\mathbf{h}_{fi}(\mathbf{X}_{k+1|k}, \mathbf{l}^w_i) - \boldsymbol{o}_i\right) \\ &\quad= \boldsymbol{F_k}\delta\boldsymbol{X} + \boldsymbol{E_f}\delta\mathbf{l}^w_i - \delta\boldsymbol{o}_i \end{aligned} \tag{6.75}$$

$\delta\boldsymbol{o}_i = \mathbf{h}_{fi}(\mathbf{X}_{k+1|k}, \mathbf{l}^w_i) - \boldsymbol{o}_i$ is the $2m$-vector of predicted residual error. $\boldsymbol{E_f}$ is a $2m \times 3$ matrix of rank 3 ; m denoting the number of observations for the feature. The rank of $\boldsymbol{E_f}$ is guaranteed during the triangulation. Its QR decomposition writes:

$$\boldsymbol{E_f} = [\mathbf{O}_{\boldsymbol{E}1}, \mathbf{O}_{\boldsymbol{E}0}] \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{E}1} \\ \mathbf{0}_{2m-3\times 2m} \end{bmatrix} \tag{6.76}$$

$$\mathbf{O}_{\boldsymbol{E}1} \in \mathbb{R}^{2m\times 3},\ \mathbf{O}_{\boldsymbol{E}0} \in \mathbb{R}^{2m\times 2m-3}, \boldsymbol{R}_{\boldsymbol{E}1} \in GL_3(\mathbb{R}); \tag{6.77}$$

Properties of square orthogonal matrices on the L_2 norm of vectors allow to split the cost function into two terms, one depending only on the current predicted state vector.

$$\begin{aligned}\min_{\delta\boldsymbol{X},\delta\mathbf{l}_i^w}\|\mathbf{r}_i\|^2{}_{\boldsymbol{\Sigma}_C} &= \min_{\delta\boldsymbol{X},\delta\mathbf{l}_i^w}\left\|\begin{bmatrix}\mathbf{O}_{\boldsymbol{E}1}^\top\\ \mathbf{O}_{\boldsymbol{E}0}^\top\end{bmatrix}\mathbf{r_i}\right\|^2_{\boldsymbol{\Sigma}_C}\\ &= \min_{\delta\boldsymbol{X},\delta\mathbf{l}_i^w}\left\|\mathbf{O}_{\boldsymbol{E}1}^\top\boldsymbol{F}_i\delta\boldsymbol{X}+\boldsymbol{R}_{\boldsymbol{E}1}\delta\mathbf{l}_i^w-\mathbf{O}_{\boldsymbol{E}1}^\top\delta\boldsymbol{o}_i\right\|^2_{\boldsymbol{\Sigma}_C}\\ &+\min_{\delta\boldsymbol{X}}\left\|\boldsymbol{Q}_{\boldsymbol{E}0}^\top\boldsymbol{F}_i\delta\boldsymbol{X}-\mathbf{O}_{\boldsymbol{E}0}^\top\delta\boldsymbol{o}_i\right\|^2_{\boldsymbol{\Sigma}_C}\end{aligned} \tag{6.78}$$

As $\boldsymbol{R}_{\boldsymbol{E}1}$ is invertible, the first term of the quantity to be minimized can be reduced to zero for all $\delta\boldsymbol{X}$. Minimization reduces thus to:

$$\min_{\delta\boldsymbol{X}}\left\|\mathbf{O}_{\boldsymbol{E}0}^\top\boldsymbol{F}_i\delta\boldsymbol{X}-\mathbf{O}_{\boldsymbol{E}0}^\top\delta\boldsymbol{o}_i\right\|^2_{\boldsymbol{\Sigma}_C} \tag{6.79}$$

Finally, the previous linearized residual is used in the cost function (6.65), i.e., we use for the $\boldsymbol{H}$ matrix, the $\mathbf{z}$ error vector and the covariance of measurement $\Sigma_{\mathbf{h}}$ the quantities:

$$\boldsymbol{H}_f=\mathbf{O}_{\boldsymbol{E}0}^\top\boldsymbol{F}_i,\qquad \mathbf{z}_f=\mathbf{O}_{\boldsymbol{E}0}^\top\delta\boldsymbol{o}_i,\qquad \boldsymbol{\Sigma}_{\mathbf{h}_f}=\sigma_C I_{2m-3}. \tag{6.80}$$

Note that everything happens here as if we introduced the features position into the state, but instantly marginalized it.

6.4.4 Filter Initialization

One sensitive issue in a VINS filter is initialization. It usually involves some specific algorithm as described in [Yang and Shen, 2016; Dong-Si and Mourikis, 2012] for instance. In our implementation, we proceed as follows: after switch on, the current state is initialized at the origin with an attitude matching the current acceleration direction and a zero speed, both with high variance. The filter is then run using the high-frequency magnetic update equation in order to get a stabilized trajectory rapidly from the first few seconds. This trajectory is used to bootstrap the first keyframe poses state and then to start using features information. This initialization process relies on the empirical observation that the MI-DR filter convergence basin is large. Admittedly, it would degrade severely the filter if the system's switch on occurs in an area where magneto-inertial dead-reckoning is not reliable.

6.5 Experimental Study

6.5.1 Hardware Prototype Description and Data Syncing

The sensor system is pictured on Figure 6.4. The camera is rigidly attached 47 cm away of the MIMU system to avoid any potential magnetic perturbation. Such a large distance is necessary because the off-the-shelf camera has not been specifically designed for reducing its magnetic footprint. Reducing the hardware to a wearable size would involve sensors co-design that was not in the scope of this work. For the vision part, we use an IDS uEye 3241-LE equipped with a Lensagon BM4018S118 lens. It provides around 100 degrees of field of view. Camera intrinsics and extrinsic parameters are calibrated with the Kalibr toolbox [Furgale et al., 2013]. The MIMU provides data at 325 Hz, the camera at 20 Hz. Magnetometers, accelerometers and gyro-meters are all MEMS sensors digitized with sigma-delta ADCs which were carefully calibrated.

The camera and MIMU provide timestamps computed from different clocks. We synchronize them offline: timestamps shifts are estimated both at the start and at the end of the records by the checkerboard-based Kalibr calibration toolbox; clock drift is then deduced and corrected for.

The camera exposure time is fixed at 10 milliseconds maximum, reducing worst-case motion blur and allowing to timestamp each image observation accordingly. However, this choice limits the camera's ability to adapt to a very low-light environment.

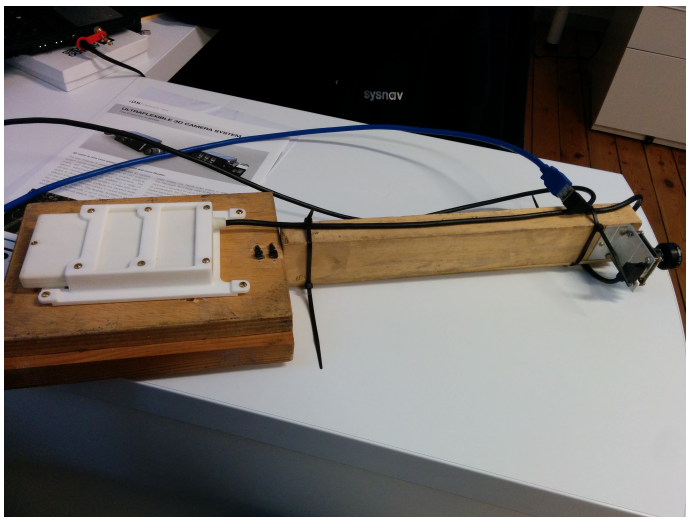


Figure 6.4: The sensor setup used in this work. The white box on the left side contains the MIMU sensor, the camera is on the right side. Both sensors are rigidly attached through a non-magnetic, non conductive material (wood).

6.5.2 Filter Parameters Tuning

Most parameters of the filter are chosen in a deterministic and consistent fashion. MIMU noise standard deviation $\sigma_{\mathbf{a}}, \sigma_{\boldsymbol{\omega}}, \sigma_{\mathbf{B}}$ and biases evolution parameters $\tau_{bg}, \tau_{ba}, \sigma_{bg;c}, \sigma_{ba;c}$ are derived from sensors characteristics measured empirically with an Allan standard deviation. The pixel reprojection noise σ_C is set to $\sqrt{2}$, which is the diagonal size of a pixel. Only the magnetic part of covariance $\mathbf{\Sigma}_{\nabla\mathbf{B}}$ is tuned empirically. It is set higher than what would derive from the covariance of magnetometers white noise, so as to absorb some modeling errors of the magnetic field, such as small non-stationarities or high values of the 2nd order spatial term. Note that parameter tuning is unchanged for all presented datasets in order to draw fair conclusions.

6.5.3 Visual Processing Implementation

The visual processing pipeline aims at constantly tracking 200 interest points well spread in the image. In order to enforce a proper repartition of corners across the entire image, we use a bucket strategy. Harris-corner response is computed over the whole image, and we retain only the strongest features in buckets for which the number of already tracked points is below a threshold. We use here a partition of the image in 6×8 buckets. Detected corners are tracked from frame to frame with OpenCV pyramidal KLT algorithm until either:

- they go out of the field of view;
- the tracking fails;
- they are classified as outliers;
- the frame where they were first detected is to be marginalized at next propagation step.

We ran a 2-point RANSAC algorithm between subsequent frames for outliers detection and rejection, using the relative orientation from the integrated gyroscope as rotation between the two frames.

An ended feature track is used as a measurement, only if:

- it spans at least three poses;
- its initial triangulation did not exhibit any degeneracy;
- its re-projection error is below a threshold.

Contrarily to our previous implementation [Caruso et al., 2017a], this threshold is dynamically set with a χ^2 threshold. As a result, the criterion becomes looser when the estimated uncertainty of relative poses increases.

In order to make the visual pipeline more robust to dark areas, we normalize each input image by its averaged intensity before corner detection and tracking. Some very dark images then become usable for corner detection despite a significant increase of the photometric noise which affects them. Noise amplification leads to spurious features track, but these are most often correctly rejected by our outlier rejection strategy. Overall, we found that normalization significantly improves the performance of pure MSCKF VINS algorithm on our dataset. Some raw/normalized images are presented in Figure 6.5.

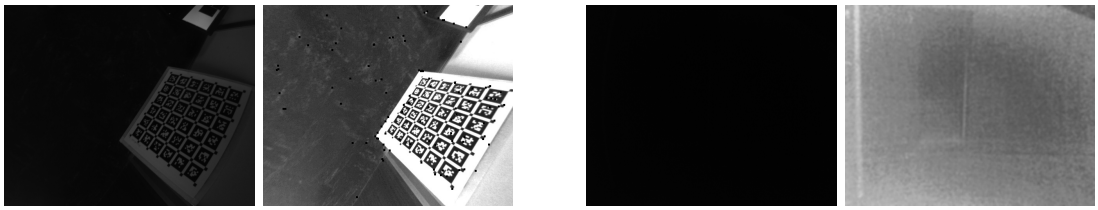


Figure 6.5: Image processing pipeline. (**Left**): Raw input images. (**Right**): after rectification, intensity normalization, and corner detection. On the second example, the normalization reveals a faint signal, but it is too noisy for corner detection to works.

Note that, in contrast with the tuning of the filter, the parameters of the vision pipeline (KLT window size, number of octaves in the pyramid of KLT, RANSAC threshold and minimum Harris score for corner detection) were chosen empirically.

6.5.4 Trajectory Evaluation

6.5.4.1 Dataset Presentation

We evaluate our algorithm on a dataset of five test trajectories. A pedestrian is carrying the system depicted in Figure 6.4 and walks through an industrial facility. Trajectories are specifically designed to be challenging for MI-DR and VINS: they are partly done outdoor, with very low magnetic field gradient, and they also contain portions visually non-informative made in the non-lit basement of the building. Detailed results on TRAJ2 and TRAJ5 are presented on Figures 6.6 , 6.7 and 6.8, the others being more briefly depicted in Figure 6.9. In all cases, a dedicated plot indicates with a color code the parts of the trajectory where the magnetic field gradient vanishes and parts of the trajectories where the mean intensity is low.

6.5.4.2 Overall Comparison

Three estimators derived from the presented filter have been compared. The MI-DR is the presented filter without any visual update. MSCKF is the presented filter without any magnetic update. MI-MSCKF is the proposed filter fusing both information. Moreover, we also compare our results with the state of the art VINS filter of [Paul et al., 2017].

Unfortunately, we do not have access to a ground truth trajectory for our datasets. We then consider three evaluation criteria:

- the estimated trajectories are superimposed to a georeferenced orthoimage in which one pixel represents precisely 0.5 m. We compute an alignment of the trajectory when the checkerboard detected for the first time in the sequence. This alignment results from setting manually the *position and heading* of the checkerboard frame relative to the coordinates system of the satellite image. Note that no manual scale alignment has been made; hence this visualization allows to evaluate roughly the correctness of the global scale of the estimate, for instance on Figure 6.6a;
- the z profile is globally known as the pedestrian walks along flat corridors—except when he takes stairs to change levels;

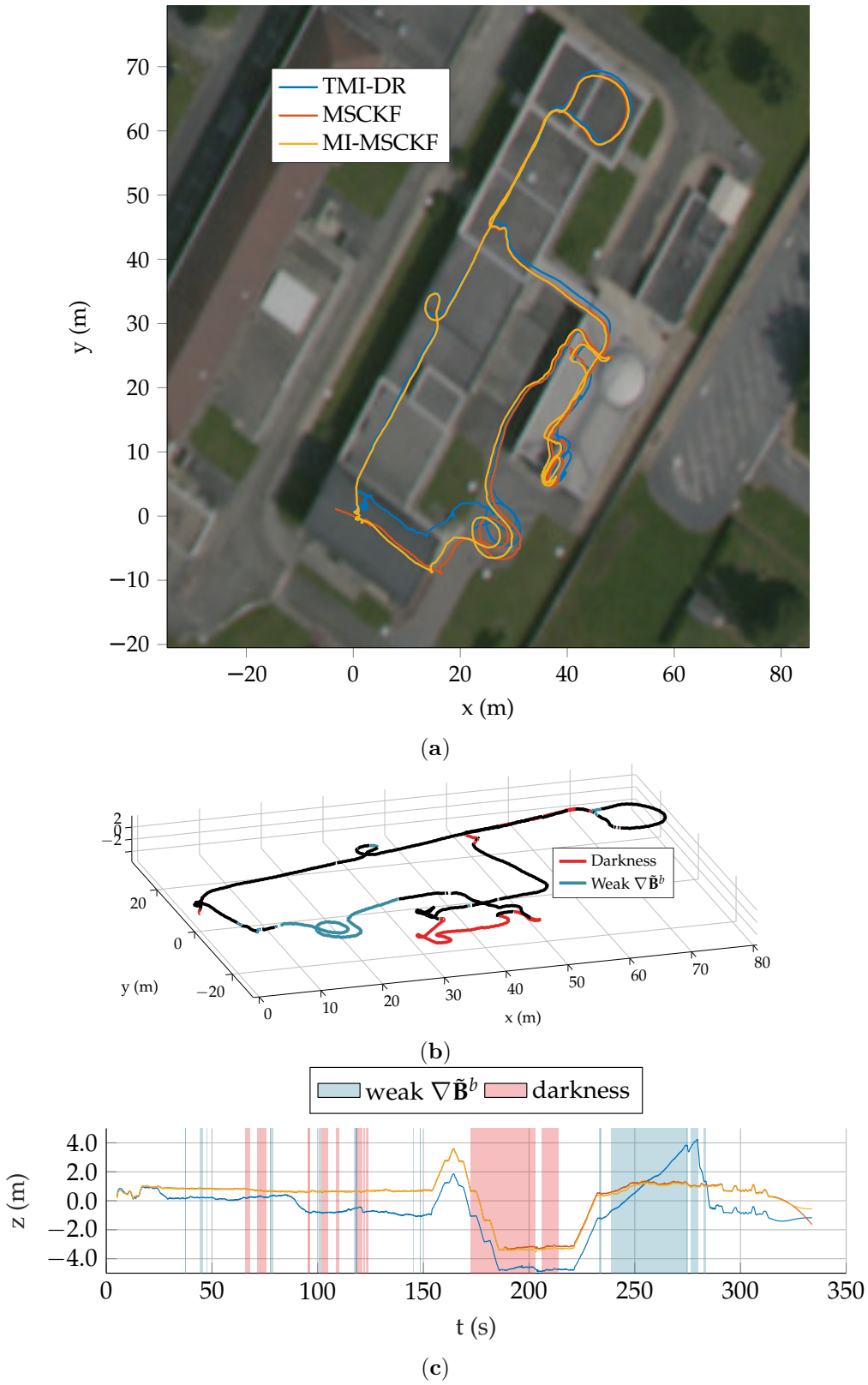


Figure 6.6: (**a**) Overview of trajectory TRAJ2 as reconstructed by the three filters. (**b**) Visualisation of dark areas and low-gradient areas over the entire trajectory surimposed on MI-MSCKF estimate. (**c**) Height profile of the three estimators on this trajectory.

- a translational error is computed each time the system comes back to its initial position, thanks to a static checkerboard placed at starting point. This criterion can be visualized in Figure 6.6c for Traj2 where it is clear the MI-DR estimate is less stable vertically over the entire trajectory.

The last criterion can be quantitatively evaluated: results are displayed in Table 6.1 with an error given in percentage of the trajectory length. Next sections emphasize some differences in the behaviors of the tested methods.

	Traj1	Traj2	Traj3	Traj4	Traj5
MI-DR	1.11	1.98	1.81	1.54	2.87
MSCKF (VINS)	0.33	0.63	0.59	1.05	0.21
MI-MSCKF	**0.20**	**0.31**	**0.49**	0.71	**0.15**
State of the art VINS [Paul et al., 2017]	0.26	0.52	0.79	**0.62**	0.20

Table 6.1: Summary of final drift error on full dataset (% of trajectory length).

6.5.4.3 The Fused Estimate Improves MI-DR in Outdoor Trajectories

The Figure 6.6 shows the three versions of our filter on Traj2. The trajectory estimated by MI-DR is very close to the others until some point in the outdoor part—note that the outdoor part corresponds to the weak gradient part of the trajectory as depicted on Figure 6.6b. During this outdoor part, as expected, the MI-DR drifts away compared to the two vision-based estimates which directly leads to a higher final translation error. The same effect is also clearly visible on Traj3, see Figure 6.9b.

6.5.4.4 Data Fusion Improves Local Consistency

By reducing the drift in dark areas or low gradient areas, the fused estimates improve the local position estimate consistency, an effect which is not always visible on the metrics of Table 6.1.

The benefit of magnetometry information in this sense is demonstrated in the details of results on the Traj2 displayed in Figure 6.7. The two left plots show a similar situation: the VINS estimate (red) is for a few seconds strongly corrected by the filter, leading to non-continuous estimates (see green circles). The MI-MSCKF filter stays smoother during the entire trajectories, thanks to the speed observability provided by Equation (6.7). Interestingly, the pure VINS estimate joins the MI-MSCKF estimate later in the sequence, which makes the temporary drift mostly invisible in the final loop error metric of Table 6.1. This correction happens when visual information becomes available again. It means that the VINS filter is still able to correct itself after reasonable drift through the information stored in the prior.

The same effect occurs on all trajectories. Consider for instance the trajectory Traj5 depicted on Figure 6.8, where the lowest part also goes through the dark basement of the building. The MSCKF drifts vertically before being corrected when the pedestrian takes the stair up again. This effect is depicted on details in Figure 6.8c, which clarifies the evolution of the estimated height with respect to time.

In turn, visual information also helps trajectory consistency. It is clearly visible on the strong vertical drift of MI-DR displayed on Figure 6.6c around 250 s. Again, this drift is corrected as soon as magnetic gradients become sufficiently high to make speed observable again. Note the horizontal drift was never corrected though, leading to the larger final drift shown in Table 6.1.

The reader may have noticed at that point that Figure 6.8c shows that MI-DR fails badly on Traj5. Yet, we would like to stress that the fully fused estimate is still able to outperform the VINS estimate, leveraging magnetic information correctly beyond the breaking point of MI-DR. The reason for the failure of MI-DR is a nonstationary local magnetic field in the first few seconds of Traj5. It perturbs the MI-DR initialization, which has dramatic consequences on all the rest of the trajectory.

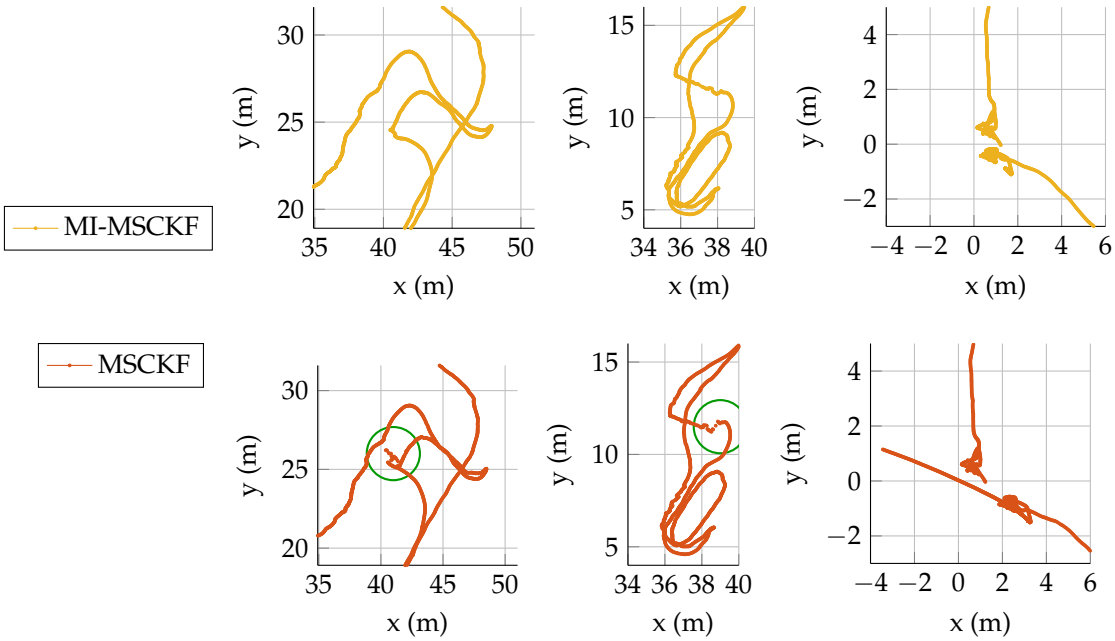


Figure 6.7: Details of estimation results on TRAJ2 showing a different behavior between the MSCKF and MI-MSCKF filter. **Left and middle plots**: while transitioning from dark area to lit environment some strong filter correction happen for the MSCKF and lead to discontinuities of the position estimate. In the same areas, MI-MSCKF stays smoother. **Right plot**: here the device is laid on the ground at the end of the trajectory. A large drift of MSCKF occurs, as visual information does not provide any feedback on position. Here again, the MI-MSCKF appears more stable.

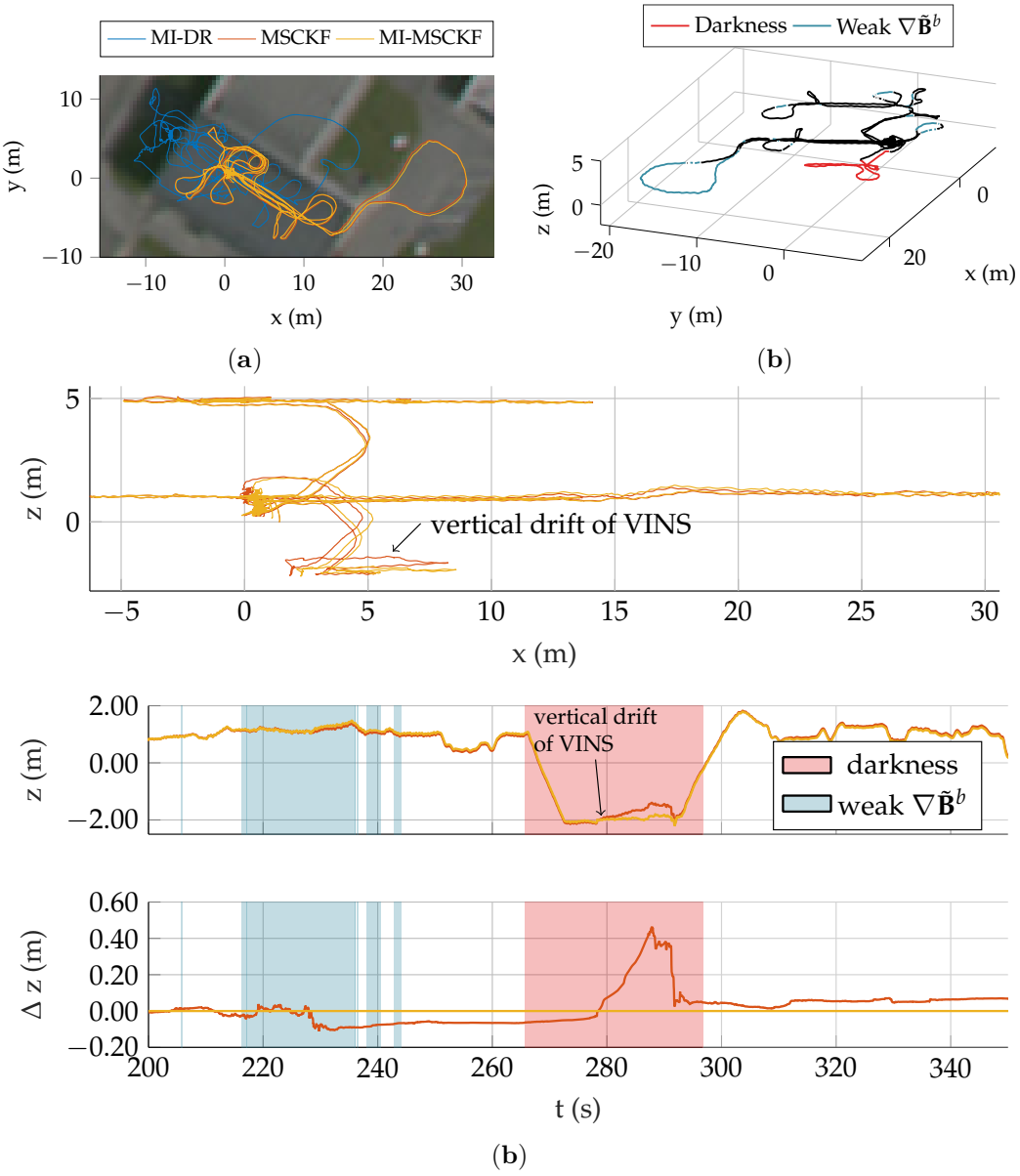


Figure 6.8: (**a**) Overview of trajectory Traj2 as reconstructed by the three filters. (**b**) Visualization of dark areas and low-gradient areas over the entire trajectory surimposed on MI-MSCKF estimate. (**c**) Height profile of the three estimators on this trajectory.

The fact that the fused estimate prevents local drift could be, in our opinion, highly beneficial to the long-term performance of an Extended Kalman Filter strategy. Indeed, it could reduce overall linearization errors and the maximum magnitude of corrections, which are recognized, in VINS community, as an important drawback of filtering approaches compared to bundle adjustment or optimization-based methods.

6.5.4.5 Comparison with a State of the Art Filter

We also ran the released binary version [2] of [Paul et al., 2017] on our dataset. Indeed, we think it is the state-of-the-art in VINS filters for pedestrian navigation. As [Paul et al., 2017] takes only stereo images (even if their method works well with monocular setup also) we had to trick slightly their software to use a monocular input: we generate an virtual black image for the second input image of their software. Note that, to be as fair as possible, we have entered in their code the same normalized monocular images we use. In doing so, we observed that normalization has also drastically improved the performance of their filter on our data.

An in-depth and comprehensive comparison between the two implementations is difficult as the code of [Paul et al., 2017] is not open. If inertial handling in both implementations should be close, the visual pipeline is very sensitive to parameters value and implementation details that are not known by us. Nevertheless, Table 6.1 demonstrates that both filters compare reasonably and are clearly below 1% of trajectory length error.

We also observed in [Paul et al., 2017] implementation the presence of strong filters correction after dark areas, as in described in Section 6.5.4.4. It indicates that this local consistency problem, which is essentially solved by the proposed fused estimate, is indeed a general issue of all VINS filters.

6.6 Conclusions

This work presented a filter to fuse information from a magnetometer array with other sensors traditionally used in VINS. We described in detail, the method we used and discussed its results on real datasets. Comparing the results of three estimators—one using only magnetic-inertial information, one using only visual-inertial information and one fusing both pieces of information —we showed that the fused estimate leads to a more robust trajectory estimate. First, our fused estimate is able to reconstruct the trajectory outdoor where MI-DR techniques break because of the lack of gradient; secondly, our system avoids the unrealistic trajectory correction of VINS after significant duration without proper illumination. One trajectory also highlighted the need for either robust estimation technique or outlier rejection scheme for magnetic information. This should deserve more work in the future.

We think that the proposed approach could help to improve localization systems for AR that are currently using VINS with consumer grade cameras and IMUs. Not only this combination extends the applicability domain of traditional VINS to degraded environments, but we also foresee opportunities to reduce power consumption. Indeed, taking advantages of the good trajectories given by the MI-DR in various conditions, it might be possible to reduce the computational load of the visual pipeline, which could be of significant interest in practical applications.

[2] available on https://github.com/MARSLab-UMN/MARS-VINS, we used the commit 8531daf.

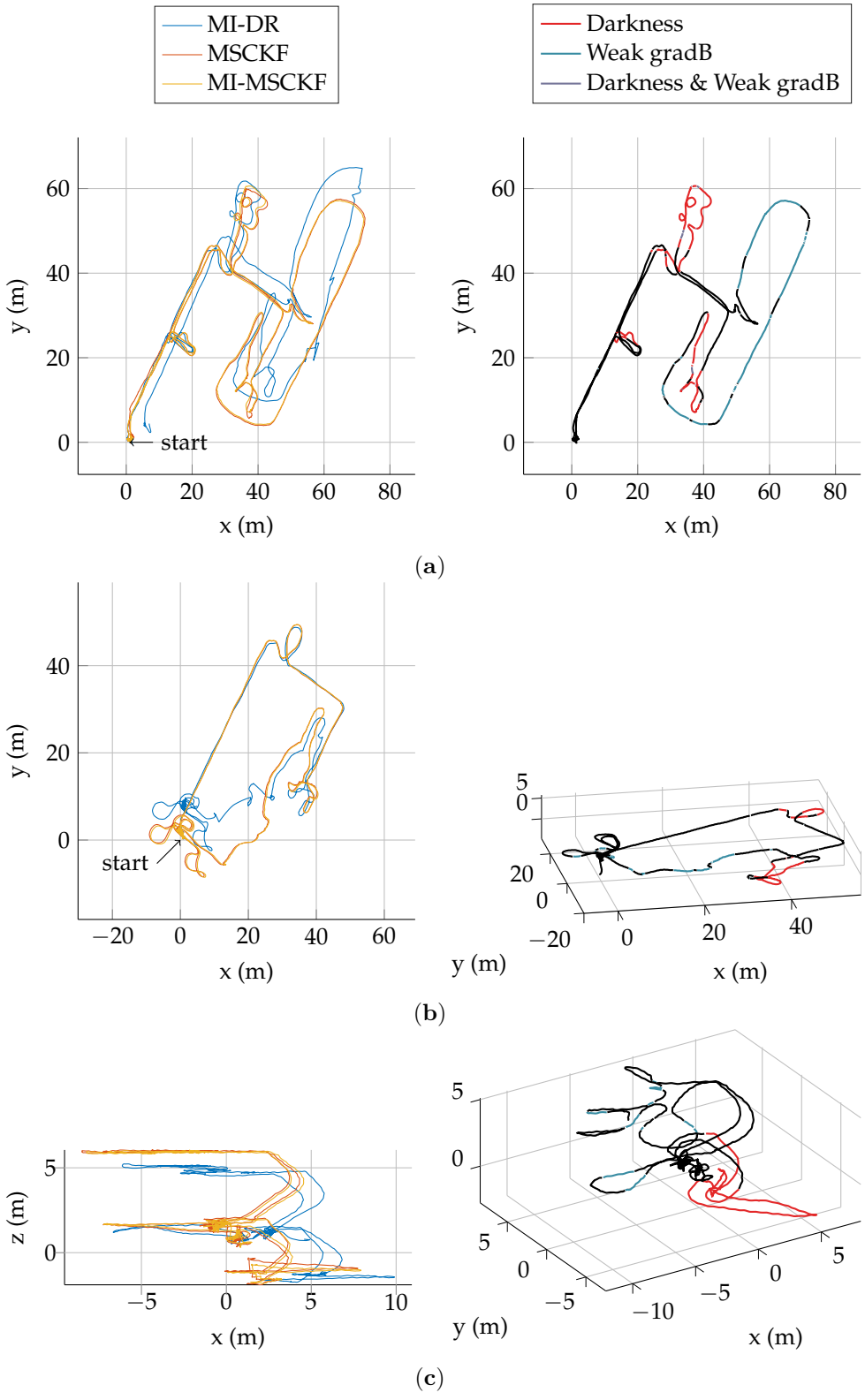


Figure 6.9: Summary of trajectories on the remaining sequences of the dataset. (**Left**): Estimate of the three configuration of our filter. (**Right**) Color coded MI-MSCKF trajectory showing areas of weak gradient and weak illumination. (**a**) TRAJ1 Length: ∼530 m; (**b**) TRAJ3 Length: ∼368 m; (**c**) TRAJ4 Length: ∼180 m.

6.7 Conclusion of the Chapter

Note This section is an addition to the journal paper and presents new results.

6.7.1 Difference with the Sliding Window Smoother of Chapter 5

This chapter presented a MVINS method relying on an EKF. This filter exploits the magnetic and visual information somewhat similarly to the optimization method described in Chapter 5, but with some differences that we underline hereafter:

- *High-frequency magnetometer measurement.* The presented filter processes magnetic field value measurement at the MIMU sample rate. This is in contrast to the sliding window smoother where magnetic field *value* measurement was used at the *image* sample rate only (even if preintegrated magneto-inertial measurements were computed using gradient information at 325Hz).
- *Delayed landmark measurement.* Landmark are marginalized as soon as their corresponding feature track ends, and do not influence position estimate until then. This is in contrast with the sliding window smoother that integrates landmarks and reprojection residual in its cost function as soon as they can be triangulated.
- *Earlier marginalization of* MIMU *states.* In contrast with the sliding windows smoother of Chapter 5, MIMU variables (magnetic field, speed, and biases) are marginalized earlier: there is no short-term window as in the previous chapter.
- *Non-robust cost* The MSCKF cannot be robustified with a robust loss. A χ^2 gating test is used instead before applying the update step. This test gives a binary answer: either the landmark is used, either it is not. This is in contrast with the robust loss used in previous chapters that blurs the frontier between outliers and inliers in optimization.

The implementation of the filter whose results are depicted in this chapter shares with the optimization the image processing and features tracking part while the inference process has been reimplemented from scratch. Note that it was not a necessity; indeed, one could see the MSCKF filter as a different marginalization strategy for the cost function (5.19), Page 78. In results, MI-MSCKF could be implemented using factor-graph machinery that GTSAM library internally uses. We actually started with this approach, yielding the inverse square root filter presented in our paper [Caruso et al., 2017a].

In contrast, the results presented in this chapter were obtained with an entirely different implementation of the filter leveraging solely raw c++ and Eigen[3] linear algebra library and discarding GTSAM library totally (it is also the case for the following chapter results). The aim of this reimplementation was to stop relying on the inference process of GTSAM, taken as a black box in [Caruso et al., 2017a] and [Caruso et al., 2017b]. Also, this permits investigating fine properties of the filter that are presented in the next chapter: the experiments presented in Chapter 7 required more control on the parametrization of the state that what was provided by the GTSAM library API.

6.7.2 How does optimization based and filtering based estimators compare?

As can be seen in Table 6.2, the order of magnitude of the drift of both methods are similar, with a slight improvement coming from optimization method on TRAJ3, TRAJ4, TRAJ5. However, the difference in percentage of trajectory length stays small. Detailed comparison are provided on Figure 6.11.

This would advocate in our opinion in favor of filtering approach for pure dead-reckoning applications: they seem to attain similar accuracy while being lighter and easier to implement.

[3] http://eigen.tuxfamily.org

	Traj1	Traj2	Traj3	Traj4	Traj5
MI-DR	1.11	1.98	1.81	1.54	2.87
MSCKF (VINS)	**0.13**	0.40	0.36	0.38	0.34
MI-MSCKF	0.34	**0.33**	0.55	0.32	0.23
VINS optim	0.35	0.54	0.32	0.29	0.25
MVINS optim	0.53	0.43	**0.31**	**0.25**	**0.18**

Table 6.2: Summary of final drift error on full dataset (% of trajectory length). Note: these number can no be compared directly to the number in MDPI paper, because the way outlier are handled is slightly different, they can however be compared to number of chapter 7.

More importantly, we observe the same discontinuities of trajectories at the end of long run of the estimator (see Figure 6.10 and Figures 6.11a and 6.11b); a phenomenon already observed for the sliding windows smoother and noted in Section 5.7.

This fact is intriguing, as the implementation of [Paul et al., 2017] does not exhibit such behaviors on our data, despite showing a slightly higher drift in general (see for instance how the yellow curve on Figure 6.11a behaves differently from the blue and red ones).

If a workaround was used to mask this effect in Section 5.7, it was not entirely satisfying, both from a theoretical point of view and from a drift point of view. This phenomenon motivated us to study more theoretical property that might explain it in a filtering context; one candidate was the known lack of consistency often remarked in filter-based SLAM systems and for which several solutions exist in the literature. This was the original motivation of the work presented in next chapter.

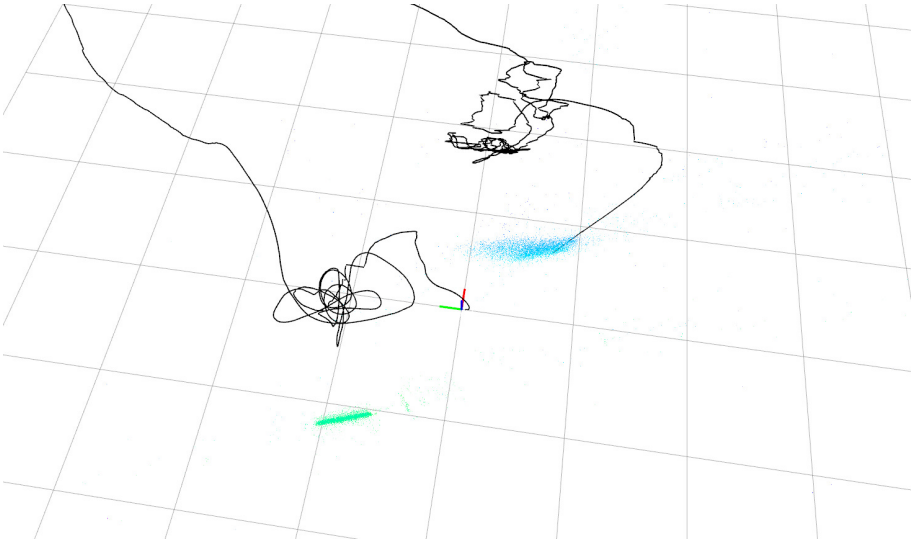
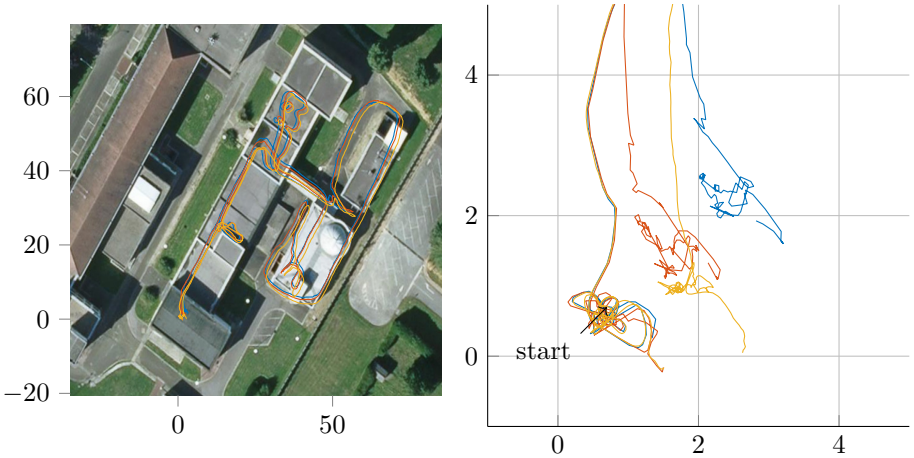
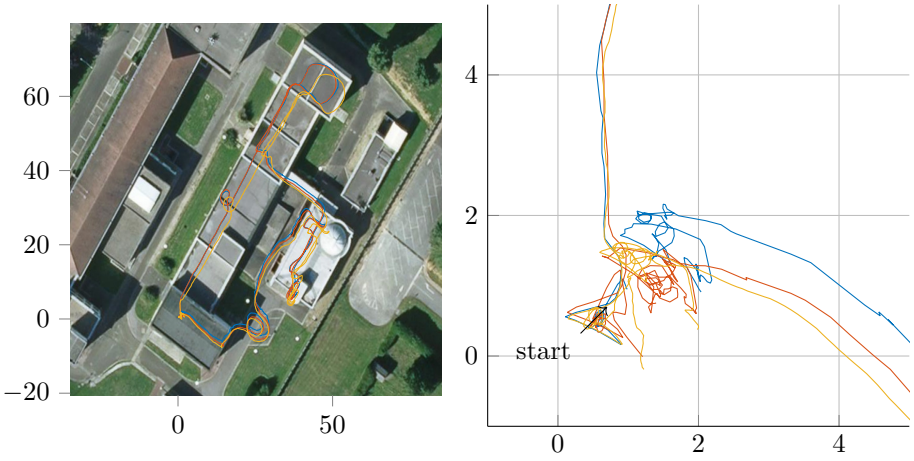


Figure 6.10: Instability and trajectory discontinuity of the MI-MSCKF. Similarly to the sliding window smoother of Chapter 5, after a long run of the filter, the trajectory becomes unstable. View on Traj1: the left trajectory is the beginning of the trajectory, the right (discontinuous) part is the end. The green point cloud is the reprojection of point belonging to a fixed targetboard at the beginning of the trajectory while the blue one is the same at the end of the trajectory. The blue one is distorted, proving the drift of the filter simultaneously to its instability

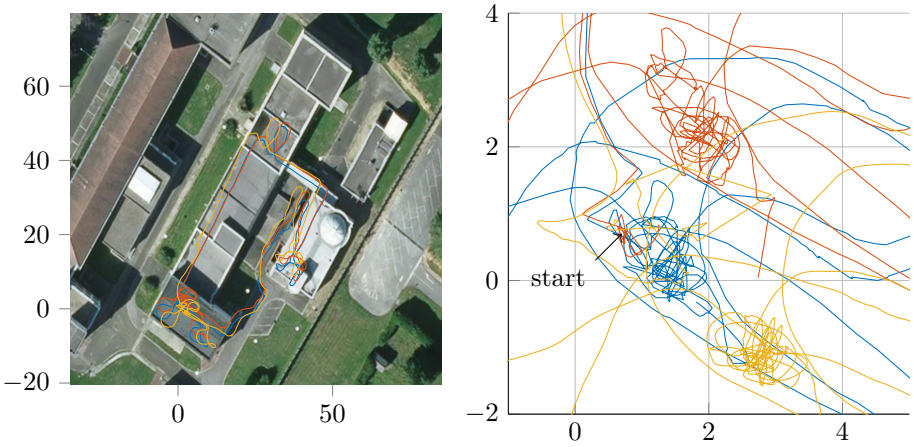
(a)
TRAJ1
Length:
∼ 530 m



(b)
TRAJ2
Length:
∼ 380 m



(c)
TRAJ3
Length:
∼ 368 m



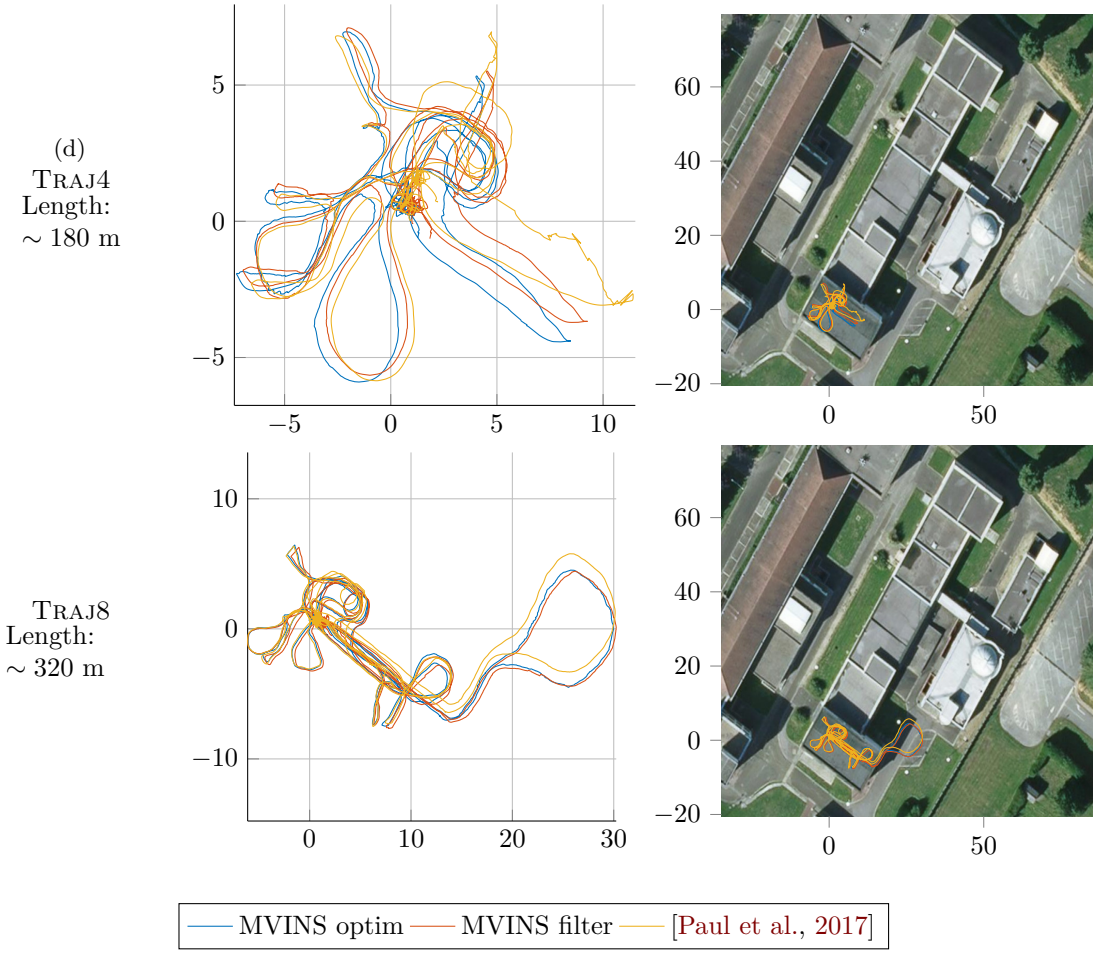


Figure 6.11: Comparison of MVINS filter, MVINS optim and state-of-the-art VINS algorithm on our dataset. For TRAJ1 to TRAJ3, we focus on the beginning of the trajectory in order to visualize drift and local behavior at the end of the trajectory. Our filter and optimization methods behave quite similarly.

Chapter 7

Invariance and consistency properties of MVINS filters

This chapter deals with fine properties of the filtering approach presented in Chapter 6. After exposing our motivations and some notations specific to this chapter in Section 7.1, we start in Section 7.2 by observing, on real data, that the filter of the previous chapter is not consistent in general. These inconsistencies arise from both linearization errors and the way feature-tracks are handled in the visual update step. We then show and demonstrate how the behavior of the filter can be altered by a change of parametrization in Section 7.3. We analyze the result of a new parametrization built around symmetries of the filter model and show – on real data – that the resulting filter exhibits better consistency and invariance properties. This work has not been published in peer-reviewed conference or journal at the time of writing.

7.1 Introduction

7.1.1 Motivation

It is well-known that the EKF-SLAM and the MSCKF are not consistent. By "not consistent", we mean here that the error and its propagation are not well represented by the estimated covariance matrix of the filter (or its inverse in information filter). Several old and recent works in SLAM and visual odometry community have shown that some parts of this inconsistency arise from phenomena involving the estimator chosen solely; i.e., they also appear in the case of data simulated with true gaussian noises, as was demonstrated for instance in the simulation result of [Hesch et al., 2014]. Recently some works linked the error definition used to build the EKF with its consistency properties. In [Barrau and Bonnabel, 2015], the authors introduce a beneficial error reparametrization for SLAM problem, that had been extended to 3D-VINS model in [Zhang et al., 2017a].

The work presented in this chapter was initially an attempt to solve the discontinuous trajectory we observed after a long run of our filter, and also to determine if these inconsistencies were a significant issue on our data, and finally, to extend modelization work from other parts of the VINS community to our MVINS framework.[1]

7.1.2 Kalman Filtering with Non-linear Error

In this chapter, we study the filtering approach with a very general formulation of the Kalman filter. We assume that uncertainty of the filtering process is represented through a covariance matrix $\mathbf{\Sigma_e}$

[1] No need to maintain uselessly any suspense here; the methods presented, even if they have interesting properties that we demonstrate, did not solve our practical problem of trajectory discontinuity presented in Section 5.7.

of a non-linear error vector $\mathbf{e}$ defined through an abstract $\boxplus$ operator, such that:

$$\underbrace{\mathbf{X}}_{\text{true}} = \underbrace{\hat{\mathbf{X}}}_{\text{estimated}} \boxplus \underbrace{\mathbf{e}}_{\text{error}} \tag{7.1}$$

$$\mathbf{e} \propto \mathcal{N}(0, \mathbf{\Sigma}_{\mathbf{e}}) \tag{7.2}$$

The binary operator $\boxplus$ is sometimes called the *retraction operator*. We already employed this notation in previous chapters; it was required mainly to seamlessly use gradient-based algorithms or Kalman update equation with variables belonging to the SO(3) Lie group. However, we emphasize that the choice of the error – and simultaneously of retraction operator – is merely a design choice. For instance, if we were to filter on position $\mathbf{p}$ and orientation $\mathtt{R}$ of a device, we could see both variables together as an element of SE(3) group and thus filter on an error in SE(3) Lie algebra such as,

$$(\boxplus \text{ definition }) \quad \boldsymbol{\xi} = \mathrm{Exp}_{\mathrm{SE}(3)}(\mathbf{e})\,\hat{\boldsymbol{\xi}} \tag{7.3}$$

$$\text{with} \quad \mathbf{e} \in \mathbb{R}^6, \quad \boldsymbol{\xi}, \hat{\boldsymbol{\xi}} \in \mathrm{SE}(3), \tag{7.4}$$

or we could alternatively choose to filter on an error built on the space $\mathrm{SO}(3) \times \mathbb{R}^3$, for instance, defined as

$$\boldsymbol{\xi} = (\mathtt{R}, \mathbf{p}) \quad , \hat{\boldsymbol{\xi}} = (\hat{\mathtt{R}}, \hat{\mathbf{p}}) \tag{7.5}$$

$$(\boxplus \text{ definition }) \quad \begin{cases} \mathtt{R} & = \mathrm{Exp}_{\mathrm{SO}(3)}(\mathbf{e}_{\mathtt{R}})\,\hat{\mathtt{R}} \\ \mathbf{p} & = \mathbf{e}_{\mathbf{p}} + \hat{\mathbf{p}} \end{cases} \tag{7.6}$$

$$\text{with } \hat{\mathtt{R}}, \mathtt{R} \in \mathrm{SO}(3), \quad \hat{\mathbf{p}}, \mathbf{p} \in \mathbb{R}^3, \quad \mathbf{e} = [\mathbf{e}_{\mathtt{R}}, \mathbf{e}_{\mathbf{p}}] \in \mathbb{R}^6. \tag{7.7}$$

This choice changes the space in which the uncertainty of the filter is approximated as Gaussian and governs the linearization process of the filter. For these reasons, it has quite naturally some impact on the filter performance.

In this chapter, we will introduce an alternative choice with respect to the one adopted in the previous chapter for the $\boxplus$ operator for MI-MSCKF filter, and we will study the effect of such a change on the filtering process.

This error operator must verify expected properties for an error such as $\mathbf{X}_0 \boxplus 0 = \mathbf{X}_0$ and have a reciprocal operator $\boxminus$ so that $\mathbf{e} = \mathbf{X} \boxminus \hat{\mathbf{X}}$ around zero. To use it in a EKF, we also need it to be continuous and differentiable at least in the vicinity of zero.

Within this framework, general continuous and discrete EKF algorithms can be written. Let us consider a generic discrete model:

$$\begin{aligned} &\text{(propagation)} && \mathbf{X}_{k+1} = f(\mathbf{X}_k, \tilde{\mathbf{u}}_k, \eta_k), \\ &\text{(measurement)} && y = h(\mathbf{X}_k, \nu_k) \end{aligned} \tag{7.8}$$

with:

$$\tilde{\mathbf{u}}_k \propto \mathcal{N}(\mathbf{u}_k, \mathbf{\Sigma}_{\mathbf{u}_k}), \quad \eta_k \propto \mathcal{N}(0, \mathbf{\Sigma}_\eta), \quad \nu_k \propto \mathcal{N}(0, \mathbf{\Sigma}_c)$$

In this case, the Kalman equations write:

- Propagation:

$$\hat{\mathbf{X}}_{k+1|k} = f(\hat{\mathbf{X}}_k, \tilde{\mathbf{u}}_k, 0) \tag{7.9}$$

$$\mathbf{\Sigma}_{\mathbf{e}k+1|k} = \mathbf{\Phi}_{k+1}\mathbf{\Sigma}_{\mathbf{e}k}{\mathbf{\Phi}_{k+1}}^\mathsf{T} + \mathbf{G}_{k+1}\mathbf{\Sigma}_{\mathbf{u}}{\mathbf{G}_{k+1}}^\mathsf{T} + \mathbf{C}_{k+1}\mathbf{\Sigma}_\eta{\mathbf{C}_{k+1}}^\mathsf{T} \tag{7.10}$$

  Where $\tilde{\mathbf{u}}_k$ is a corrupted measurement of the input $\mathbf{u}_k$ and matrices $\mathbf{\Phi}_{k+1}$, $\mathbf{G}_{k+1}$, $\mathbf{C}_{k+1}$ are respectively the Jacobian matrices of the process function f with respect to the state, the

input, and the stochastic input of the model:

$$\mathbf{\Phi}_{k+1} = \frac{\partial}{\partial \mathbf{e}} \Big(f(\hat{\mathbf{X}}_k \boxplus \mathbf{e}, \mathbf{u}_k, \eta) \boxminus f(\hat{\mathbf{X}}_k, \tilde{\mathbf{u}}_k, 0) \Big) \Big|_{\mathbf{e}=0, \mathbf{u}_k=\tilde{\mathbf{u}}_k, \eta=0} \tag{7.11}$$

$$\mathbf{G}_{k+1} = \frac{\partial}{\partial \mathbf{u}_k} \Big(f(\hat{\mathbf{X}}_k \boxplus \mathbf{e}, \mathbf{u}_k, \eta) \boxminus f(\hat{\mathbf{X}}_k, \tilde{\mathbf{u}}_k, 0) \Big) \Big|_{\mathbf{e}=0, \mathbf{u}_k=\tilde{\mathbf{u}}_k, \eta=0} \tag{7.12}$$

$$\mathbf{C}_{k+1} = \frac{\partial}{\partial \eta} \Big(f(\hat{\mathbf{X}}_k \boxplus \mathbf{e}, \mathbf{u}_k, \eta) \boxminus f(\hat{\mathbf{X}}_k, \tilde{\mathbf{u}}_k, 0) \Big) \Big|_{\mathbf{e}=0, \mathbf{u}_k=\tilde{\mathbf{u}}_k, \eta=0} \tag{7.13}$$

- Update with measurement $\tilde{y}_{k+1}$:

$$\hat{\mathbf{X}}_{k+1} = \hat{\mathbf{X}}_k \boxplus \left(\underbrace{\mathbf{K}_{k+1}}_{\text{Gain}} \underbrace{\Big(\tilde{y}_{k+1} - h(\hat{\mathbf{X}}_{k+1|k}, 0) \Big)}_{\text{innovation}} \right) \tag{7.14}$$

$$\mathbf{\Sigma}_{\mathbf{e}k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{\Sigma}_{\mathbf{e}k+1|k} \tag{7.15}$$

Where the linearized measurement matrix $\mathbf{H}_{k+1}$ is defined as:

$$\mathbf{H}_{k+1} = \frac{\partial}{\partial \mathbf{e}} \Big(h(\hat{\mathbf{X}}_{k+1|k} \boxplus \mathbf{e}) \Big) \Big|_{\mathbf{e}=0} \tag{7.16}$$

and $\mathbf{K}_{k+1}$ is the *Kalman gain* defined here as:

$$\mathbf{K}_{k+1} = \mathbf{\Sigma}_{ek+1|k}\mathbf{H}_{k+1}^{\mathsf{T}} \underbrace{\big(\mathbf{H}_{k+1}\mathbf{\Sigma}_{ek+1|k}\mathbf{H}_{k+1} + \mathbf{\Sigma}_c\big)^{-1}}_{\mathbf{S}_{k+1}^{-1} \text{ : inverse covariance of innovation}} \tag{7.17}$$

Note that the choice of the error is in theory entirely decoupled from how the mean estimates are kept in memory. For instance, in the MI-DR filter, we could store the speed and magnetic field estimates in *world* frame while using an error for magnetic and speed in *body* frame. In the previous chapter we were actually storing the speed and magnetic in *body* frame; if we were to do the opposite and to keep the error in *world* frame, it would correspond to choosing the following retraction operator:

$$\hat{\mathbf{X}}_k \boxplus \mathbf{e} = \begin{pmatrix} e^{\mathbf{e}_\mathbb{R}} \hat{\mathbb{R}}^w, \\ e^{\mathbf{e}_\mathbb{R}} (\hat{\mathbf{v}}^w + \hat{\mathbb{R}}^w \mathbf{e}_\mathbf{v}), \\ \hat{\mathbf{p}}^w + \mathbf{e}_\mathbf{p}, \\ e^{\mathbf{e}_\mathbb{R}} (\hat{\mathbf{B}}^w + \hat{\mathbb{R}}^w \mathbf{e}_\mathbf{B}), \\ \hat{\mathbf{b}_g} + \mathbf{e}_{\mathbf{b}_g}, \\ \hat{\mathbf{b}_a} + \mathbf{e}_{\mathbf{b}_a}, \end{pmatrix} \tag{7.18}$$

Of course, to obtain a valid filter, we need to use the corresponding transition matrix and measurement matrix.

In this chapter, we will choose to write the magnetic field $\mathbf{B}$ and the velocity $\mathbf{v}$ in *world* frame instead of *body* frame.

Finally, we want to underline that, as done in this introduction, it was preferred in this chapter the traditional formulation of the Kalman filter for demonstrations, i.e. with uncertainty kept as a covariance matrix instead of square root information matrix and with a Kalman gain feedback instead of the least squares update used in Chapter 6. However, the experimental results presented hereafter were still obtained from an inverse square root filter as presented in Chapter 6, albeit modified according to the change of parametrization. Note that the inverse square root formulation does **not** change the conclusions as both formulations are equivalent except for numerical properties.

7.2 Consistency Problem of the Filtering Approach

7.2.1 Unobservabilities in the MVINS Model

We recall here the MVINS model we used in previous chapter. The deterministic part of the MVINS model, as used in a MSCKF framework, writes:

State :

$$\mathbf{X}_k = (\underbrace{\mathtt{R}^w_{k-nc}, \mathbf{p}^w_{k-nc}, \dots, \mathtt{R}^w_{k-1}, \mathbf{p}^w_{k-1}}_{\text{stochastic clones}} \mid \underbrace{\mathtt{R}^w_k, \mathbf{p}^w_k, \mathbf{v}^w_k, \mathbf{B}^w_k, \mathbf{b}_{g_k}, \mathbf{b}_{a_k}}_{\text{current state}}) \tag{7.19}$$

We recall we name here "stochastic clones" the poses of past keyframes still in the sliding window, as originally done in the MSCKF seminal paper [Mourikis and Roumeliotis, 2007].

MIMU state dynamic: The function f is defined by the following discrete model. (see Section 6.3.3 for symbol definition.)

$$\text{(inertial prediction)} \quad \begin{cases} \mathtt{R}^w_{k+1} = \mathtt{R}^w_k \widetilde{\Delta \mathtt{R}}_{kk+1}, \\ \mathbf{v}^w_{k+1} = \mathbf{v}^w_k + \mathbf{g}^w \Delta t_{k;k+1} + \mathtt{R}^w_k \widetilde{\Delta \mathbf{v}}_{kk+1}, \\ \mathbf{p}^w_{k+1} = \mathbf{p}^w_k + \mathbf{v}^w_k \Delta t_{k;k+1} + \frac{1}{2}\mathbf{g}^w \Delta t^2_{k;k+1} + \mathtt{R}^w_k \widetilde{\Delta \mathbf{p}}_{kk+1}, \end{cases} \tag{7.20}$$

$$\begin{aligned} \text{(magnetic prediction)} \quad \mathbf{B}^w_{k+1} &= \mathbf{B}^w_k + \mathtt{R}_k \widetilde{\mathbf{\Delta B}}_{\mathbf{v};kk+1} \mathtt{R}^\mathsf{T}_k \mathbf{v}^w_k \\ &\quad + \mathtt{R}_k \widetilde{\mathbf{\Delta B}}_{\mathbf{g};kk+1} \mathtt{R}^{w\mathsf{T}}_k \mathbf{g}^w + \mathtt{R}_k \widetilde{\mathbf{\Delta B}}_{\mathbf{a};kk+1}. \end{aligned} \tag{7.21}$$

$$\text{(bias model)} \quad \begin{cases} \mathbf{b}_{g\,k+1} = e^{\frac{\Delta t_{k;k+1}}{\tau_{bg}}} \mathbf{b}_{g\,k} + \boldsymbol{\eta}_{bg} \\ \mathbf{b}_{a\,k+1} = e^{\frac{\Delta t_{k;k+1}}{\tau_{ba}}} \mathbf{b}_{a\,k} + \boldsymbol{\eta}_{ba} \end{cases} \tag{7.22}$$

Magnetic measurement: This is still a direct measurement of magnetic field state, in *body* frame.

$$h_{\text{magn}}(\mathbf{X}_k) = \mathtt{R}^{w\mathsf{T}} \mathbf{B}^w_k \tag{7.23}$$

Visual measurement: As in previous chapter, Section 6.4.3.2 Page 122, the $\mathbf{H}_{\text{feat}}$ matrix is based on the following linearization:

$$h_{\text{feat}}(\hat{\mathbf{X}} \boxplus \mathbf{e}, \mathbf{l}^* + \delta\mathbf{l}) = h_{\text{feat}}(\hat{\mathbf{X}}, \mathbf{l}^*) + \mathbf{F}\mathbf{e} + \mathbf{E}\delta\mathbf{l} + o(\mathbf{e}, \delta\mathbf{l}) \tag{7.24}$$

where $\mathbf{F}$ are the Jacobian of the measurement function with respect to the state (in fact solely part associated to stochastic clones) and $\mathbf{E}$ the Jacobian to the landmark position parameters $\mathbf{l}$.[2] In this expression, the linearization point $\mathbf{l}^*$ is computed by a triangulation algorithm, with poses of the current state assumed known. The measurement matrix $\mathbf{H}_{feat}$ is computed by projecting this equation over the nullspace of $\mathbf{E}$ with (by QR decomposition):

$$\mathbf{E} = [\mathbf{O}_1, \mathbf{O}_0] \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix}, \quad [\mathbf{O}_1, \mathbf{O}_0] \in \boldsymbol{\mathcal{O}}, \text{ and } \mathbf{R} \text{ an upper triangular matrix} \tag{7.25}$$

$$\mathbf{H}_{feat} = \mathbf{O}_0^\mathsf{T} \mathbf{F} \tag{7.26}$$

It is well-known that the VINS part of the model has four degrees of freedom that are not observable: one degree of rotation around the gravity vector, and three degrees of translation.

[2] The landmark parametrization can be chosen indifferently: e.g. 3D position of landmark in world frame, 3D position in the coordinate frame of the first camera which has seen it, inverse depth parametrization on first ray, etc.

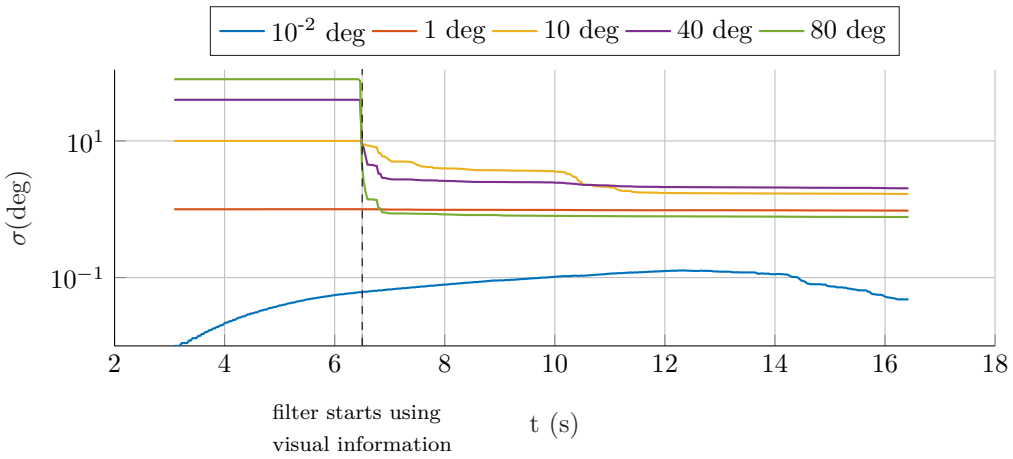


Figure 7.1: Covariance of heading on a real dataset with various initial covariance for filter of Chapter 6 (log scale). The first 6.5 seconds are filter initialization, where only magnetic update is employed. Albeit being non observable in the MVINS model, the covariance of heading decreases as soon as visual measurement are used. This is striking if this covariance was initialized with very high values. This phenomenon is inconsistent with an observability analysis. The blue curves has slightly different shape for scale reason (see text)

[Jones and Soatto, 2011; Martinelli, 2013b]. These degrees of freedom correspond to the invariance of the entire model to the change of the *world* reference frame anchor point and heading. The additional magnetic information of the MVINS model does not bring any absolute heading or position information neither, so it stays invariant to such a change of reference frame.[3] Consequently, the rotation around gravity and absolute translation are also unobservable MVINS.

7.2.2 Observation on Real Data

7.2.2.1 Spurious gain of information on unobservable angle yaw

The Figure 7.1 depicts the uncertainty of the filter on the heading degree of freedom.[4] Five instances of the filter of Chapter 6 are run with different initialization mean and covariance value and the same input data. The initialization values only differ from the initial uncertainty of the heading angle. We use for these experiments real data: the first few seconds of dataset TRAJ5 of our Indoor/Outdoor dataset.

We observe that for high initial uncertainty, the heading uncertainty estimated by the filter ends up lower than the initial uncertainty. This is a sign of inconsistency of the estimator, as the model does not provide information on the absolute heading. Interestingly, this drastic decrease of uncertainty co-occurs with the first use of visual measurements, proving these are the primary source of this inconsistency.

Note that as we use a logarithmic scale on this plot, we can see a small increase of uncertainty due to gyroscope noise at the smallest value of the initial uncertainty (see blue curve corresponding to an initial uncertainty of 1.10^{-2} rad), while they are hidden at the higher uncertainty levels.

[3] We recall that we do not use the magnetic field as a north direction measure, as assuming the perceived magnetic field is equal to the earth magnetic field is a dangerous assumption in indoor environments.
[4] More precisely the corresponding diagonal value of the covariance.

7.2.2.2 A definition of filter invariance

We will work in this chapter with a definition of the invariance of an EKF to *unobservable stochastic transform* which was proposed in [Zhang et al., 2017a] (Definition 3). This definition has the advantage to be specific to the EKF, to deal directly on the EKF state (in a broad sense, i.e., the mean and covariance) and to represent a natural characteristic we would want for such a filter in a very concrete way.

We start with the following definition the unobservable transformation of the filter model.

Definition 1. (Unobservable transformation of a model) A transformation – as a function of element of the state space $\mathcal{T} : \mathcal{X} \to \mathcal{X}$ – is an unobservable transformation for discrete model such as (7.8) at a timestamp i, if the iteration of subsequent discrete propagation steps starting for any initial conditions $\mathbf{X}_i^a$ and $\mathbf{X}_i^b = \mathcal{T}(\mathbf{X}_i^a)$ leads to the same output measurement at each subsequent time step:

$$\forall n > i, \quad h(\mathbf{X}_n^a) = h(\mathbf{X}_n^b) \tag{7.27}$$

This definition means that, in what concerns the deterministic part of the model, applying an unobservable transformation to the initial conditions does not change the measurement sequence value in the future time steps.

Regarding the corresponding filter behavior, that property would translate to the innovation sequence of the filter to be the same in the case the filter was initialized with $\mathbf{X}_0^a$ or $\mathbf{X}_0^b$.

However, in an EKF, the initialization also includes the choice of the first covariance. This value is set either by the user, either by a specific initialization process. In the previous definitions, this initial condition on the uncertainty is disregarded entirely. Thus, one can introduce, as done in [Zhang et al., 2017a], the concept of stochastic transform of an EKF state and covariance.

Definition 2. (Stochastic transform of an EKF state) Assume we have an EKF built on a non-linear error $\mathbf{e}$. Let $\mathcal{T}_S$ be a transformation of an element of the state space defined as a two-argument function $\mathcal{T}_S : (\mathcal{X}, \mathbb{R}^N) \to \mathcal{X}$, where the second argument is seen as a stochastic input of the transformation drawn from a centered Gaussian distribution of covariance $\mathbf{\Sigma}$. We will call a stochastic transform of an EKF state at time k the following transformation:

$$\hat{\mathbf{X}}_k \mapsto \mathcal{T}_S(\hat{\mathbf{X}}_k, 0) \tag{7.28}$$

$$\mathbf{\Sigma}_{\mathbf{e}k} \mapsto \mathbf{M}\mathbf{\Sigma}_{\mathbf{e}k}\mathbf{M}^\mathsf{T} + \mathbf{N}\mathbf{\Sigma}\mathbf{N}^\mathsf{T} \tag{7.29}$$

With:

$$\mathbf{M} = \frac{\partial}{\partial \mathbf{e}} \left(\mathcal{T}_S(\hat{\mathbf{X}} \boxplus \mathbf{e}, 0) \boxminus \mathcal{T}_S(\hat{\mathbf{X}}, 0) \right) \Big|_{\mathbf{e}=0} \tag{7.30}$$

$$\mathbf{N} = \frac{\partial}{\partial \eta} \left(\mathcal{T}_S(\hat{\mathbf{X}}, \eta) \boxminus \mathcal{T}_S(\hat{\mathbf{X}}, 0) \right) \Big|_{\eta=0} \tag{7.31}$$

The usefulness of such transformation might not be intuitive. We will thus illustrate it by a concrete example, the case of reference frame change for a running filter.

Let's assume a navigation Kalman filter is tracking the 6DOF $\boldsymbol{\xi}^{\mathcal{A}\leftarrow b}$ position of a rigid body with respect to a reference frame $\mathcal{A}$. But for some reason [5] we would like it to track the position of the same rigid body with respect to a new reference frame, $\mathcal{B}$. Imagine we got an estimate of the transform $\mathcal{T}^{\mathcal{B}\leftarrow\mathcal{A}}$ at the current time but with uncertainty $\mathbf{\Sigma}$. In that case, we would transform the estimate and covariance of the filter in the following way:

$$\hat{\boldsymbol{\xi}}^{\mathcal{A}\leftarrow b} \mapsto \hat{\boldsymbol{\xi}}^{\mathcal{B}\leftarrow\mathcal{A}} \hat{\boldsymbol{\xi}}^{\mathcal{A}\leftarrow b} \tag{7.32}$$

$$\mathbf{\Sigma}_{\mathbf{e}k} \mapsto \mathrm{Adj}_{\hat{\boldsymbol{\xi}}^{\mathcal{A}\leftarrow b}} \mathbf{\Sigma}_{\mathbf{e}k} \mathrm{Adj}_{\hat{\boldsymbol{\xi}}^{\mathcal{A}\leftarrow b}}^\mathsf{T} + \mathbf{\Sigma}, \quad (\mathrm{Adj}_{\hat{\boldsymbol{\xi}}^{\mathcal{A}\leftarrow b}} \text{is the adjoint in SE(3)}) \tag{7.33}$$

[5] One possible reason would be switching the filter to a "rendezvous" maneuver mode between a robot and a moving base station, that would require switching to a reference frame in which the moving base station is fixed at some point.

This gives an intuitive idea of the transformation involved by the stochastic transform of an EKF state and covariance.

Finally, the main definition introduced by [Zhang et al., 2017a] is written hereafter.[6] It gives a meaning to the invariance to a stochastic unobservable transform.

Definition 3. (Invariance of an EKF output to unobservable stochastic transform) The EKF output is said to be invariant to an unobservable stochastic transformation if both following statements are true:

1. For all $\eta \in \mathbb{R}^N$ the stochastic transform $\mathcal{T}_S : (\mathcal{X}, \eta) \to \mathcal{X}$ describes a unobservable transformation of the model/output on which the EKF is based on.
2. For any two estimate and covariance of the EKF at time i, say $(\hat{\mathbf{X}}_i^a, {\boldsymbol{\Sigma}_{\mathbf{e}_i}}^a)$ and $(\hat{\mathbf{X}}_i^b, {\boldsymbol{\Sigma}_{\mathbf{e}_i}}^b)$, so that b-quantities are computed from the stochastic transformation of a-quantities, we have equality of output sequence of two instances of the filter using respectively a- and b-quantities as initialization values:
$$\forall n > i,\ h(\hat{\mathbf{X}}_n^a) = h(\hat{\mathbf{X}}_n^b) \tag{7.34}$$

The invariance of an EKF output to unobservable stochastic transform is one property we would expect from a filter to respect symmetry of the original model. Intuitively, this last definition states that, whatever are the unobservable quantities values initialized to, and whatever are **the initial uncertainty along the unobservable direction** in the initial covariance, the innovation sequence of the filter should not change.

In the case of our MVINS model, this means that all the estimated trajectories with same data but different initialization values would be related to a reference one by a rotation around gravity and an arbitrary translation. One of the corollaries is that a filter respecting Definition 3 should not have its mean estimate sequence modified by changing the initial heading uncertainty solely. The next section will show that this is not true for filter presented in Chapter 6.

7.2.2.3 The filter of the previous chapter is not invariant to unobservable stochastic transform

We analyze the same data of trajectory depicted in Figure 7.1, but we now look at the position estimate of the different filter instances. Figure 7.2 shows that the mean estimate of the filter changes with the initial uncertainty of the filter along the unobservable heading direction. Again, the effect appears as soon as the filter starts using visual information.

7.2.2.4 Conclusion

The filter of Chapter 6 does not fully mimic the invariance of the original system, and this comes mainly from how the visual measurements are handled. The next section will be dedicated to a new filter that is invariant to unobservable stochastic transforms of the MVINS model.

7.3 A New Filter with Invariance Properties

7.3.1 Literature Study on EKF Invariance Issues

7.3.1.1 Full Batch Optimization/Bundle Adjustment

The demonstrated lack of invariance to unobservable stochastic transformations of the filter of Chapter 6 comes from errors induced by the linearization process implied by the EKF methodology. The authors of [Huang et al., 2008] demonstrated that this stems mainly from the fact some

[6] Note that if we rewrote [Zhang et al., 2017a] definitions slightly to give them more context, Definition 3 is exactly equivalent to their definition of unobservable stochastic transform

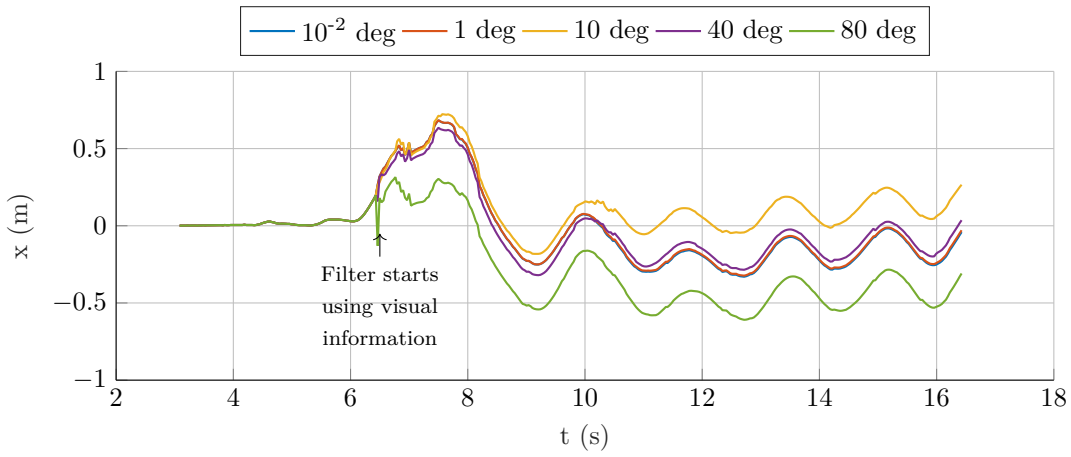


Figure 7.2: X position during initialization with various initial covariance on unobservable heading for filter of Chapter 6. All other initialization values fixed, the X position depends on the initial heading covariance, showing that the filter does not respect the property of invariance to stochastic unobservable transform.

variables are used in several measurements but with two different linearization points. A full batch optimization, or full bundle adjustment, relinearizes the cost function around the estimate of the entire history of states at each iteration, and does not suffer from these source of spurious observability.

However, these batch optimizations methods are far too heavy for real-time purposes, and practical real-time estimators marginalize past poses similarly to what is done in Chapter 5. This marginalization could lead to the same inconsistencies as filtering, as for instance noted by [Engel et al., 2018].

On another hand, we are aware of two attempts to solve incrementally, efficiently, and consistently the full batch optimization problem: one presented in [Michael Kaess et al., 2012] and the other presented in [Keivan et al., 2016]. The first one relies on independence structure of the constraint graph, partial marginalization, and controlled relinearizations, the second one relies on conditioning on some state and a sliding window with adaptive size. Nevertheless, these methods are still more computationally intensive than pure filtering.

7.3.1.2 First Jacobian estimate

For VINS filters, spurious observability of the yaw can be related to linearizations of the same variable for two different updates of the filter. Hence, a workaround is to fix the linearization point of a variable at the first linearization point needed by the algorithm. In practice, this means, all the subsequent Jacobian with respect to this variable will be computed at this linearization point, even if the estimate of this variable was refined since.

This idea was introduced for an EKF-SLAM based VINS system in [Huang et al., 2009] and for MSCKF in [Li and Mourikis, 2012]. This idea was also used in an optimization context in [Engel et al., 2018], but using image intensity error term instead of geometrical reprojection error; note that, as these authors do not use an IMU, the full device orientation is unobservable in their case, in contrast to our work of Chapter 5. We also have exploited the same idea in Chapter 5 within a reprojection error minimization framework, and using IMU and magnetic data.

One issue with this workaround is that it is not clear what is lost by not using the last – and hopefully more accurate – linearization point, compared to what is gained by the improved consistency.

7.3.1.3 Observability-constrained EKF

In our opinion, the theoretical and empirical evidence supporting the approaches reviewed in the two preceding sections are not totally convincing. Therefore, we have considered another piece of work which focuses on reparametrization.

Another idea, introduced by [Huang et al., 2010] and adapted to EKF-SLAM VINS and MSCKF in [Hesch et al., 2013], is to modify artificially the transition matrix and the measurement matrix in order to enforce the filter invariance to stochastic transform property. This method is referred as OC-EKF (Observability Constrained EKF) in the literature.

The advantage of the OC-EKF method compared to the previous approach is that, at each time, the most recent estimate point is used as linearization point. The main disadvantage comes from the fact that the modification of the transition and measurement matrices, albeit being small in general case in the sense of Frobenius norm, is not strongly mathematically founded, and could have unexpected results, especially in the case where the filter estimate is far from the real estimate.

Also, the proofs or empirical evidence of the usefulness of the first-jacobian-estimate and observability-constrained methods found in the literature are not totally convincing in our opinion.

7.3.1.4 Influence of Parametrization of State

The influence of parametrization on the consistency property of the filter has been the subject of some research recently. The authors of [Sola, 2010] study the effect of parametrization on the consistency of the EKF-SLAM algorithm. It has also been known for a long time that the robocentric parametrization – in which features are expressed relatively to current camera pose – improves consistency. (see for instance [Castellanos et al., 2007, 2004]). The paper [Huang et al., 2014] claims to improve consistency by parameterizing the rotation error in the *world* frame instead of the *body* frame.

However, the most promising work regarding this idea is arising from the theory of invariance of estimators ([Martin and Salaun, 2007; Bonnabel et al., 2008]). These ideas were since applied successfully to famous EKF tool; for instance in [Bonnabel and Barrau, 2017]. This line of research attempts to exploit the geometrical symmetries of the problem to improve the filter consistency and behavior. It has been recently applied to SLAM problems in [Barrau and Bonnabel, 2015], where the author show one can solve the false observability issue with an elegant mathematical framework and without relying on previously presented Observability-Constraint hack. Valuable information about the methodology applied to obtain these results can be found in the Ph.D. thesis of Axel Barrau : [Barrau, 2015]. This "invariant" parametrization was after extended to the case of VINS in [Zhang et al., 2017a].

Other interesting work can be found in [Brossard et al., 2017] that explores these ideas in the context of Unscented Kalman Filter, and in [Robert and Perrot, 2017] that applied them in the context of an industrial-grade navigation system.

We decided to take inspiration from this last line of research because of its more theoretically grounded framework. We thus applied the idea of Lie group based parametrization of the error of [Barrau and Bonnabel, 2015] to the particular case of our MVINS system.

7.3.2 Invariant Kalman Filter for MVINS has no guaranteed convergence properties

We take inspiration from the IEKF (Invariant-EKF) theory introduced in [Bonnabel and Barrau, 2017]. In this work, they demonstrate that, for a certain class of systems and errors, the the IEKF achieves provable *local asymptotically stability*: a deterministic property the EKF does not provide in general. This property basically guarantees that the filter can not diverge, if initialized close to the real trajectory, whatever the actual trajectory followed by the system.

In order to use their proven result, we will, solely for this subsection, use the continuous model through the process function f_c of the MI-DR *model. These equation were given in (1.4)-(1.6),*

Page 14 and (2.18), Page 23.

Unfortunately, as we will show in the following, even without including the biases estimate in the state as done in IEKF theory, the MI-DR continuous time model does not fit the framework of [Bonnabel and Barrau, 2017].

The key point of the framework is to exhibit a non-linear error $\boldsymbol{\epsilon}(t)$ which relates the estimate $\hat{\mathbf{X}}$ tracked by the filter and the real value $\mathbf{X}$ and that follows an autonomous propagation:

$$\dot{\boldsymbol{\epsilon}}(t) = g(\boldsymbol{\epsilon}(t), \mathbf{u}(t)), \tag{7.35}$$

In other words, the evolution of the chosen non-linear error solely depends on the input and the error (and does *not* depend on the estimate or the real trajectory). In order to find this error, [Bonnabel and Barrau, 2017] suggest relying on the system symmetry through the use of Lie group structure of the state. They show that for a state embedded in a *matrix Lie group* $\mathcal{G}$ for which the continuous model writes:

$$\frac{d\mathbf{X}(t)}{dt} = f_c(\mathbf{X}(t), \mathbf{u}(t)) \tag{7.36}$$

with $\mathbf{X} \in \mathcal{G}$, the errors defined by $\boldsymbol{\epsilon}^R = \mathbf{X}\hat{\mathbf{X}}^{-1}$ ("right-invariant" error) and $\boldsymbol{\epsilon}^L = \hat{\mathbf{X}}^{-1}\mathbf{X}$ ("left-invariant" error) follow an autonomous propagation if and only if the propagation model verifies the following equality:

$$\begin{aligned} &\forall t > 0, \mathbf{X}, \mathbf{Y} \in \mathcal{G}, \\ &\quad f_c\left(\mathbf{X}\mathbf{Y}, \mathbf{u}(t)\right) = f_c\left(\mathbf{X}, \mathbf{u}(t)\right)\mathbf{Y} + \mathbf{X} f_c\left(\mathbf{Y}, \mathbf{u}(t)\right) - \mathbf{X} f_c\left(\mathbf{I}_n, \mathbf{u}(t)\right)\mathbf{Y} \end{aligned} \tag{7.37}$$

Moreover, assuming that the measurement process can be written in the following form:

$$h(\mathbf{X}) = \mathbf{X}^{-1}\mathbf{d} \tag{7.38}$$

with $\mathbf{d}$ a known vector, the authors prove, under reasonable assumptions similar to the linear Kalman filter case, the local stability of the IEKF built on the error $\mathbf{e}^R$. This is a sound mathematical result which, to our knowledge, has no equivalent in the EKF literature.

7.3.2.1 Another Matrix Lie group embedding for the MI-DR state

We thus study in this chapter filters based on the following matrix Lie group embedding, named $SE_3(3)$ by the authors of [Bonnabel and Barrau, 2017], of the MI-DR state (visual information are disregarded in in this section):

$$\mathbf{X} = \begin{bmatrix} \mathtt{R}^w & \mathbf{v}^w & \mathbf{p}^w & \mathbf{B}^w \\ \mathbf{0}_{3\times 1} & & \mathbf{I}_{3\times 3} & \end{bmatrix} \quad \mathtt{R}^w \in \mathrm{SO}(3), \mathbf{v}^w, \mathbf{p}^w, \mathbf{B}^w \in \mathbb{R}^3 \tag{7.39}$$

$$\mathbf{X}^{-1} = \begin{bmatrix} \mathtt{R}^{w\mathsf{T}} & -\mathtt{R}^{w\mathsf{T}}\mathbf{v}^w & -\mathtt{R}^{w\mathsf{T}}\mathbf{p}^w & -\mathtt{R}^{w\mathsf{T}}\mathbf{B}^w \\ \mathbf{0}_{3\times 1} & & \mathbf{I}_{3\times 3} & \end{bmatrix} \tag{7.40}$$

The form of the magnetic observation in this formalism follows (7.38):

$$h(\mathbf{X}) = -\mathtt{R}^{w\mathsf{T}}\mathbf{B}^w = \mathbf{X}^{-1}\begin{bmatrix} \mathbf{0}_{5\times 1} \\ -1 \end{bmatrix}, \tag{7.41}$$

which makes us focus on the "right-invariant" error that writes:

$$\boldsymbol{\epsilon} = \mathbf{X}\hat{\mathbf{X}}^{-1} = \begin{bmatrix} \mathtt{R}^w\hat{\mathtt{R}}_k^{w\mathsf{T}} & \mathbf{v}^w - \mathtt{R}^w\hat{\mathtt{R}}^{w\mathsf{T}}\hat{\mathbf{v}}^w & \mathbf{p}^w - \mathtt{R}^w\hat{\mathtt{R}}^{w\mathsf{T}}\hat{\mathbf{p}}^w & \mathbf{B}^w - \mathtt{R}^w\hat{\mathtt{R}}_k^{w\mathsf{T}}\hat{\mathbf{B}}^w \\ \mathbf{0}_{3\times 1} & & \mathbf{I}_{3\times 3} & \end{bmatrix} \tag{7.42}$$

This error can actually be associated to a vector error through the vectorized matrix Lie group logarithm $\mathrm{Log}_{SE_3(3)}(\epsilon)$.

If we were trying to build an IEKF based on this error, the following simple computations prove that condition (7.37) is not verified. This is because of the magnetic prediction equation as shown hereafter.

Recall that the continuous model of MI-DR writes:

$$\dot{\mathtt{R}}^w = \mathtt{R}^w[\boldsymbol{\omega}^b]_\times, \tag{7.43}$$
$$\dot{\mathbf{v}}^w = \mathtt{R}^w\mathbf{a}^b + \mathbf{g}^w, \tag{7.44}$$
$$\dot{\mathbf{p}}^w = \mathbf{v}^w, \tag{7.45}$$
$$\dot{\mathbf{B}}^w = \mathtt{R}^w\nabla\mathbf{B}^b\mathtt{R}^{w\mathsf{T}}\mathbf{v}^w. \tag{7.46}$$

It can be written in matrix form (we drop frame exponent hereafter for readability):

$$\dot{\mathbf{X}} = f_c(\mathbf{X},\mathbf{u}) = \begin{bmatrix} \mathtt{R}\,[\boldsymbol{\omega}]_\times & \mathtt{R}\mathbf{a}+\mathbf{g} & \mathbf{v} & \mathtt{R}\nabla\mathbf{B}\mathtt{R}^T\mathbf{v} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{7.47}$$

In order to verify (7.37), we proceed by separating the process function f_c in three terms:

$$f(\mathbf{X},\mathbf{u}) = \mathbf{X}\begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{a} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{g} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{v} & \mathtt{R}\nabla\mathbf{B}\mathtt{R}^T\mathbf{v} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{7.48}$$

The first term is right-invariant while the second term is stationary, both verify easily the characterization of (7.37). We will focus on the last one, that requires a bit more work.

The left-hand side of (7.37) is:

$$f(\mathbf{X}_1\mathbf{X}_2,\mathbf{u}(t)) = \begin{bmatrix} \mathbf{0}_{3\times2} & \mathtt{R}_1\mathbf{v}_2+\mathbf{v}_1 & \mathtt{R}_1\mathtt{R}_2\nabla\mathbf{B}(\mathtt{R}_1\mathtt{R}_2)^T(\mathbf{v}_1+\mathtt{R}_1\mathbf{v}_2) \\ \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{7.49}$$

While its right-hand side writes:

$$f(\mathbf{X}_1,\mathbf{u}(t))\mathbf{X}_2 + \mathbf{X}_1 f(\mathbf{X}_2,\mathbf{u}(t)) - \mathbf{X}_1 f(\mathbf{I}_n,\mathbf{u}(t))\mathbf{X}_2 \tag{7.50}$$
$$= \begin{bmatrix} \mathbf{0}_{3\times2} & \mathtt{R}_1\mathbf{v}_2 & \mathtt{R}_1\mathtt{R}_2\nabla\mathbf{B}(\mathtt{R}_2)^T\mathbf{v}_2 \\ \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times2} & \mathbf{v}_1 & \mathtt{R}_1\nabla\mathbf{B}(\mathtt{R}_1)^T\mathbf{v}_1 \\ \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} + \mathbf{0}_{6\times6} \tag{7.51}$$

The difference of the two sides does not give zero but instead the following matrix :

$$\begin{bmatrix} \mathbf{0}_{3\times3} & \mathtt{R}_1(\mathtt{R}_2\nabla\mathbf{B}\mathtt{R}_2^\mathsf{T} - \nabla\mathbf{B})(\mathtt{R}_1)^T\mathbf{v}_1 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{7.52}$$

We conclude that, for MVINS system, we can *not* build autonomous equation the same way it was done in [Bonnabel and Barrau, 2017]; nor leverage associated stability property of the IEKF. And, contrarily to the pure VINS model, this is not because of the biases of the sensors, but because of the "idealized sensor" continuous model.

Note also, that, for readability, we only handle the magneto-inertial dynamic and measurement here. However, we would draw the same conclusion for the full MVINS system, including poses states from the stochastic cloning of the MSCKF technique.

Nevertheless, the idea of this new parametrization is retained: as demonstrated in [Barrau and Bonnabel, 2015], this idea stays interesting from a consistency point of view. We will exhibits a filter that has the property of invariance to unobservable stochastic transform for the MVINS model, extending the application of this idea to VINS that was already presented in [Zhang et al., 2017a]. In the next sections, we do focus on the particular case of MVINS model; the reader interested by the theory could refer to [Barrau, 2015] for more general results and background on invariant parametrization applied to navigation systems.

7.3.3 A Right-Invariant EKF invariant to Unobservable Stochastic Transformation

We leverage the "right-invariant" error presented in the previous section. But we add to the state the estimated biases so that the matrix Lie group used to form the error is now:

$$\mathbf{X} = \begin{bmatrix} \mathtt{R} & \mathbf{v} & \mathbf{p} & \mathbf{B} & & \\ 0 & 1 & 0 & 0 & \multicolumn{3}{c}{\multirow{3}{*}{$\mathbf{0}_{6\times5}$}} \\ 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 1 & & \\ & & & & \mathbf{I}_3 & \mathbf{b}_g & \mathbf{b}_{a1} \\ \multicolumn{4}{c}{\multirow{3}{*}{$\mathbf{0}_{5\times6}$}} & 0 & 1 & 0 \\ & & & & 0 & 0 & 1 \end{bmatrix} \tag{7.53}$$

We will name the matrix Lie group that contains the matrix element of this form with the symbol $\mathcal{M}$.[7] Also, in order to lighten the notation in the following derivation we will write groups elements, group product and group inverse in a reduced form, dropping the trivial term of the matrix:

$$\mathbf{X}_1 = \begin{pmatrix} \mathtt{R}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_1 \\ \mathbf{B}_1 \\ \mathbf{b}_{g1} \\ \mathbf{b}_{a1} \end{pmatrix}, \quad \mathbf{X}_1\mathbf{X}_2 = \begin{pmatrix} \mathtt{R}_1\mathtt{R}_2 \\ \mathbf{v}_1 + \mathtt{R}_1\mathbf{v}_2 \\ \mathbf{p}_1 + \mathtt{R}_1\mathbf{p}_2 \\ \mathbf{B}_1 + \mathtt{R}_1\mathbf{B}_2 \\ \mathbf{b}_{g1} + \mathbf{b}_{g2} \\ \mathbf{b}_{a1} + \mathbf{b}_{a2} \end{pmatrix}, \quad \mathbf{X}_2^{-1} = \begin{pmatrix} \mathtt{R}_2^T \\ \mathtt{R}_2^T\mathbf{v}_2 \\ \mathtt{R}_2^T\mathbf{p}_2 \\ \mathtt{R}_2^T\mathbf{B}_2 \\ -\mathbf{b}_{g2} \\ -\mathbf{b}_{a2} \end{pmatrix}, \quad \mathbf{X}_1, \mathbf{X}_2 \in \mathcal{M} \tag{7.54}$$

Its Lie algebra $\mathfrak{m}$ is the set of matrices of the following form:

$$\mathbf{e}^\wedge = \begin{bmatrix} [\mathbf{e}_\theta]_\times & \mathbf{e}_\mathbf{v} & \mathbf{e}_\mathbf{p} & \mathbf{e}_\mathbf{B} & & \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \multicolumn{3}{c}{\multirow{3}{*}{$\mathbf{0}_{6\times5}$}} \\ & & & & & & \\ \multicolumn{4}{c}{\multirow{3}{*}{$\mathbf{0}_{5\times6}$}} & & & \\ & & & & \mathbf{0}_{3\times3} & \mathbf{e}_{\mathbf{b}_g} & \mathbf{e}_{\mathbf{b}_a} \\ & & & & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix}, \quad \mathbf{e}^\wedge \in \mathfrak{m} \tag{7.55}$$

with $\mathbf{e} = \begin{bmatrix} \mathbf{e}_\theta^\mathsf{T} & \mathbf{e}_\mathbf{v}^\mathsf{T} & \mathbf{e}_\mathbf{p}^\mathsf{T} & \mathbf{e}_\mathbf{B}^\mathsf{T} & \mathbf{e}_{\mathbf{b}_g}^\mathsf{T} & \mathbf{e}_{\mathbf{b}_a}^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{18}$

We will use the exponential operator on this Lie group to associate an element of the group to an element of its Lie algebra (on matrix Lie group it is equivalent to matrix exponential).

In order to keep notation succinct, we will use the shortcut $e^{\mathbf{e}_\mathtt{R}}$ for $\mathrm{Exp}_{\mathrm{SO}(3)}(\mathbf{e}_\mathtt{R})$ and $\exp(\mathbf{e}^\wedge)$ for $\mathrm{Exp}_\mathcal{M}(\mathbf{e}^\wedge)$. One has:

$$\exp(\mathbf{e}^\wedge) = \begin{pmatrix} e^{\mathbf{e}_\mathtt{R}} \\ \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{v} \\ \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{p} \\ \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{B} \\ \mathbf{e}_{\mathbf{b}_g} \\ \mathbf{e}_{\mathbf{b}_a} \end{pmatrix} \tag{7.56}$$

Where $\mathbb{R}^3 \to \mathbb{R}^{3\times3} : \theta \mapsto \mathbf{J}_r(\theta)$ is the *right jacobian of* $\mathrm{SO}(3)$ [8], a function defined as:

$$\mathbf{J}_r(\theta) = \mathbf{I}_3 - \frac{1-\cos\|\theta\|}{\|\theta\|^2}[\theta]_\times + \frac{\|\theta\| - \sin\|\theta\|}{\|\theta\|^3}[\theta]_\times^2 \tag{7.58}$$

[7] $\mathcal{M}$ is used here as a shortcut for the group $SE_3(3) \times \mathbb{R}^6$ with notations of [Bonnabel and Barrau, 2017]

[8] this Jacobian is more often encountered in the exponential map of SE(3) group:

$$\mathrm{Exp}_{\mathrm{SE}(3)}\left(\mathbf{e}_\mathtt{R}^\mathsf{T}, \mathbf{e}_\mathbf{p}^\mathsf{T}\right) = \begin{bmatrix} \mathbf{e}^{\mathbf{e}_\mathtt{R}} & \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{p} \\ 0 & 1 \end{bmatrix} \tag{7.57}$$

We use the Lie algebra to define the error by the following $\boxplus$ and $\boxminus$ operators:

$$\mathbf{X}_k = \hat{\mathbf{X}}_k \boxplus \mathbf{e}_k = \exp(\mathbf{e}_k^{\wedge})\hat{\mathbf{X}}_k = \begin{pmatrix} e^{\mathbf{e}_{\mathtt{R}}}\hat{\mathtt{R}}^w \\ \mathbf{J}_r(-\mathbf{e}_{\mathtt{R}})\mathbf{e}_{\mathbf{v}} + e^{\mathbf{e}_{\mathtt{R}}}\hat{\mathbf{v}}^w \\ \mathbf{J}_r(-\mathbf{e}_{\mathtt{R}})\mathbf{e}_{\mathbf{p}} + e^{\mathbf{e}_{\mathtt{R}}}\hat{\mathbf{p}} \\ \mathbf{J}_r(-\mathbf{e}_{\mathtt{R}})\mathbf{e}_{\mathbf{B}} + e^{\mathbf{e}_{\mathtt{R}}}\hat{\mathbf{B}}^w \\ \mathbf{e}_{\mathbf{b}_g} + \hat{\mathbf{b}_g} \\ \mathbf{e}_{\mathbf{b}_a} + \hat{\mathbf{b}_a} \end{pmatrix} \tag{7.59}$$

$$\mathbf{e}_k = \mathbf{X}_k \boxminus \hat{\mathbf{X}}_k = \mathrm{Log}_{\mathcal{M}}\left(\mathbf{X}_k\hat{\mathbf{X}}_k^{-1}\right) = \begin{pmatrix} \mathbf{e}_{\mathtt{R}} \\ \mathbf{J}_r(-\mathbf{e}_{\mathtt{R}})^{-1}(\mathbf{v} - \mathtt{R}\hat{\mathtt{R}}^T\hat{\mathbf{v}}) \\ \mathbf{J}_r(-\mathbf{e}_{\mathtt{R}})^{-1}(\mathbf{p} - \mathtt{R}\hat{\mathtt{R}}^T\hat{\mathbf{p}}) \\ \mathbf{J}_r(-\mathbf{e}_{\mathtt{R}})^{-1}(\mathbf{B} - \mathtt{R}\hat{\mathtt{R}}^T\hat{\mathbf{B}}) \\ \mathbf{b}_g - \hat{\mathbf{b}_g} \\ \mathbf{b}_g - \hat{\mathbf{b}_a} \end{pmatrix} \in \mathbb{R}^{18} \tag{7.60}$$

$$\text{with } \mathbf{e}_{\mathtt{R}} = \mathrm{Log}_{\mathrm{SO}(3)}\left(\mathtt{R}\hat{\mathtt{R}}^T\right) \tag{7.61}$$

and we build an EKF on $\mathbf{e}_k$ with the methodology presented in the beginning of this chapter.

7.3.3.1 Invariance of the filter to unobservable stochastic transformation

This section now proves that the filter based on the previous error is invariant to any unobservable transform of MVINS model. We draw inspiration from the work presented in [Zhang et al., 2017a] that studied the invariance to stochastic unobservable transform for the VINS model. We mainly use their way of presenting results and proofs and extend them to the MVINS case. We also took care to fix some of their unfortunate inaccuracies in theorem statements and proofs.

We first explicitly parametrize the family of unobservable transform for the model.

Definition 4. (Unobservable stochastic transform for MVINS-DR model) We parametrize the family of unobservable stochastic transform for the model in the following way:

$$\mathcal{T}(\mathbf{X}, \theta, \eta) \overset{\text{def}}{=} \begin{pmatrix} e^{(\eta_1+\theta_1)\mathbf{g}}\mathtt{R} \\ e^{(\eta_1+\theta_1)\mathbf{g}}\mathbf{v} \\ e^{(\eta_1+\theta_1)\mathbf{g}}\mathbf{p} + \theta_{2:4} + \eta_{2:4} \\ e^{(\eta_1+\theta_1)\mathbf{g}}\mathbf{B} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{pmatrix}, \quad \text{with } \mathbf{X} = \begin{pmatrix} \mathtt{R} \\ \mathbf{v} \\ \mathbf{p} \\ \mathbf{B} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{pmatrix} \in \mathcal{M}, \eta \in \mathbb{R}^4, \theta \in \mathbb{R}^4 \tag{7.62}$$

Which spans the set of composition of rotation around the gravity vector and global translation of world coordinates.

Note that we can decompose this unobservable stochastic transform into two transforms that the authors [Zhang et al., 2017a] call *stochastic identity transform* $\mathcal{T}(\hat{\mathbf{X}}, 0, \eta)$ and *deterministic transform* $\mathcal{T}(\hat{\mathbf{X}}, \theta, 0)$ the following way:

$$\mathcal{T}(\hat{\mathbf{X}}, \theta, \eta) = \mathcal{T}\left(\mathcal{T}(\hat{\mathbf{X}}, \theta, 0), 0, \eta\right) = \mathcal{T}\left(\mathcal{T}(\hat{\mathbf{X}}, 0, \eta), \theta, 0\right) \tag{7.63}$$

This results stems from the identity $exp_{\mathfrak{so}(3)}(\mathbf{a}+\mathbf{b}) = exp_{\mathfrak{so}(3)}(\mathbf{a})exp_{\mathfrak{so}(3)}(\mathbf{b})$ if $\mathbf{a}$ and $\mathbf{b}$ are collinear.

We will use this decomposition to prove that the filter is invariant to all unobservable stochastic transform: we first prove that it is invariant to the family of (deterministic) transform $\mathcal{T}(\cdot\,, \theta, 0)$ then prove that it is invariant to the family of stochastic transform at identity $\mathcal{T}(\cdot\,, 0, \eta)$.

Invariance to $\mathcal{T}(\cdot\,,\theta,0)$ We will first exhibit a sufficient condition for the invariance to $\mathcal{T}(\cdot\,,\theta,0)$, and then show that our parametrization fulfills this condition.

Property 1. *The output of an EKF for the* MVINS *model is invariant under $\mathcal{T}(\cdot\,,\theta,0)$ if there exists a constant invertible matrix $\mathbf{W}_\theta$ such that:*

$$\forall \mathbf{e}, \forall \hat{\mathbf{X}}, \quad \mathcal{T}(\hat{\mathbf{X}} \boxplus \mathbf{e}, \theta, 0) = \mathcal{T}(\hat{\mathbf{X}}, \theta, 0) \boxplus \mathbf{W}_\theta \mathbf{e} \tag{7.64}$$

Proof. Let us consider an EKF built on the model with a choice of error verifying (7.64) and let us assume that two such filters are running simultaneously. The first starts from the initial estimate $(\hat{\mathbf{X}}_i, \mathbf{\Sigma}_{\hat{\mathbf{X}}_i})$ at time i. The second starts from the initial estimate $(\hat{\mathbf{Y}}_i, \mathbf{\Sigma}_{\hat{\mathbf{Y}}_i}) = (\mathcal{T}(\hat{\mathbf{X}}_i, \theta, 0), \mathbf{W}_\theta \mathbf{\Sigma}_{\hat{\mathbf{X}}_i} \mathbf{W}_\theta^\mathsf{T})$, a deterministic transform of the first one. We will show that after one propagation and one update, the two filter estimate and covariance are still related one to the other with the same unobservable deterministic transform. Conclusion will be drawn by recursion.

The propagation yields for the first filter prediction:

$$\begin{cases} \hat{\mathbf{X}}_{i+1|i} = f(\hat{\mathbf{X}}_i, \mathbf{u}_i, 0) \\ \mathbf{\Sigma}_{\hat{\mathbf{X}}_{i+1|i}} = \Phi_{\hat{\mathbf{X}}_{i+1}} \mathbf{\Sigma}_{\hat{\mathbf{X}}_i} {\Phi_{\hat{\mathbf{X}}_{i+1}}}^\mathsf{T} + \mathbf{Gc}_{\hat{\mathbf{X}}_{i+1}} \mathbf{\Sigma}_{\mathbf{u},\eta} {\mathbf{Gc}_{\hat{\mathbf{X}}_{i+1}}}^\mathsf{T} \end{cases} \tag{7.65}$$

(where we condensed the noise from measurement noise and stochastic model matrices $\mathbf{Gc} = \mathrm{diag}(\mathbf{G}_{k+1,k}^{\mathrm{mimu}}, \mathbf{C}_{k+1,k}^{\mathrm{mimu}})$ and $\mathbf{\Sigma}_{\mathbf{u},\eta} = \mathrm{diag}(\mathbf{\Sigma}_\mathbf{u}, \mathbf{\Sigma}_\eta)$.)

While for the second, one gets,

$$\begin{cases} \hat{\mathbf{Y}}_{i+1|i} = f(\mathcal{T}(\hat{\mathbf{X}}_i, \theta, 0), \mathbf{u}_i, 0) \\ \mathbf{\Sigma}_{\hat{\mathbf{Y}}_{i+1|i}} = \Phi_{\hat{\mathbf{Y}}_{i+1}} \mathbf{W}_\theta \mathbf{\Sigma}_{\hat{\mathbf{X}}_i} \mathbf{W}_\theta^\mathsf{T} {\Phi_{\hat{\mathbf{Y}}_{i+1}}}^\mathsf{T} + \mathbf{Gc}_{\hat{\mathbf{Y}}_{i+1}} \mathbf{\Sigma}_{\mathbf{u},\eta} {\mathbf{Gc}_{\hat{\mathbf{Y}}_{i+1}}}^\mathsf{T} \end{cases} \tag{7.66}$$

As, in the MVINS model the function f always commutes with $\mathcal{T}$ (this is seen intuitively as $\mathcal{T}$ corresponds solely to a change of reference frame that does not affect the model), the second filter prediction writes equivalently:

$$\begin{cases} \hat{\mathbf{Y}}_{i+1|i} = \mathcal{T}(\hat{\mathbf{X}}_{i+1}, \theta, 0) \\ \mathbf{\Sigma}_{\hat{\mathbf{Y}}_{i+1|i}} = \Phi_{\hat{\mathbf{Y}}_{i+1}} \mathbf{W}_\theta \mathbf{\Sigma}_{\hat{\mathbf{X}}_i} \mathbf{W}_\theta^\mathsf{T} {\Phi_{\hat{\mathbf{Y}}_{i+1}}}^\mathsf{T} + \mathbf{Gc}_{\hat{\mathbf{Y}}_{i+1}} \mathbf{\Sigma}_{\mathbf{u},\eta} {\mathbf{Gc}_{\hat{\mathbf{Y}}_{i+1}}}^\mathsf{T} \end{cases} \tag{7.67}$$

According to the definition of transition matrices Φs and measurement/stochastic model Jacobians , we can show that:

$$\Phi_{\hat{\mathbf{Y}}_{i+1}} = \mathbf{W}_\theta \Phi_{\hat{\mathbf{X}}_{i+1}} \mathbf{W}_\theta^{-1} \qquad\qquad \mathbf{Gc}_{\hat{\mathbf{Y}}_{i+1}} = \mathbf{W}_\theta \mathbf{Gc}_{\hat{\mathbf{X}}_{i+1}} \tag{7.68}$$

So that the second filter prediction at time $i+1$ also writes:

$$\begin{cases} \hat{\mathbf{Y}}_{i+1|i} = \mathcal{T}(\hat{\mathbf{X}}_{i+1}, \theta, 0) \\ \mathbf{\Sigma}_{\hat{\mathbf{Y}}_{i+1|i}} = \mathbf{W}_\theta \mathbf{\Sigma}_{\hat{\mathbf{X}}_{i+1|i}} \mathbf{W}_\theta^\mathsf{T} \end{cases} \tag{7.69}$$

The second part of this proof inspects how this second filter estimates are transformed by the filter update step. Going back to the definition of the measurement matrices $\mathbf{H}$s, and Kalman gain $\mathbf{K}$s of Equations (7.16) and (7.17), Page 141, we can show that:

$$\mathbf{H}_{\hat{\mathbf{Y}}_{i+1}} = \mathbf{H}_{\hat{\mathbf{X}}_{i+1}} \mathbf{W}_\theta^{-1} \tag{7.70}$$

$$\mathbf{K}_{\hat{\mathbf{Y}}_{i+1}} = \mathbf{W}_\theta \mathbf{K}_{\hat{\mathbf{X}}_{i+1}} \tag{7.71}$$

So that if we compute the update equation for the second filter, we have:

$$\hat{\mathbf{Y}}_{i+1} = \hat{\mathbf{Y}}_{i+1|i} \boxplus \mathbf{K}_{\hat{\mathbf{Y}}_{i+1}} \mathbf{z} \tag{7.72}$$

$$= \mathcal{T}\left(\hat{\mathbf{X}}_{i+1|i}, \theta, 0\right) \boxplus \mathbf{W}_\theta \mathbf{K}_{\hat{\mathbf{X}}_{i+1}} \mathbf{z} \tag{7.73}$$

$$\text{(Using the hypothesis of the property to be demonstrated:)} \tag{7.74}$$

$$= \mathcal{T}\left(\hat{\mathbf{X}}_{i+1|i} \boxplus \mathbf{K}_{\hat{\mathbf{X}}_{i+1}} \mathbf{z}, \theta, 0\right) \tag{7.75}$$

$$= \mathcal{T}\left(\hat{\mathbf{X}}_{i+1}, \theta, 0\right) \tag{7.76}$$

and, regarding the covariance:

$$\tag{7.77}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_{i+1}} = (\mathbf{I} - \mathbf{K}_{\hat{\mathbf{Y}}_{i+1}} \mathbf{H}_{\hat{\mathbf{Y}}_{i+1}^\intercal}) \boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_{i+1|i}} \tag{7.78}$$

$$= (\mathbf{I} - \mathbf{W}_\theta \mathbf{K}_{\hat{\mathbf{X}}_{i+1}} \mathbf{H}_{\hat{\mathbf{X}}_{i+1}^\intercal} \mathbf{W}_\theta^{-1}) \mathbf{W}_\theta \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_{i+1}} \mathbf{W}_\theta^\intercal \tag{7.79}$$

$$= \mathbf{W}_\theta \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_{i+1}} \mathbf{W}_\theta^\intercal \tag{7.80}$$

We conclude that after propagation, the second filter mean and covariance estimates at time $i+1$ are transformed from those of the first filter the same way than they were at time i. By recursion, we deduce that:

$$\forall n > i, \hat{\mathbf{Y}}_{n+1|n} = \mathcal{T}(\hat{\mathbf{X}}_{n+1|n}, \theta, 0) \tag{7.81}$$

Which induces that $h(\hat{\mathbf{Y}}_{n+1|n}) = h(\hat{\mathbf{X}}_{n+1|n})$ for all $n > i$, which concludes the proof.
■

The condition (7.64) is verified for our parametrization with:

$$\mathbf{W}_\theta = \begin{bmatrix} e^{\theta_1 \mathbf{g}} & 0 & 0 & 0 & 0 \\ 0 & e^{\theta_1 \mathbf{g}} & 0 & 0 & 0 \\ [\theta_{2:4}]_\times e^{\theta_1 \mathbf{g}} & 0 & e^{\theta_1 \mathbf{g}} & 0 & 0 \\ 0 & 0 & 0 & e^{\theta_1 \mathbf{g}} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_6 \end{bmatrix} \tag{7.82}$$

which proves the invariance to $\mathcal{T}(\cdot\,, \theta, 0)$. The demonstration consists mainly in calculus with SO(3) properties and is rejected in Appendix B.1, Page 171.

Invariance to $\mathcal{T}(\cdot\,, 0, \eta)$ The invariance to stochastic identity transform will be proven leveraging the following property:

Property 2. *The output of an* EKF *for the* MVINS *system is invariant under $\mathcal{T}(\cdot\,, 0, \eta)$ if:*

$$\forall n \text{ and } i \geq 0, \quad \mathbf{H}_{n+i+1} \Phi_{n+i} \Phi_{n+i-1} ... \Phi_i \mathbf{N}_i = 0 \tag{7.83}$$

Proof. Let us consider an EKF built on the model with a error verifying (7.83) and suppose two instances of it are started. The first one from estimate $(\hat{\mathbf{X}}_i, \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_i})$ at time i. The second from estimate $(\hat{\mathbf{X}}_i, \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_i} + \mathbf{N}_i \boldsymbol{\Sigma} \mathbf{N}_i^\intercal)$, an unobservable stochastic transform of the first one. By calling the subsequent estimate of the second filter $\hat{\mathbf{Y}}_n$ and $\boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_n}$, we will show by recursion the fact that:

$$\forall n \geq i, \hat{\mathbf{Y}}_n = \hat{\mathbf{X}}_n \text{ and } \boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_n} = \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n} + \Phi_n \Phi_{n-1} ... \Phi_i \mathbf{N}_i \boldsymbol{\Sigma} \mathbf{N}_i^\intercal \Phi_i^\intercal ... \Phi_{n-1}^\intercal \Phi_n^\intercal \tag{7.84}$$

Which first equality induces that the filter output is invariant to identity stochastic transform.

- Initialization: For $n = i$ it is true by assumption.
- Recursion: Assume for one $n \geq i$ we have:

$$\hat{\mathbf{Y}}_n = \hat{\mathbf{X}}_n \text{ and } \boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_n} = \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n} + \prod_{k=i}^{n}(\Phi_k)\mathbf{N}_i\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_i}\mathbf{N}_i^\mathsf{T}\prod_{k=n}^{i}(\Phi_k^\mathsf{T}) \tag{7.85}$$

Then prediction for the second filter writes:

$$\hat{\mathbf{Y}}_{n+1|n} = \hat{\mathbf{X}}_{n+1|n} = f(\hat{\mathbf{X}}_n) \text{ and} \tag{7.86}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_{n+1|n}} = \Phi_{n+1}\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n}\Phi_{n+1}^\mathsf{T} + \mathbf{G}\mathbf{c}_k\boldsymbol{\Sigma}_{\mathbf{u}_k,\eta_k}\mathbf{G}\mathbf{c}_k{}^\mathsf{T} + \prod_{k=i}^{n+1}(\Phi_k)\mathbf{N}_i\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_i}\mathbf{N}_i^\mathsf{T}\prod_{k=n+1}^{i}(\Phi_k^\mathsf{T}) \tag{7.87}$$

The covariance of innovation writes:

$$\mathbf{S}_{\hat{\mathbf{Y}}_{n+1}} = \mathbf{H}_{n+1}\Phi_{n+1|n}\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n}\Phi_{n+1|n}^\mathsf{T} + \mathbf{G}\mathbf{c}_k\boldsymbol{\Sigma}_{\mathbf{u}_k,\eta_k}\mathbf{G}\mathbf{c}_k{}^\mathsf{T}\mathbf{H}_{n+1}^\mathsf{T} \tag{7.88}$$

$$+ \underbrace{\mathbf{H}_{n+1}\prod_{k=i}^{n+1}(\Phi_k)\mathbf{N}_i\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_i}\mathbf{N}_i^\mathsf{T}\prod_{k=n+1}^{i}(\Phi_k^\mathsf{T})\mathbf{H}_{n+1}^\mathsf{T}}_{0 \text{ (by assumption of (7.83))}} \tag{7.89}$$

$$= \mathbf{H}_{n+1}\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_{n+1|n}}\mathbf{H}_{n+1}^\mathsf{T} \tag{7.90}$$

$$= \mathbf{S}_{\hat{\mathbf{X}}_{n+1}} \tag{7.91}$$

And the Kalman gain writes:

$$\mathbf{K}_{\hat{\mathbf{Y}}_{n+1}} = \boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_{n+1|n}}\mathbf{H}_{n+1}^\mathsf{T}\mathbf{S}_{\hat{\mathbf{X}}_{n+1}}^{-1} \text{ (by previous derivation)} \tag{7.92}$$

$$= \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_{n+1|n}}\mathbf{H}_{n+1}^\mathsf{T}\mathbf{S}_{\hat{\mathbf{X}}_{n+1}}^{-1} \text{ (by recursion assumption)} \tag{7.93}$$

$$= \mathbf{K}_{\hat{\mathbf{X}}_{n+1}} \tag{7.94}$$

Which finally yields:

$$\hat{\mathbf{Y}}_{n+1} = \hat{\mathbf{X}}_{n+1|n} \boxplus \mathbf{K}_{\hat{\mathbf{X}}_{n+1}} \tag{7.95}$$

$$= \hat{\mathbf{X}}_{n+1} \tag{7.96}$$

$$\text{and} \tag{7.97}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_{n+1}} = \left(\mathbf{I} - \mathbf{K}_{\hat{\mathbf{X}}_{n+1}}\right)\boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_n} \tag{7.98}$$

$$= \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_{n+1}} + \prod_{k=i}^{n+1}(\Phi_k)\mathbf{N}_i\boldsymbol{\Sigma}_{\hat{\mathbf{X}}_i}\mathbf{N}_i^\mathsf{T}\prod_{k=n+1}^{i}(\Phi_k^\mathsf{T}) \tag{7.99}$$

We demonstrated by mathematical induction that:

$$\forall n \geq i, \hat{\mathbf{Y}}_n = \hat{\mathbf{X}}_n \text{ and } \boldsymbol{\Sigma}_{\hat{\mathbf{Y}}_n} = \boldsymbol{\Sigma}_{\hat{\mathbf{X}}_n} + \Phi_n\Phi_{n-1}...\Phi_i\mathbf{N}_i\boldsymbol{\Sigma}\mathbf{N}_i^\mathsf{T}\Phi_i^\mathsf{T}...\Phi_{n-1}^\mathsf{T}\Phi_n^\mathsf{T} \tag{7.100}$$

whose first part induces that the filter is invariant to identity stochastic transform.

■

The goal is now to show that in our MSCKF formulation, we indeed, have the Property 2 verified.

Remark: In the Property 2, the measurement $\mathbf{H}$s and transition Φs matrices are those used effectively by the filter, *ie.* linearized at the estimated values, which are, in general, different of the real (and unknown) values. Also, since we use an MSCKF approach in practice, we have to

compute these matrices in the way MSCKF filter does, implying the stochastic cloned states, and the elimination of landmark from the measurement equation.[9]

First, we will compute $\mathbf{N}_i$, as in Definition 2. Secondly, we will compute the structure of the transition matrix Φ (which is different of the transition matrix given in previous chapter Section 6.4.2.1, Page 118 because of the choice of the error used in this section). We will be able to show that $\mathbf{N}_i$ is left unchanged by left multiplication with a member of the family of all possible transition functions Φ. We will finally show that for both magnetic and visual measurement update, the linearized measurement matrices $\mathbf{H}$ involved are so that the Property 2 is verified. We will conclude that the MSCKF filter based on the error (7.61) is invariant to any stochastic unobservable transform of the MVINS system.

(i) Computation of $\mathbf{N}_i$ We recall that the matrix $\mathbf{N}_i$ at time i, is defined as (cf. Definition 2, Page 144):

$$\mathbf{N}_i = \frac{\partial}{\partial \eta} \left(\mathcal{T}(\mathbf{X}_i, 0, \eta) \boxminus \mathcal{T}(\mathbf{X}_i, 0, 0) \right) \Big|_{\eta=0} \tag{7.101}$$

Taking into account the stochastic cloned part of the state. The computation of $\mathbf{N}_i$ leads to:

$$\mathbf{N}_i = \begin{bmatrix} \vdots & \vdots \\ \mathbf{g} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \\ \vdots & \vdots \\ \hline \mathbf{g} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} nc \text{ stochastic clones} \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\} \text{current state} \end{matrix} \tag{7.102}$$

Remarkably, this matrix does not depend on the state estimate, thanks to the choice of the parametrization.

(ii) Structure of transition matrix Φ Computation of Φ_{k+1} is rejected in Appendix B.2. We only need some part of the structure of Φ_{k+1} here.

Recall that in the MSCKF algorithm, the prediction step can also involve stochastic cloning: we write Φ in the following form:

$$\Phi_{k+1} = \begin{bmatrix} \Phi_{k+1}^{Sc1} & \Phi_{k+1}^{Sc2} \\ 0 & \Phi_{k+1}^{\text{mimu}} \end{bmatrix} \tag{7.103}$$

where $[\Phi_{k+1}^{Sc1} \quad \Phi_{k+1}^{Sc2}]$ is a matrix with zeros and ones only whose exact expression depends on if stochastic cloning occurs at time k or not.

Φ_{k+1}^{mimu} structure writes:

$$\Phi_{k+1}^{\text{mimu}} = \left[\begin{array}{cccc|cc} \mathbf{I}_{3\times3} & * & \mathbf{0}_{3\times3} & * & * & * \\ \Delta t_k \left[\mathbf{g}\right]_\times & * & \mathbf{0}_{3\times3} & * & * & * \\ \frac{\Delta t_k^2}{2} \left[\mathbf{g}\right]_\times & * & \mathbf{I}_{3\times3} & * & * & * \\ \mathtt{R}_k \widetilde{\mathbf{\Delta B}}_{\mathbf{g};kk+1} \mathtt{R}_k^T \left[\mathbf{g}\right]_\times & * & \mathbf{0}_{3\times3} & * & * & * \\ \hline \mathbf{0}_{3\times3} & * & \mathbf{0}_{3\times3} & * & * & * \\ \mathbf{0}_{3\times3} & * & \mathbf{0}_{3\times3} & * & * & * \end{array}\right] \tag{7.104}$$

[9] In contrast with [Zhang et al., 2017a], we work here directly on the MSCKF state for proving the property of invariance to unobservable stochastic transform.

With this structure, we verify easily that $\Phi_{k+1}\mathbf{N}_k = \mathbf{N}_k$ and that, by recursion, we have:

$$\forall n \text{ and } i \geq 0, \quad \Phi_{n+i}\Phi_{n+i-1}...\Phi_i\mathbf{N}_i = \mathbf{N}_i \tag{7.105}$$

This is very handy to prove condition (7.83) as it is sufficient to show that $\mathbf{H}\mathbf{N}_i = 0$. We now prove it both for magnetic and visual measurement.

(iii.a) Magnetic update This case is easy. The magnetic update is related directly to the current states (and not to stochastic clones of poses):

$$h_{\text{magn}}(\mathbf{X}_k) = \mathtt{R}_k^T\mathbf{B}_k$$

Computing the first order approximation yields:

$$h_{\text{magn}}(\mathbf{X}_k \boxplus \mathbf{e}) = \mathtt{R}_k^T e^{-\mathbf{e}_\mathtt{R}}\left(e^{\mathbf{e}_\mathtt{R}}\mathbf{B}_k + \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{B}\right)$$
$$h_{\text{magn}}(\mathbf{X}_k \boxplus \mathbf{e}) = h_{\text{magn}}(\mathbf{X}_k) + \mathtt{R}_k^T\mathbf{e}_\mathbf{B} + o(\|\mathbf{e}\|)$$

Thus the measurement matrix to use is:

$$\mathbf{H}_{\text{magn}_k} = \begin{bmatrix}\mathbf{0}_{3\times 6nc} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathtt{R}_k^\mathsf{T} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3}\end{bmatrix}$$

And we have by simple computation:

$$\mathbf{H}_{\text{magn}_k}\mathbf{N}_i = 0 \tag{7.106}$$

This proves the condition (7.83) if we had only the magnetic measurement equation.

(iii.b) Visual update The MI-MSCKF filter processes also visual measurement. For these, showing the relation is more cumbersome by direct computation. This is because the way the landmark position parameters are eliminated from the measurement function in the MSCKF technique. Instead, we propose to leverage the invariance of the reprojection function h_{feat} to demonstrate the result without having to compute $\mathbf{H}_{\text{feat}}$ explicitly.

Using the notation of the introduction of this chapter (7.24), Page 142, we are going to show that $\mathbf{H}_{\text{feat}}\mathbf{N}_i = 0$.

We assume here that landmarks are parameterized by their position $\mathbf{l}$ in *world* frame. One trick is to note that we have the following equality:

$$\forall \hat{\mathbf{X}} \in \mathcal{M}, \eta \in \mathbb{R}^4, \mathbf{l} \in \mathbb{R}^3, \quad h_{\text{feat}}(\hat{\mathbf{X}}, \mathbf{l}) = h_{\text{feat}}(\mathcal{T}(\hat{\mathbf{X}}, 0, \eta), e^{\eta_1 \mathbf{g}}\mathbf{l} + \eta_{2:4}) \tag{7.107}$$

which merely translates frame invariance definition.

The fact that this equality is true for all vector η allows to differentiate both side of the equality with respect to η. Using the chain rule, one has:

$$\mathbf{F}\boldsymbol{N}_i + \mathbf{E}\partial_\eta(e^{\eta_1 \mathbf{g}}\mathbf{l} + \eta_{2:4}) = 0 \tag{7.108}$$
$$\mathbf{F}\boldsymbol{N}_i = \mathbf{E}\left[[\mathbf{l}]_\times \mathbf{g}, \mathbf{I}_3\right] \tag{7.109}$$

We have recalled at the beginning of the paragraph the expression of measurement matrix used for landmark measurement error. We can write further:

$$\mathbf{H}_{\text{feat}}\mathbf{N}_i = \boldsymbol{O}_0^\mathsf{T}\mathbf{F}\mathbf{N}_i = \boldsymbol{O}_0^\mathsf{T}\mathbf{E}\left[[\mathbf{l}]_\times \mathbf{g}, \mathbf{I}_3\right] \tag{7.110}$$

But, by definition of $\mathbf{O}_0$, ((7.25), Page 142). $\mathbf{O}_0^\mathsf{T}\mathbf{E} = 0$ and the condition (7.83) holds: $\mathbf{H}_{\text{feat}}\mathbf{N}_i = 0$.

Note: if we were to use an *inverse depth in first ray* parametrization of features, the condition is also true, and can be demonstrated similarly. $\forall \hat{\mathbf{X}}, \eta, \mathbf{l}, h(\hat{\mathbf{X}}, \mathbf{l}) = h(\mathcal{T}(\hat{\mathbf{X}}, 0, \eta), d)$ so that by differentiating with respect to η one directly has: $0 = \mathbf{F}\boldsymbol{N}_i$.

We thus have proven that the output of a EKF for the MVINS system is invariant under $\mathcal{T}(\cdot, 0, \eta)$.

Combined with the invariance to deterministic transform proved in (7.82) and using the equality noticed in (7.63), we prove that the EKF is invariant to any stochastic unobservable transform of the MVINS system.

7.3.3.2 How does this relate to OC-EKF technique?

The OC-EKF has been proposed for VINS in [Hesch et al., 2012] to enforce the condition (7.83) by artificially modifying the transition function and the measurement process functions. The conditions (7.83):

$$\forall n \text{ and } i \geq 0, \quad \mathbf{H}_{n+i+1}\Phi_{n+i}\Phi_{n+i-1}...\Phi_i\mathbf{N}_i = 0$$

is enforced the following way.

- At each new propagated state they explicitly compute $\mathbf{N}_{k+1}$ thanks to the definition:

$$\mathbf{N}_{k+1} = \frac{\partial}{\partial \eta}\left(\mathcal{T}(\hat{\mathbf{X}}_{k+1|k}, 0, \eta) \boxminus \mathcal{T}(\hat{\mathbf{X}}_{k+1|k}, 0, 0)\right)\Big|_{\eta=0} \tag{7.111}$$

- Before propagating covariance of the filter, OC-EKF slightly modifies the transition matrix $\Phi_{k+1,k}$ to $\Phi^*_{k+1,k}$ that respects the constraint:

$$\mathbf{N}_{k+1} = \Phi^*_{k+1,k}\mathbf{N}_k \tag{7.112}$$

  It finds the matrix $\Phi^*_{k+1,k}$ either by an analytical solution, either by solving the minimization problem:

$$\min_{\Phi^*_{k+1,k}} \left\|\Phi^*_{k+1,k} - \Phi_{k+1,k}\right\|_{\text{Frobenius}} \tag{7.113}$$

$$\text{s.t. } \mathbf{N}_{k+1} = \Phi^*_{k+1,k}\mathbf{N}_k \tag{7.114}$$

  and uses this matrix instead of the original one to propagate covariance.
- Similarly, the measurement matrix $\mathbf{H}_{k+1}$ is modified to satisfy:

$$\mathbf{N}_{k+1}\mathbf{H}_{k+1} = 0 \tag{7.115}$$

  by minimizing again the problem:

$$\min_{\mathbf{H}^*_{k+1}} \left\|\mathbf{H}^*_{k+1} - \mathbf{H}_{k+1}\right\|_{\text{Frobenius}} \tag{7.116}$$

$$\text{s.t. } \mathbf{H}^*_{k+1}\mathbf{N}_k = 0 \tag{7.117}$$

All in all, the result is that the OC-EKF methodology applied to Chapter 6 estimator also verifies Definition 3. However, it is not clear this artificial modification does not introduce unexpected problems, as it stems quite arbitrarily (in our opinion) from a Frobenius norm minimization.

7.3.4 Numerical Results

On the same data as previous experiments, the filter heading uncertainty and position output is given in Figure 7.3. This figure is to be analyzed comparatively to the previous Figures 7.1 and 7.2.

The translation of this difference of behavior on the trajectory can be assessed on Figure 7.4. This figure represents the result of three versions of the MI-MSCKF filter.

- **MI-MSCKF** the version of Chapter 6.
- **RI-MI-MSCKF** the version with the parametrization of Section 7.3.3.
- **MI-MSCKF-OC** which is the MI-MSCKF implemented with an observability-constrained strategy to enforce invariance to unobservable stochastic transforms.

We did the following experiments: we ran the three filters exactly in the same conditions and with the same measurements. They were initialized with the same mean and covariance estimate using the *true* heading and a large heading covariance. Thus, we would expect the three trajectories to superimpose on the satellite map without any registration step, as they were all initialized with the correct heading.

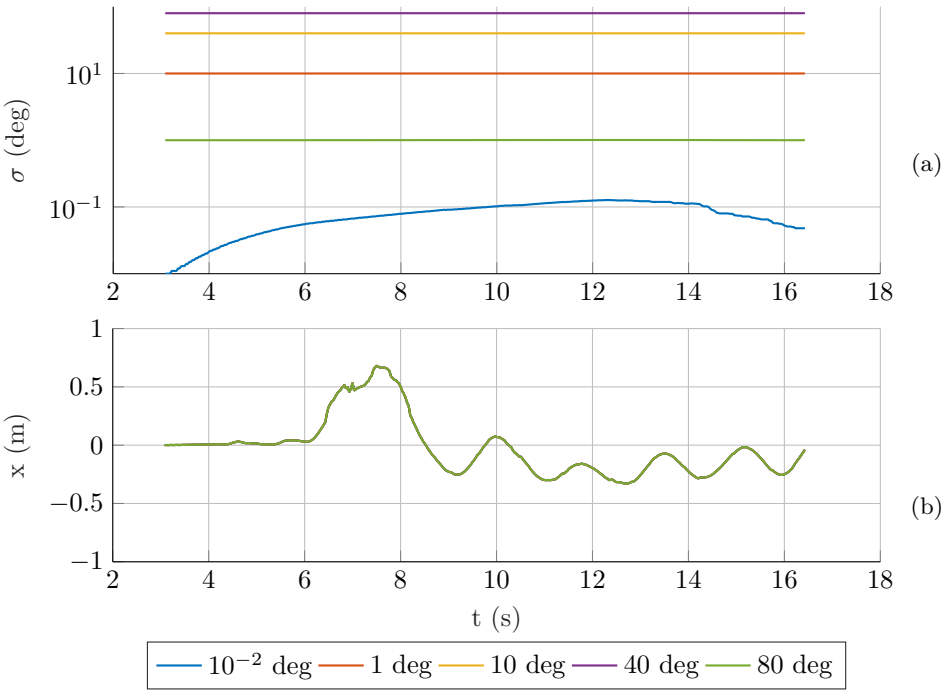


Figure 7.3: (**a**): heading uncertainty propagated by the invariant filter with different initial heading uncertainty. (**b**): the x position estimated by the filter with the same different initial heading uncertainty. The five curves on (**b**) figure cannot be distinguished. With the new parametrization, the initial heading uncertainty does not influence the position estimate of the filter, and the filter does not reduce its estimation of heading uncertainty as expected by analysis of unobservable modes of the MVINS model. Those graphics are to be compared to the one depicting the behavior of the parametrization of Chapter 6 in Figure 7.1, Page 143 and Figure 7.2, Page 146.

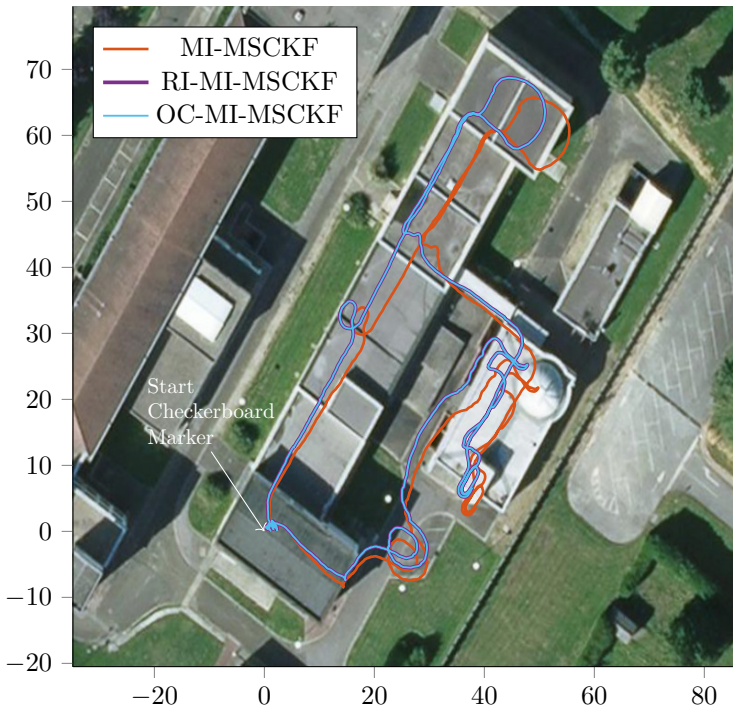


Figure 7.4: Results of trajectories of MI-MSCKF, RI-MI-MSCKF and MI-MSCKF-OC on Traj1 when initialized with the correct heading value and a large covariance on it. (The two last trajectories are nearly identical and can hardly be distinguished) Because of lack of invariance of the MI-MSCKF to stochastic unobservable transformation, the heading gets corrupted near the beginning of the trajectory, in the same time the heading covariance decreases. This translates to a rotated trajectory for this non invariant filter, while the two other filters are correctly aligned with the map.

This is indeed what is observed for RI-MI-MSCKF and MI-MSCKF-OC filters. But not for MI-MSCKF, whose trajectory output is rotated. For the two last filters, final drifts are nearly insensitive to initial covariance while for the first one, angular drift strongly depends on the initial heading covariance. The behavior of MI-MSCKF leads to a low translational drift but a strong heading drift compared to the first pose estimate. Table 7.1 shows that this behavior is quite general on the entire dataset with MI-MSCKF showing a very large angular drift compared to all other methods. In fact, this drift is mainly created during the first seconds of the filter run, so that, the trajectory shape does not show significant visible heading drift. These numerical results also clearly show that for all other methods the final drift is insensitive to the magnitude of initial heading covariance.

7.4 Conclusion of this Chapter

In an attempt to solve the discontinuity of the trajectory noted in the end of Chapter 5 and Chapter 6, we investigated Invariant (and observability-constrained) version of the MI-MSCKF algorithm. We proposed a new error parametrization of the MI-MSCKF and demonstrated that this parametrization comes with invariance to unobservable stochastic transform property. This property has been rigorously defined. It is naturally expected from an EKF when filtering with partially unobservable states.

We have shown on real data that the proposed parametrization and OC-EKF provided improvements in the behavior of the filter in a specific case, where the initial heading uncertainty is

	Traj1	Traj2	Traj3	Traj4	Traj5
MI-MSCKF	0.35	0.33	0.54	0.32	0.23
MI-MSCKF-LCOV	0.33	0.38	0.56	0.30	0.20
RI-MI-MSCKF	0.38	0.33	0.55	0.30	0.23
RI-MI-MSCKF-LCOV	0.38	0.34	0.55	0.30	0.23
OC-MI-MSCKF	0.38	0.33	0.55	0.30	0.23
OC-MI-MSCKF-LCOV	0.38	0.33	0.55	0.30	0.23

(a) Final translational drift (% of trajectory length)

	Traj1	Traj2	Traj3	Traj4	Traj5
MI-MSCKF	0.401	2.440	0.047	3.458	1.962
MI-MSCKF-LCOV	2.584	7.186	5.153	12.047	8.230
RI-MI-MSCKF	0.715	2.507	0.057	3.910	1.995
RI-MI-MSCKF-LCOV	0.715	2.507	0.057	3.910	1.995
OC-MI-MSCKF	0.717	2.485	0.010	3.854	1.991
OC-MI-MSCKF-LCOV	0.717	2.485	0.010	3.854	1.991

(b) Final angular drift (deg)

Table 7.1: Final translational and angular drift for various filters. The LCov suffix denotes try with a high covariance of the initial heading. For original MI-MSCKF filter, the angular drift strongly depends on the initial heading uncertainty, while it does not for the others.) Note that the numbers can not be compared to previous chapter as outlier rejection scheme has been slightly modified to ensure every filter uses exactly the same visual information as input (non-deterministic RANSAC step was bypassed).

substantial.

Even if it was not demonstrated here by lack of time, we guess that this improvement could be beneficial in the case of temporary observability of the heading. In our specific case, we could imagine observing the north direction only in the outdoor part of the trajectory for instance, which occurs several minutes after the initialization. This outdoor scenes could be detected thanks to the very low gradient measurement. In that specific case, it is likely that the standard MI-MSCKF would suffer from linearization error and inconsistencies because of the non-consistent decrease of the covariance observed at the beginning of the trajectory, while invariant parametrization and observability constrained version would react better to this new observation.

Conclusion

This work initially aimed to answer the question: "Can a MIMU improves visual-inertial navigation systems?". Noting that the two technologies had complementary failure modes and applicability domains, we focused on combining their respective strengths to improve the – now well-established – VINS-based position tracking. As we were willing to derive the fairest possible conclusion, we have put many efforts on building a state-of-the-art visual-inertial odometry system and attempted to introduce magneto-inertial dead-reckoning ideas into them. We described mathematically sound ways to tackle the fusion problem and we found that MIMU *can* indeed improves VIO estimate by increasing estimation robustness to unfavorable visual environments. We demonstrated that fact on a real dataset and non-caricatural scenarios. This is the primary result of the thesis.

Recall of our contributions

We started by a proof-of-concept using a depth sensor system combined with the MIMU hardware, where we showed that exploiting the stationary magnetic field disturbances leads to improved robustness in scenarios complicated for a simple incremental 3D-3D alignment algorithm or for situations where MI-DR techniques failed. Noticing exclusively passive sensors were actually used in state-of-the-art position tracking for AR, we investigated the passive vision sensor case subsequently. This configuration was more difficult to handle from an algorithm point of view as the 3D geometry should be inferred from regular 2D images and thus required a tight fusion of all of the sensors information for best performances. We presented two distinct ways to fuse magneto-inertial and visual information tightly; the first one was based on an optimization paradigm and the second one on filtering. Our primary contributions on this issue were: (i) a consistent derivation of a magneto-inertial residual that can be used in a bundle adjustment framework, foundation of several SLAM algorithms for position tracking; (ii) a thorough description of an application of the former to a sliding-window smoother; (iii) the evaluation of the dead-reckoning performance on a real dataset; (iv) the description of a filtering formulation of the sensor fusion problem with its evaluation on the same dataset; and finally (v) a study of an invariant version of the filter, and of related filter behavior improvement in specific situations difficult for the standard EKF methodology.

Improvements and perspectives

Regarding the implemented algorithm, some directions are still left to work on. First, one should work to identify what causes the discontinuity of the trajectory presented in Section 5.7 and Section 6.7 and answer why this effect does not appear on [Paul et al., 2017] for instance. Secondly, our MVINS dead-reckoning could also be integrated into a pose-graph SLAM or relocalization framework with multi-session features as described for instance in [McDonald et al., 2013; Qin et al., 2017; Schneider et al., 2017]. We believe that the framework of [Schneider et al., 2017] is a perfect candidate for this work and that our work could improve the ease of constructing the visual map in challenging conditions.

Also, our algorithm could be extended to deal with omnidirectional or multi-cameras to improve further the robustness of our method in specific scenarios as demonstrated in [Paul et al., 2017; Forster et al., 2017].

About the applications of invariance theory to our problem, we show results on real-data of specific situations in which the behavior of the invariant filter outperforms the non-invariant one. We noticed though that these improvements did not drastically change the global shape of the trajectories obtained. This might mean that the algorithmic source of inconsistencies was hidden behind more substantial errors from uncontrolled other sources that have yet to be found. It is not clear yet if this remark is general for VINS filters or if this is specific to our implementation. On this subject also, we note that more experiments could be made: the more consistent orientation

behavior of the filters could bring benefits in the case of intermittent measure of heading, but this has not been thoroughly tested yet.

On an applicative side, this work made a significant step in the use of a MIMU to improve VINS navigation for pedestrian dead-reckoning and exploration scenarios but some work is still needed in the evaluation of this solution for specific use-cases: other technical and scientific challenges would also have to be overcome depending on the application among which hardware integration of the MIMU sensors in AR headset, size reduction or MVINS in-the-field calibration algorithm. All these subjects lead directly or indirectly to exciting future topic of research. Our successful proof-of-concept should encourage looking closely at these.

In general, thanks to its improved availability, magneto-visual-inertial navigation systems could bring benefits in traditional application of VINS where high reliability and integrity is paramount, and, thanks to its reduced power consumption overhead compared to an IMU, it could bring benefits in every embedded application relying on battery as the source of power. We think that MIMU could help to reduce the power consumption of VINS by decreasing the computational load on the camera and image processing, for an equal tracking quality; proving strictly this claim would need more research and engineering.

At a time where some ask if the new event-camera hardware does solve wholly the VINS problem ([Vidal et al., 2017]), our work presented a new and promising orthogonal way to improve VINS that could potentially also be used in conjunction with future generations of optical sensors.

Appendix A

Computation of B_X and C_X of Chapter 5

This sections explicits the computation of matrices $\mathbf{B}_\mathbf{X}$ and $\mathbf{C}_\mathbf{X}$ used in Section 5.2.1.2 of the thesis.

We recall $\mathbf{\Sigma}_{\text{preint}}$ denoted the covariance matrix of the preintegrated measurements. It can be written:

$$\mathbf{\Sigma}_{\text{preint}} = \begin{bmatrix} \mathbf{\Sigma}_{\text{i}} & * \\ * & * \end{bmatrix} \tag{A.1}$$

We dropped here ij indices compared the Chapter 5 and reduced the notation $_{\text{imu}}$ to $_{\text{i}}$. We have said that, under this approximation, the probability density of the MIMU residual (conditioned on the state) is also a Gaussian, but whose covariance depends on the value of the state:

$$\mathbf{r}_{\text{mimu}} \propto \mathcal{N}(\mathbf{0}_{12\times 1}, \mathbf{A}_\mathbf{X} \mathbf{\Sigma}_{\text{preint}} \mathbf{A}_\mathbf{X}^\mathsf{T}) \tag{A.2}$$

$$\mathbf{A}_\mathbf{X} = \begin{bmatrix} \mathbf{I}_9 & \mathbf{0}_{9\times 5} & \mathbf{0}_{9\times 5} & \mathbf{0}_{9\times 3} \\ \mathbf{0}_{3\times 9} & \mathbf{v}_i \otimes \mathbf{I}_3 \mathcal{P}_{\nabla \mathbf{B}} & \mathbf{g}^\mathsf{T} \mathbb{R}_i^w \otimes \mathbf{I}_3 \mathcal{P}_{\nabla \mathbf{B}} & \mathbf{I}_3 \end{bmatrix} \tag{A.3}$$

We will write in this appendix the covariance of the MIMU error terms with the following decomposition:

$$\mathbf{\Sigma}_{\text{mimu}} = (\mathbf{A}_\mathbf{X} \mathbf{\Sigma}_{\text{preint}} \mathbf{A}_\mathbf{X}^\mathsf{T}) = \begin{bmatrix} \mathbf{\Sigma}_{\text{i}} & \mathbf{\Sigma}_{\text{m;i}}^\mathsf{T} \\ \mathbf{\Sigma}_{\text{m;i}} & \mathbf{\Sigma}_{\text{m}} \end{bmatrix} \tag{A.4}$$

The inverse of this covariance can be computed leveraging the Schur complement of the covariance matrix. This ones writes $(\mathbf{\Sigma}_{\text{m}} - \mathbf{\Sigma}_{\text{m;i}} \mathbf{\Sigma}_{\text{i}}^{-1} \mathbf{\Sigma}_{\text{m;i}}^\mathsf{T})$. One has:

$$\mathbf{\Sigma}_{\text{mimu}}^{-1} = \begin{bmatrix} \mathbf{I}_9 & -\mathbf{\Sigma}_{\text{i}}^{-1} \mathbf{\Sigma}_{\text{m}}^\mathsf{T} \\ 0 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{\text{i}}^{-1} & 0 \\ 0 & (\mathbf{\Sigma}_{\text{m}} - \mathbf{\Sigma}_{\text{m;i}} \mathbf{\Sigma}_{\text{i}}^{-1} \mathbf{\Sigma}_{\text{m;i}}^\mathsf{T})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_9 & -\mathbf{\Sigma}_{\text{m}} \mathbf{\Sigma}_{\text{i}}^{-1} \\ 0 & \mathbf{I}_3 \end{bmatrix} \tag{A.5}$$

Finally, the error term $\mathbf{r}_{\text{mimu}}^\mathsf{T} \mathbf{\Sigma}_{\text{mimu}} \mathbf{r}_{\text{mimu}}$ writes equivalently:

$$\underbrace{\mathbf{r}_{\text{imu}}^\mathsf{T} \mathbf{\Sigma}_{\text{i}}^{-1} \mathbf{r}_{\text{imu}}}_{\text{Pure inertial error term}} \tag{A.6}$$

$$+ \tag{A.7}$$

$$(\mathbf{r}_{\text{imu}} - \underbrace{\mathbf{\Sigma}_{\text{m;i}} \mathbf{\Sigma}_{\text{i}}^{-1}}_{\mathbf{B}_\mathbf{X}} \mathbf{r}_{\text{mag}})^\mathsf{T} \underbrace{(\mathbf{\Sigma}_{\text{m}} - \mathbf{\Sigma}_{\text{m;i}} \mathbf{\Sigma}_{\text{i}}^{-1} \mathbf{\Sigma}_{\text{m;i}}^\mathsf{T})^{-1}}_{\mathbf{C}_\mathbf{X}} (\mathbf{r}_{\text{imu}} - \underbrace{\mathbf{\Sigma}_{\text{m;i}} \mathbf{\Sigma}_{\text{i}}^{-1}}_{\mathbf{B}_\mathbf{X}} \mathbf{r}_{\text{mag}}) \tag{A.8}$$

Where we have made the matrices $\mathbf{B}_\mathbf{X}$ and $\mathbf{C}_\mathbf{X}$ appeared :

$$\mathbf{B}_\mathbf{X} = \mathbf{\Sigma}_{\text{m;i}} \mathbf{\Sigma}_{\text{i}}^{-1} \tag{A.9}$$

$$\mathbf{C}_\mathbf{X} = \mathbf{\Sigma}_{\text{m}} - \mathbf{\Sigma}_{\text{m;i}} \mathbf{\Sigma}_{\text{i}}^{-1} \mathbf{\Sigma}_{\text{m;i}}^\mathsf{T} \tag{A.10}$$

In previous expression $\mathbf{\Sigma}_{\text{m}}$and $\mathbf{\Sigma}_{\text{m;i}}$both depend on the state. We clarify what can be precomputed at preintegration time by introducing $\mathbf{S}_\mathbf{X} = \begin{bmatrix} \mathbf{v}_i \otimes \mathbf{I}_3 \mathcal{P}_{\nabla \mathbf{B}} & \mathbf{g}^\mathsf{T} \mathbb{R}_i^w \otimes \mathbf{I}_3 \mathcal{P}_{\nabla \mathbf{B}} & \mathbf{I}_3 \end{bmatrix}$ the non-trivial

bottom-right corner of $\mathbf{A_X}$:

$$\mathbf{B_X} = \mathbf{S_X} \underbrace{\left[\boldsymbol{\Sigma}_{\text{preint}}\right]_{10:23;10:23} \boldsymbol{\Sigma}_{\text{i}}^{-1}}_{\text{precomputed}} \tag{A.11}$$

$$\mathbf{C_X} = \mathbf{S_X} \underbrace{\left(\left[\boldsymbol{\Sigma}_{\text{preint}}\right]_{10:23;10:23} - \left[\boldsymbol{\Sigma}_{\text{preint}}\right]_{10:23;10:23} \boldsymbol{\Sigma}_{\text{i}}^{-1} \left[\boldsymbol{\Sigma}_{\text{preint}}\right]_{10:23;10:23}^{\mathsf{T}} \right)}_{\text{precomputed}} \mathbf{S}_{\mathbf{X}}^{\mathsf{T}} \tag{A.12}$$

Appendix B

Appendices of Chapter 7

B.1 Proof of Invariance to $\mathcal{T}(\cdot\,, \theta, 0)$

The aim is to prove that for the following unobservable stochastic transform and retraction operator:

$$\mathcal{T}(\mathbf{X}, \theta, \eta) \overset{\text{def}}{=} \begin{pmatrix} e^{(\eta_1+\theta_1)\mathbf{g}}\mathtt{R} \\ e^{(\eta_1+\theta_1)\mathbf{g}}\mathbf{v} \\ e^{(\eta_1+\theta_1)\mathbf{g}}\mathbf{p} + \theta_{2:4} + \eta_{2:4} \\ e^{(\eta_1+\theta_1)\mathbf{g}}\mathbf{B} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{pmatrix}, \qquad \mathbf{X}_k \boxplus \mathbf{e}_k = \begin{pmatrix} e^{\mathbf{e}_\mathtt{R}}\hat{\mathtt{R}}^w \\ \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{v} + e^{\mathbf{e}_\mathtt{R}}\hat{\mathbf{v}}^w \\ \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{p} + e^{\mathbf{e}_\mathtt{R}}\hat{\mathbf{p}} \\ \mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{B} + e^{\mathbf{e}_\mathtt{R}}\hat{\mathbf{B}}^w \\ \mathbf{e}_{\mathbf{b}_g} + \hat{\mathbf{b}_g} \\ \mathbf{e}_{\mathbf{b}_a} + \hat{\mathbf{b}_a} \end{pmatrix} \tag{B.1}$$

we have:

$$\forall \mathbf{e}, \forall \mathbf{X}, \quad \mathcal{T}(\mathbf{X} \boxplus \mathbf{e}, \theta, 0) = \mathcal{T}(\mathbf{X}, \theta, 0) \boxplus \mathbf{W}_\theta \mathbf{e} \tag{B.2}$$

with

$$\mathbf{W}_\theta = \begin{bmatrix} e^{\theta_1\mathbf{g}} & 0 & 0 & 0 & 0 \\ 0 & e^{\theta_1\mathbf{g}} & 0 & 0 & 0 \\ [\theta_{2:4}]_\times e^{\theta_1\mathbf{g}} & 0 & e^{\theta_1\mathbf{g}} & 0 & 0 \\ 0 & 0 & 0 & e^{\theta_1\mathbf{g}} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_6 \end{bmatrix}. \tag{B.3}$$

We will prove it by a direct computation. We will need the following identity though:

$$\forall \mathtt{R} \in \mathrm{SO}(3), \mathbf{e}_\mathtt{R} \in \mathfrak{so}(3), \quad \mathtt{R}\mathbf{J}_r(-\mathbf{e}_\mathtt{R}) = \mathbf{J}_r(-\mathtt{R}\mathbf{e}_\mathtt{R})\mathtt{R} \tag{B.4}$$

$$\forall \mathtt{R} \in \mathrm{SO}(3), a \in \mathfrak{so}(3)^3, \quad \mathbf{J}_r(a)\,[a]_\times = \mathbf{I}_3 - e^a \tag{B.5}$$

Both can be shown using $\mathbf{J}_r$ definition and Rodriguez formula for rotation:

$$\mathbf{J}_r(\theta) = \mathbf{I}_3 - \frac{1-\cos\|\theta\|}{\|\theta\|^2}\,[\theta]_\times + \frac{\|\theta\| - \sin\|\theta\|}{\|\theta\|^3}\,[\theta]_\times^2 \tag{B.6}$$

$$e^\theta = \mathbf{I}_3 + \frac{\sin\theta}{\|\theta\|}\,[\theta]_\times + \frac{(1-\cos(\theta))}{\|\theta\|^2}\,[\theta]_\times^2 \tag{B.7}$$

Computation of $\mathcal{T}(\mathbf{X} \boxplus \mathbf{e}, \theta, 0)$

$$\mathcal{T}(\mathbf{X} \boxplus \mathbf{e}, \theta, 0) = \begin{pmatrix} e^{\theta_1\mathbf{g}} e^{\mathbf{e}_\mathtt{R}}\mathtt{R}^w \\ e^{\theta_1\mathbf{g}}\left(\mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{v} + e^{\mathbf{e}_\mathtt{R}}\mathbf{v}^w\right) \\ e^{\theta_1\mathbf{g}}\left(\mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{p} + e^{\mathbf{e}_\mathtt{R}}\mathbf{p}\right) + \theta_{2:4} \\ e^{\theta_1\mathbf{g}}\left(\mathbf{J}_r(-\mathbf{e}_\mathtt{R})\mathbf{e}_\mathbf{B} + e^{\mathbf{e}_\mathtt{R}}\mathbf{B}^w\right) \\ \mathbf{e}_{\mathbf{b}_g} + \mathbf{b}_g \\ \mathbf{e}_{\mathbf{b}_a} + \mathbf{b}_a \end{pmatrix} \tag{B.8}$$

Computation of $\mathcal{T}(\mathbf{X}, \theta, 0) \boxplus \mathbf{W}_\theta \mathbf{e}$

$$\mathcal{T}(\mathbf{X}, \theta, 0) \boxplus \mathbf{W}_\theta \mathbf{e} = \begin{pmatrix} e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} e^{\theta_1 \mathbf{g}} \mathtt{R}^w \\ \left(\mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathbf{v} + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} e^{\theta_1 \mathbf{g}} \mathbf{v}^w \right) \\ \left(\mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) (e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathbf{p} + [\theta_{2:4}]_\times e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} (e^{\theta_1 \mathbf{g}} \mathbf{p} + \theta_{2:4}) \right) \\ \left(\mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathbf{B} + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} e^{\theta_1 \mathbf{g}} \mathbf{B}^w \right) \\ \mathbf{e}_{\mathbf{b}_g} + \mathbf{b}_g \\ \mathbf{e}_{\mathbf{b}_a} + \mathbf{b}_a \end{pmatrix} \tag{B.9}$$

Now we will do the difference between these element per element to proves the egality:

- Rotation:
  one has $e^{\theta_1 \mathbf{g}} e^{\mathbf{e}_\mathtt{R}} \mathtt{R}^w - e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} e^{\theta_1 \mathbf{g}} \mathtt{R}^w = 0$. By definition of the Adjoint on SO(3).
- Velocity:
  one has $e^{\theta_1 \mathbf{g}} \left(\mathbf{J}_r(-\mathbf{e}_\mathtt{R}) \mathbf{e}_\mathbf{v} + e^{\mathbf{e}_\mathtt{R}} \mathbf{v}^w \right) - \left(\mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathbf{v} + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} e^{\theta_1 \mathbf{g}} \mathbf{v}^w \right) = 0$ using (B.4)
- Position. See afterward.
- Magnetic field:
  one has $e^{\theta_1 \mathbf{g}} \left(\mathbf{J}_r(-\mathbf{e}_\mathtt{R}) \mathbf{e}_\mathbf{B} + e^{\mathbf{e}_\mathtt{R}} \mathbf{B}^w \right) - \left(\mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathbf{B} + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} e^{\theta_1 \mathbf{g}} \mathbf{B}^w \right) = 0$ using (B.4).
- Biases.
  This is trivial.

The equality for position is a bit more complicated. One has:

$$\begin{aligned} & e^{\theta_1 \mathbf{g}} \left(\mathbf{J}_r(-\mathbf{e}_\mathtt{R}) \mathbf{e}_\mathbf{p} + e^{\mathbf{e}_\mathtt{R}} \mathbf{p} \right) + \theta_{2:4} \\ & \qquad - \left(\mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) (e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathbf{p} + [\theta_{2:4}]_\times e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} (e^{\theta_1 \mathbf{g}} \mathbf{p} + \theta_{2:4}) \right) && \text{(B.10)} \\ & \qquad \text{(using (B.4) is not enough to find zero, it remains:)} && \text{(B.11)} \\ & \qquad = \theta_{2:4} - \mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) \, [\theta_{2:4}]_\times \, e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R} + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} \theta_{2:4} && \text{(B.12)} \\ & \qquad \text{(using } [a]_\times \, b = - \, [b]_\times \, a \text{:)} && \text{(B.13)} \\ & \qquad = \theta_{2:4} - \mathbf{J}_r(-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}) \left[-e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R} \right]_\times \theta_{2:4} + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} \theta_{2:4} && \text{(B.14)} \\ & \qquad \text{(using (B.5):)} && \text{(B.15)} \\ & \qquad = \theta_{2:4} - (\mathbf{I}_3 - e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}}) \theta_{2:4} + e^{e^{\theta_1 \mathbf{g}} \mathbf{e}_\mathtt{R}} \theta_{2:4} && \text{(B.16)} \\ & \qquad = 0 && \text{(B.17)} \end{aligned}$$

B.2 Expression of transition matrix Φ in invariant parametrization

In this appendix we write the full mimu state propagation with the parametrization of Chapter 7, Page 139.

State :

$$\mathbf{X}_k = (\mathtt{R}_k^w, \mathbf{v}_k^w, \mathbf{p}_k^w, \mathbf{B}_k^w, \mathbf{b}_{g_k}, \mathbf{b}_{a_k}) \tag{B.18}$$

Current State dynamic: The discrete state dynamic writes

$$\mathtt{R}^w_{k+1} = \mathtt{R}^w_k \widetilde{\Delta \mathtt{R}}_{kk+1}, \tag{B.19}$$

$$\mathbf{v}^w_{k+1} = \mathbf{v}^w_k + \mathbf{g}^w \Delta t_{k;k+1} + \mathtt{R}^w_k \widetilde{\Delta \mathbf{v}}_{kk+1}, \tag{B.20}$$

$$\mathbf{p}^w_{k+1} = \mathbf{p}^w_k + \mathtt{R}^w_k \mathbf{v}^b_k \Delta t_{k;k+1} + \frac{1}{2}\mathbf{g}^w \Delta t^2_{k;k+1} + \mathtt{R}^w_k \widetilde{\Delta \mathbf{p}}_{kk+1}, \tag{B.21}$$

$$\mathbf{B}^w_{k+1} = \widetilde{\Delta \mathtt{R}}^{\mathsf{T}}_{kk+1} \mathbf{B}^w_k + \mathtt{R}_k \widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{v};kk+1} \mathtt{R}_k \mathbf{v}^w_k + \mathtt{R}_k \widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{g};kk+1} \mathtt{R}^{w\mathsf{T}}_k \mathbf{g}^w + \mathtt{R}_k \widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{a};kk+1}. \tag{B.22}$$

$$\mathbf{b}_{g\,k+1} = e^{\frac{1}{\tau_{bg}} \Delta t_{k;k+1}} \mathbf{b}_{g\,k} + \boldsymbol{\eta}_{bg} \tag{B.23}$$

$$\mathbf{b}_{a\,k+1} = e^{\frac{1}{\tau_{ba}} \Delta t_{k;k+1}} \mathbf{b}_{a\,k} + \boldsymbol{\eta}_{ba} \tag{B.24}$$

We recall the definition of the transition matrix:

$$\boldsymbol{\Phi}_{k+1} = \frac{\partial}{\partial \mathbf{e}} \left(f(\hat{\mathbf{X}}_k \boxplus \mathbf{e}, \mathbf{u}_k, \eta) \boxminus f(\hat{\mathbf{X}}_k, \tilde{\mathbf{u}}_k, 0) \right) \Big|_{\mathbf{e}=0, \mathbf{u}_k = \tilde{\mathbf{u}}_k, \eta = 0} \tag{B.25}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k} = \left[\begin{array}{cccc|cc} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{R}\mathbf{b}_g}} & \mathbf{0}_3 \\ \Delta t_i \left[\mathbf{g}\right]_\times & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_g}} & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_a}} \\ \frac{\Delta t_i^2}{2} \left[\mathbf{g}\right]_\times & \Delta t \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_g}} & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_g}} \\ \mathtt{R}_k \widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{g};i} \mathtt{R}^T_k \left[\mathbf{g}\right]_\times & \mathbf{0}_3 & \mathtt{R}_k \widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{g};i} \mathtt{R}^T_k & \mathbf{I}_3 & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_g}} & \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_a}} \\ \hline \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\exp\left(\frac{\Delta t}{\tau_g}\right) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\exp\left(\frac{\Delta t}{\tau_a}\right) \end{array}\right] \tag{B.26}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{R}\mathbf{b}_g}} = -\mathtt{R}_k \partial_{\mathbf{b}_g} (\widetilde{\Delta \mathtt{R}}_{k;k+1}), \tag{B.27}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_g}} = -\Delta t \left[\mathbf{v}_{k+1}\right]_\times \mathtt{R}_k - \partial_{\mathbf{b}_g} \left(\widetilde{\Delta \mathbf{v}}_{kk+1} \right) \tag{B.28}$$

$$\begin{aligned} \boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_g}} &= -\Delta t \left[\mathbf{B}_{k+1}\right]_\times \mathtt{R}_k \\ &\quad - \left(\mathbf{v}^{\mathsf{T}}_k \mathtt{R} \otimes \mathtt{R}^{\mathsf{T}}\right) \partial_{\mathbf{b}_g} \left(\text{Vec} \left(\widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{v};kk+1} \right) \right) \\ &\quad - \left(\mathbf{g}^{\mathsf{T}} \mathtt{R}_k \otimes \mathtt{R}^{\mathsf{T}}\right) \partial_{\mathbf{b}_g} \left(\text{Vec} \left(\widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{g};kk+1} \right) \right) \end{aligned} \tag{B.29}$$

$$- \partial_{\mathbf{b}_{g\,k}} \widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{a};kk+1} \tag{B.30}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_g}} = -\Delta t \left[\mathbf{p}_k\right]_\times \mathtt{R}_k - \partial_{\mathbf{b}_{g\,k}} \left(\widetilde{\Delta \mathbf{p}}_{kk+1} \right) \tag{B.31}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{v}\mathbf{b}_a}} = -\mathtt{R}_k \partial_{\mathbf{b}_{a\,k}} \left(\widetilde{\Delta \mathbf{v}}_{kk+1} \right) \tag{B.32}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{p}\mathbf{b}_a}} = -\mathtt{R}_k \partial_{\mathbf{b}_{a\,k}} \left(\widetilde{\Delta \mathbf{p}}_{kk+1} \right) \tag{B.33}$$

$$\boldsymbol{\Phi}^{\text{mimu}}_{k+1,k_{\mathbf{B}\mathbf{b}_a}} = -\mathtt{R}_k \partial_{\mathbf{b}_{a\,k}} \left(\widetilde{\boldsymbol{\Delta} \mathbf{B}}_{\mathbf{a};kk+1} \right) \tag{B.34}$$

Appendix C

Effect of Slightly Bad Sensor Synchronization on the Estimate of MVINS

The effect of a slight error of datation of sensors can be dramatic in a tight fusion scheme. We relate in this appendix, a detrimental effect we observed that was arising due to the changes of camera exposure time.

On our indoor/outdoor dataset, the change of exposure time varies from 0 (outdoor) to 15 ms (indoor). This can be seen on the following figures:

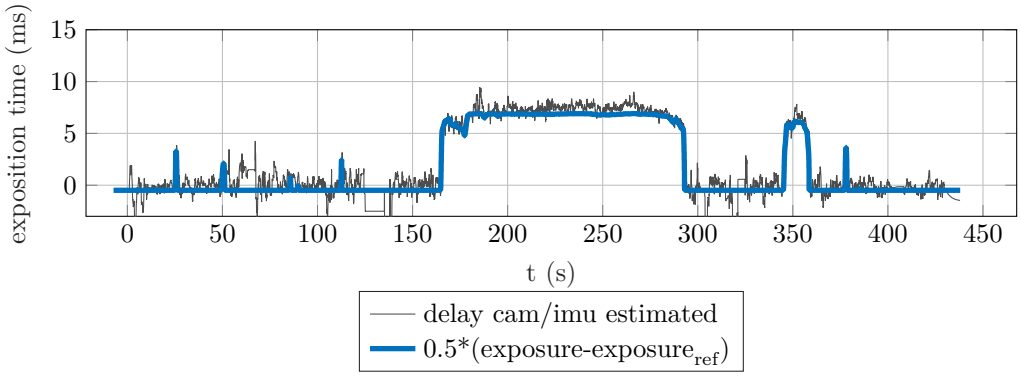


Figure C.1: Exposure of the camera across time. In light gray, the delay between camera and MIMU timestamp estimated by the filter of [Paul et al., 2017]

This induces two problems. First the visual observations are not temporally well-defined. Because the images are actually integrated over the exposure time. This is a problem because VINS algorithms generally assume an observation corresponds actually to an instant in time. Secondly, the best unique timestamp we can associate to an image is the middle of the exposition time. However, this is not what camera sensors generally send as timestamp. The sensor used in these experiments was actually giving the end of exposition.

Since we had access to the exposition time for each frame, we were able to correct the timestamp given by the camera of each image in order it to represent the middle of the exposition of the image. The following figure shows the effect of this correction on the results of the filter of Chapter 6 on TRAJ1 of our dataset.

Even if this correction magnitude was at max of 7 ms, taking it into account the exposure time in timestamp correction improves drastically the final drift.

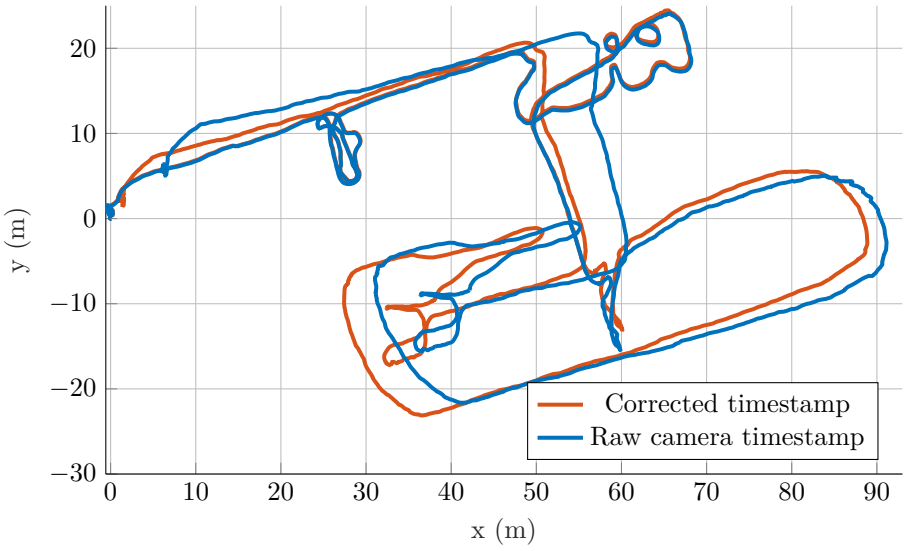


Figure C.2: Effect of synchronization error on the Traj1 trajectory. Poses are from mvins filter of Chapter 6.

Bibliography

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Prentice-Hall.

Antigny, N., Servières, M., and Renaudin, V. (2017). Pedestrian Track Estimation with Handheld Monocular Camera and Inertial-Magnetic Sensor for Urban Augmented Reality. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, Japan.

Barrau, A. (2015). *Filtres de Kalman Étendus Reposant Sur Une Variable d'erreur Non Linéaire Avec Applications à La Navigation*. PhD thesis. 2015ENMP0080.

Barrau, A. and Bonnabel, S. (2015). An EKF-SLAM algorithm with consistency properties. *arXiv:1510.06263 [cs]*.

Batista, P., Petit, N., Silvestre, C., and Oliveira, P. (2013). Further results on the observability in magneto-inertial navigation. In *American Control Conference (ACC), 2013*, pages 2503–2508. IEEE.

Bazin, J., Demonceaux, C., Vasseur, P., and Kweon, I. (2010). Motion estimation by decoupling rotation and translation in catadioptric vision. *Computer Vision and Image Understanding*, 114(2):254–273.

Bloesch, M., Burri, M., Omari, S., Hutter, M., and Siegwart, R. (2017). Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072.

Bloesch, M., Omari, S., Hutter, M., and Siegwart, R. (2015). Robust visual inertial odometry using a direct EKF-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference On*, pages 298–304. IEEE.

Boikos, K. and Bouganis, C. S. (2017). A high-performance system-on-chip architecture for direct tracking for SLAM. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–7.

Bonnabel, S., Barczyk, M., and Goulette, F. (2014a). On the Covariance of ICP-based Scan-matching Techniques. *arXiv:1410.7632 [cs]*.

Bonnabel, S., Barczyk, M., and Goulette, F. (2014b). On the covariance of scan-matching techniques for localization. *arXiv preprint arXiv:1410.7632*.

Bonnabel, S. and Barrau, A. (2017). The Invariant Extended Kalman Filter as a Stable Observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812.

Bonnabel, S., Martin, P., and Rouchon, P. (2008). Non-linear observer on Lie Groups for left-invariant dynamics with right-equivariant output. In *2008 IFAC World Congress*, pages 8594–8598, Seoul, South Korea.

Bouguet, J.-y. (2000). Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*.

Brossard, M., Bonnabel, S., and Barrau, A. (2017). Unscented Kalman Filtering on Lie Groups for Fusion of IMU and Monocular Vision.

Brunetto, N., Salti, S., Fioraio, N., Cavallari, T., and Stefano, L. (2015). Fusion of Inertial and Visual Measurements for RGB-D SLAM on Mobile Devices. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–9.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., and Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163.

Burri, M., Oleynikova, H., Achtelik, M. W., and Siegwart, R. (2015). Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference On*, pages 1872–1878. IEEE.

Bürki, M., Gilitschenski, I., Stumm, E., Siegwart, R., and Nieto, J. (2016). Appearance-based landmark selection for efficient long-term visual localization. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference On*, pages 4137–4143. IEEE.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332.

Carlone, L., Kira, Z., Beall, C., Indelman, V., and Dellaert, F. (2014). Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In *Robotics and Automation (ICRA), 2014 IEEE International Conference On*, pages 4290–4297. IEEE.

Caruso, D., Engel, J., and Cremers, D. (2015). Large-Scale Direct SLAM for Omnidirectional Cameras. In *International Conference on Intelligent Robots and Systems (IROS)*.

Caruso, D., Eudes, A., Sanfourche, M., Vissiere, D., and Le Besnerais, G. (2017a). An Inverse Square-root Filter for Robust Indoor/Outdoor Magneto-visual-inertial Odometry. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo.

Caruso, D., Eudes, A., Sanfourche, M., Vissiere, D., and Le Besnerais, G. (2017b). Robust Indoor/Outdoor Navigation through Magneto-visual-inertial Optimization-based Estimation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver.

Caruso, D., Eudes, A., Sanfourche, M., Vissière, D., and Le Besnerais, G. (2017c). A Robust Indoor/Outdoor Navigation Filter Fusing Data from Vision and Magneto-Inertial Measurement Unit. *Sensors*, 17(12):2795.

Caruso, D., Sanfourche, M., Le Besnerais, G., and Vissiere, D. (2016). Infrastructureless Indoor Navigation With an Hybrid Magneto-inertial and Depth Sensor System. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Alcalà de Henares.

Castellanos, J., Martinez-Cantin, R., Tardós, J., and Neira, J. (2007). Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1):21–29.

Castellanos, J. A., Neira, J., and Tardós, J. D. (2004). Limits to the consistency of EKF-based SLAM. *IFAC Proceedings Volumes*, 37(8):716–721.

Chesneau, C.-I., Hillion, M., Hullo, J.-F., Thibault, G., and Prieur, C. (2017). Improving magneto-inertial attitude and position estimation by means of magnetic heading observer. In *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference On*, Sapporo, Japan.

Chesneau, C.-I., Hillion, M., and Prieur, C. (2016). Motion estimation of a rigid body with an EKF using magneto-inertial measurements. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference On*, pages 1–6. IEEE.

Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M. (2010). 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631.

Clark, R., Wang, S., Wen, H., Markham, A., and Trigoni, N. (2017). VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. *arXiv:1701.08376 [cs]*.

Czarnowski, J., Leutenegger, S., and Davison, A. J. (2017). Semantic Texture for Robust Dense Tracking. *arXiv preprint arXiv:1708.08844*.

Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067.

Dellaert, F. and Kaess, M. (2006). Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203.

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2017). SuperPoint: Self-Supervised Interest Point Detection and Description. *arXiv:1712.07629 [cs]*.

Diel, D. D., DeBitetto, P., and Teller, S. (2005). Epipolar Constraints for Vision-Aided Inertial Navigation. In *Seventh IEEE Workshops on Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1*, volume 2, pages 221–228.

Dong-Si, T.-C. and Mourikis, A. (2012). Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1064–1071.

Dorveaux, E. (2011). *Magneto-Inertial Navigation: Principles and Application to an Indoor Pedometer*. PhD thesis, École Nationale Supérieure des Mines de Paris.

Dorveaux, E., Boudot, T., Hillion, M., and Petit, N. (2011). Combining inertial measurements and distributed magnetometry for motion estimation. In *American Control Conference (ACC), 2011*, pages 4249–4256.

Dorveaux, E., Vissiere, D., Martin, A.-P., and Petit, N. (2009). Iterative calibration method for inertial and magnetic sensors. In *Decision and Control, 2009 Held Jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference On*, pages 8296–8303. IEEE.

Dorveaux, E., Vissiere, D., and Petit, N. (2010). On-the-field calibration of an array of sensors. In *American Control Conference (ACC), 2010*, pages 6795–6802. IEEE.

dos Santos Fernandes, C., Rangel do Nascimento, E., and Montenegro Campos, M. F. (2013). Visual and Inertial Data Fusion for Globally Consistent Point Cloud Registration. In *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference On*, pages 210–217. IEEE.

DuToit, R. C., Hesch, J. A., Nerurkar, E. D., and Roumeliotis, S. I. (2016). Consistent Map-based 3D Localization on Mobile Devices. *arXiv:1604.08087 [cs]*.

DuToit, R. C., Hesch, J. A., Nerurkar, E. D., and Roumeliotis, S. I. (2017). Consistent map-based 3D localization on mobile devices. In *Robotics and Automation (ICRA), 2017 IEEE International Conference On*, pages 6253–6260. IEEE.

Eckenhoff, K., Geneva, P., and Huang, G. (2016). High-Accuracy Preintegration for Visual-Inertial Navigation. Technical report, Technical Report RPNG-2016-001, University of Delaware.

El-Sheimy, N., Hou, H., and Niu, X. (2008). Analysis and Modeling of Inertial Sensors Using Allan Variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149.

Engel, J., Koltun, V., and Cremers, D. (2016). Direct Sparse Odometry. In *arXiv:1607.02565*.

Engel, J., Koltun, V., and Cremers, D. (2018). Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625.

Engel, J., Schöps, T., and Cremers, D. (2014a). LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision (ECCV)*.

Engel, J., Sturm, J., and Cremers, D. (2013). Semi-Dense Visual Odometry for a Monocular Camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1456. Citeseer.

Engel, J., Sturm, J., and Cremers, D. (2014b). Scale-Aware Navigation of a Low-Cost Quadrocopter with a Monocular Camera. *Robotics and Autonomous Systems (RAS)*.

Fehr, M., Dymczyk, M., Lynen, S., and Siegwart, R. (2016). Reshaping our model of the world over time. In *Robotics and Automation (ICRA), 2016 IEEE International Conference On*, pages 2449–2455. IEEE.

Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395.

Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2015). On-Manifold Preintegration Theory for Fast and Accurate Visual-Inertial Navigation. *arXiv preprint arXiv:1512.02363*.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). SVO: Fast Semi-Direct Monocular Visual Odometry. In *Proc. IEEE Intl. Conf. on Robotics and Automation*.

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., and Scaramuzza, D. (2017). SVO: Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems. *IEEE Transactions on Robotics*, 33(2).

Furgale, P., Rehder, J., and Siegwart, R. (2013). Unified temporal and spatial calibration for multi-sensor systems. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 1280–1286. IEEE.

Gebre-Egziabher, D., Elkaim, G. H., Powell, J. D., and Parkinson, B. W. (2001). A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers. In *8th International St. Petersburg Conference on Navigation Systems (IEEE/AIAA)*.

Grabe, V., Bulthoff, H. H., and Giordano, P. R. (2013). A comparison of scale estimation schemes for a quadrotor UAV based on optical flow and IMU measurements. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 5193–5200. IEEE.

Guo, C. and Roumeliotis, S. I. (2013). IMU-RGBD camera navigation using point and plane features. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 3164–3171. IEEE.

Gutierrez-Gomez, D., Mayol-Cuevas, W., and Guerrero, J. J. (2016). Dense RGB-D visual odometry using inverse depth. *Robotics and Autonomous Systems*, 75:571–583.

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2014). RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Experimental Robotics*, pages 477–491. Springer.

Hernandez, J., Tsotsos, K., and Soatto, S. (2015). Observability, identifiability and sensitivity of vision-aided inertial navigation. In *Robotics and Automation (ICRA), 2015 IEEE International Conference On*, pages 2319–2325. IEEE.

Hertzberg, C., Wagner, R., Frese, U., and Schröder, L. (2013). Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77.

Hesch, J. A., Kottas, D. G., Bowman, S. L., and Roumeliotis, S. I. (2012). Observability-constrained vision-aided inertial navigation. *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep*, 1.

Hesch, J. A., Kottas, D. G., Bowman, S. L., and Roumeliotis, S. I. (2013). Towards consistent vision-aided inertial navigation. In *Algorithmic Foundations of Robotics X*, pages 559–574. Springer.

Hesch, J. A., Kottas, D. G., Bowman, S. L., and Roumeliotis, S. I. (2014). Camera-IMU-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201.

Hong, I., Kim, G., Kim, Y., Kim, D., Nam, B. G., and Yoo, H. J. (2014). A 27mW reconfigurable marker-less logarithmic camera pose estimation engine for mobile augmented reality processor. In *2014 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pages 209–212.

Huang, G., Anastasios I. Mourikis, and Stergios I. Roumeliotis (2010). Observability-based Rules for Designing Consistent EKF SLAM Estimators. *The International Journal of Robotics Research*, 29(5):502–528.

Huang, G., Kaess, M., and Leonard, J. J. (2014). Towards consistent visual-inertial navigation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4926–4933. IEEE.

Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I. (2008). Analysis and improvement of the consistency of extended Kalman filter based SLAM. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference On*, pages 473–479. IEEE.

Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I. (2009). A First-Estimates Jacobian EKF for Improving SLAM Consistency. In *Experimental Robotics*, Springer Tracts in Advanced Robotics, pages 373–382. Springer, Berlin, Heidelberg.

Hullo, J.-F. (2013). *Consolidation de Relevés Laser d'intérieurs Construits : Pour Une Approche Probabiliste Initialisée Par Géolocalisation*. Strasbourg.

IEEE (1998). *IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros*. publisher not identified, Place of publication not identified. OCLC: 812595703.

Irani, M. and Anandan, P. (2000). About Direct Methods. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 267–277, London, UK, UK. Springer-Verlag.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and others (2011). KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 559–568. ACM.

Jaimez, M. and González-Jiménez, J. (2015). Fast Visual Odometry for 3-D Range Sensors. *IEEE Transactions on Robotics*, 31(4):809–822.

Jones, E. S. and Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430.

Kaess, M., Williams, S., Indelman, V., Roberts, R., Leonard, J. J., and Dellaert, F. (2012). Concurrent filtering and smoothing. In *Information Fusion (FUSION), 2012 15th International Conference On*, pages 1300–1307. IEEE.

Kahler, O., Prisacariu, V. A., Ren, C. Y., Sun, X., Torr, P. H. S., and Murray, D. W. (2015). Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Transactions on Visualization and Computer Graphics*, 22(11).

Kaiser, J., Martinelli, A., Fontana, F., and Scaramuzza, D. (2017). Simultaneous State Initialization and Gyroscope Bias Calibration in Visual Inertial Aided Navigation. *IEEE Robotics and Automation Letters*, 2(1):18–25.

Keivan, N., Patron-Perez, A., and Sibley, G. (2016). Asynchronous adaptive conditioning for visual-inertial slam. In *Experimental Robotics*, pages 309–321. Springer.

Kerl, C., Stueckler, J., and Cremers, D. (2015). Dense Continuous-Time Tracking and Mapping with Rolling Shutter RGB-D Cameras. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile.

Kerl, C., Sturm, J., and Cremers, D. (2013a). Dense visual SLAM for RGB-D cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 2100–2106. IEEE.

Kerl, C., Sturm, J., and Cremers, D. (2013b). Robust odometry estimation for rgb-d cameras. In *Robotics and Automation (ICRA), 2013 IEEE International Conference On*, pages 3748–3754. IEEE.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium On*, pages 225–234. IEEE.

Kleinert, M. and Schleith, S. (2010). Inertial aided monocular SLAM for GPS-denied navigation. In *2010 IEEE Conference on Multisensor Fusion and Integration*, pages 20–25.

Konolige, K., Agrawal, M., and Sola, J. (2010). Large-scale visual odometry for rough terrain. In *Robotics Research*, pages 201–212. Springer.

Larnaout, D., Bourgeois, S., gay-bellile, V., and Dhome, M. (2012). Towards Bundle Adjustment with GIS Constraints for Online Geo-Localization of a Vehicle In Urban Center. pages 348–355.

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference On*, pages 2548–2555. IEEE.

Leutenegger, S., Furgale, P. T., Rabaud, V., Chli, M., Konolige, K., and Siegwart, R. (2013). Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization. In *Robotics: Science and Systems*.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334.

Li and Mourikis, A. I. (2012). Consistency of EKF-Based Visual-Inertial Odometry. Technical report.

Li, M. and Mourikis, A. I. (2013). High-precision, consistent EKF-based visual–inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711.

Li, M., Yu, H., Zheng, X., and Mourikis, A. I. (2014). High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference On*, pages 409–416. IEEE.

Li, R., Wang, S., Long, Z., and Gu, D. (2017). UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning. *arXiv preprint arXiv:1709.06841*.

Lothe, P., Bourgeois, S., Royer, E., Dhome, M., and Naudet-Collette, S. (2010). Real-time vehicle global localisation with a single camera in dense urban areas: Exploitation of coarse 3d city models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference On*, pages 863–870. IEEE.

Lucas, B. D. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lupton, T. and Sukkarieh, S. (2012). Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions. *IEEE Transactions on Robotics*, 28(1):61–76.

Lynen, S., Achtelik, M. W., Weiss, S., Chli, M., and Siegwart, R. (2013a). A robust and modular multi-sensor fusion approach applied to mav navigation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 3923–3929. IEEE.

Lynen, S., Omari, S., Wüest, M., Achtelik, M., and Siegwart, R. (2013b). Tightly Coupled Visual-Inertial Navigation System Using Optical Flow. *IFAC Proceedings Volumes*, 46(30):251–256.

Lynen, S., Sattler, T., Bosse, M., Hesch, J., Pollefeys, M., and Siegwart, R. (2015). Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In *Robotics: Science and Systems*.

Majdik, A. L., Till, C., and Scaramuzza, D. (2017). The Zurich urban micro aerial vehicle dataset. *The International Journal of Robotics Research*, 36(3):269–273.

Martin, P. and Salaun, E. (2007). Invariant observers for attitude and heading estimation from low-cost inertial and magnetic sensors. pages 1039–1045. IEEE.

Martinelli, A. (2013a). Closed-form solution of visual-inertial structure from motion. *International Journal of Computer Vision*, page online.

Martinelli, A. (2013b). Visual-inertial structure from motion: Observability and resolvability. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4235–4242.

Maye, J., Furgale, P., and Siegwart, R. (2013). Self-supervised calibration for robotic systems. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 473–480.

McDonald, J., Kaess, M., Cadena, C., Neira, J., and Leonard, J. J. (2013). Real-time 6-DOF multi-session visual SLAM over large-scale environments. *Robotics and Autonomous Systems*, 61(10):1144–1158.

Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235.

Middelberg, S., Sattler, T., Untzelmann, O., and Kobbelt, L. (2014). Scalable 6-dof localization on mobile devices. In *European Conference on Computer Vision*, pages 268–283. Springer.

Miller, M., Chung, S.-J., and Hutchinson, S. (2018). The Visual–Inertial Canoe Dataset. *The International Journal of Robotics Research*, 37(1):13–20.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2006). Real Time Localization and 3D Reconstruction. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 363–370.

Mourikis, A. I. and Roumeliotis, S. I. (2007). A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572.

Mourikis, A. I. and Roumeliotis, S. I. (2008). A dual-layer estimator architecture for long-term localization. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference On*, pages 1–8. IEEE.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *arXiv:1502.00956 [cs]*.

Mur-Artal, R. and Tardos, J. D. (2017). Visual-Inertial Monocular SLAM With Map Reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803.

Nerurkar, E. D., Wu, K. J., and Roumeliotis, S. I. (2014). C-KLAM: Constrained keyframe-based localization and mapping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference On*, pages 3638–3643. IEEE.

Neunert, M., Bloesch, M., and Buchli, J. (2016). An open source, fiducial based, visual-inertial motion capture system. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1523–1530.

Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference On*, pages 2320–2327. IEEE.

Nguyen, C. V., Izadi, S., and Lovell, D. (2012). Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking. pages 524–530. IEEE.

Niessner, M., Dai, A., and Fisher, M. (2014). Combining Inertial Navigation and ICP for Real-time 3D Surface Reconstruction. In *Eurographics (Short Papers)*, pages 13–16. Citeseer.

Paul, M. K., Wu, K., Hesch, J. A., Nerurkar, E. D., and Roumeliotis, S. I. (2017). A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS. In *Robotics and Automation (ICRA), 2017 IEEE International Conference On*, pages 165–172. IEEE.

Pfrommer, B., Sanket, N., Daniilidis, K., and Cleveland, J. (2017). PennCOSYVIO: A challenging Visual Inertial Odometry benchmark. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3847–3854.

Platinsky, L., Davison, A., and Leutenegger, S. (2017). Monocular visual odometry: Sparse joint optimisation or dense alternation? pages 5126–5133.

Qayyum, U., Kim, J., and others (2013). Inertial-kinect fusion for outdoor 3d navigation. In *Australasian Conference on Robotics and Automation (ACRA)*.

Qin, T., Li, P., and Shen, S. (2017). VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *arXiv:1708.03852 [cs]*.

Rambach, J. R., Tewari, A., Pagani, A., and Stricker, D. (2016). Learning to Fuse: A Deep Learning Approach to Visual-Inertial Camera Pose Estimation. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 71–76.

Rehder, J., Nikolic, J., Schneider, T., Hinzmann, T., and Siegwart, R. (2016). Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. In *Robotics and Automation (ICRA), 2016 IEEE International Conference On*, pages 4304–4311. IEEE.

Renaudin, V., Afzal, M. H., and Lachapelle, G. (2010). Complete Triaxis Magnetometer Calibration in the Magnetic Domain. https://www.hindawi.com/journals/js/2010/967245/.

Robert, E. and Perrot, T. (2017). Invariant filtering versus other robust filtering methods applied to integrated navigation. In *Integrated Navigation Systems (ICINS), 2017 24th Saint Petersburg International Conference On*, pages 1–7. IEEE.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference On*, pages 2564–2571. IEEE.

Sagnac, G. (1914). Effet tourbillonnaire optique. La circulation de l'éther lumineux dans un interférographe tournant. *J. Phys. Theor. Appl.*, 4(1):177–195.

Sanfourche, M., Vittori, V., and Le Besnerais, G. (2013). eVO: A realtime embedded stereo odometry for MAV applications. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 2107–2114. IEEE.

Santoso, F., Garratt, M. A., and Anavatti, S. G. (2017). Visual-Inertial Navigation Systems for Aerial Robotics: Sensor Fusion and Technology. *IEEE Transactions on Automation Science and Engineering*, 14(1):260–275.

Savage, P. G. (2000). *Strapdown Analytics*. Strapdown Associates.

Savage, P. G. (2015). Computational Elements For Strapdown Systems, WBN-14010.

Scaramuzza, D., Fraundorfer, F., Pollefeys, M., and Siegwart, R. (2009). Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1413–1419.

Schneider, T., Dymczyk, M., Fehr, M., Egger, K., Lynen, S., Gilitschenski, I., and Siegwart, R. (2017). Maplab: An Open Framework for Research in Visual-inertial Mapping and Localization. *arXiv:1711.10250 [cs]*.

Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J., and Cremers, D. (2018). The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. *arXiv:1804.06120 [cs]*.

Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V. (2013). Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor. In *Robotics: Science and Systems*, volume 1. Citeseer.

Sibley, G., Matthies, L., and Sukhatme, G. (2010). Sliding window filter with application to planetary landing. *Journal of Field Robotics*, 27(5):587–608.

Simon, D. (2006). *Optimal State Estimation: Kalman, H [Infinity] and Nonlinear Approaches*. Wiley-Interscience, Hoboken, N.J. OCLC: ocm64084871.

Simon, D. (2010). Kalman filtering with state constraints: A survey of linear and nonlinear algorithms. *IET Control Theory Applications*, 4(8):1303–1318.

Sola, J. (2010). Consistency of the monocular ekf-slam algorithm for three different landmark parametrizations. In *Robotics and Automation (ICRA), 2010 IEEE International Conference On*, pages 3513–3518. IEEE.

Strasdat, H., Davison, A. J., Montiel, J. M. M., and Konolige, K. (2011). Double window optimisation for constant time visual SLAM. In *Computer Vision (ICCV), 2011 IEEE International Conference On*, pages 2352–2359. IEEE.

Strasdat, H., Montiel, J. M., and Davison, A. J. (2012). Visual SLAM: Why filter? *Image and Vision Computing*, 30(2):65–77.

Tardif, J.-P., George, M., Laverne, M., Kelly, A., and Stentz, A. (2010). A new approach to vision-aided inertial navigation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference On*, pages 4161–4168. IEEE.

Teichman, A., Miller, S., and Thrun, S. (2013). Unsupervised Intrinsic Calibration of Depth Sensors via SLAM. volume 09.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372. Springer.

Troiani, C., Martinelli, A., Laugier, C., and Scaramuzza, D. (2014). 2-point-based outlier rejection for camera-imu systems with applications to micro aerial vehicles. In *Robotics and Automation (ICRA), 2014 IEEE International Conference On*, pages 5530–5536. IEEE.

Tsotsos, K., Chiuso, A., and Soatto, S. (2015). Robust inference for visual-inertial sensor fusion. In *Robotics and Automation (ICRA), 2015 IEEE International Conference On*, pages 5203–5210. IEEE.

Tsotsos, K., Pretto, A., and Soatto, S. (2012). Visual-inertial ego-motion estimation for humanoid platforms. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference On*, pages 704–711. IEEE.

Um, T. T., Babakeshizadeh, V., and Kulic, D. (2016). Exercise Motion Classification from Large-Scale Wearable Sensor Data Using Convolutional Neural Networks. *arXiv preprint arXiv:1610.07031*.

Usenko, V., Engel, J., Stückler, J., and Cremers, D. (2016). Direct visual-inertial odometry with stereo cameras. In *Robotics and Automation (ICRA), 2016 IEEE International Conference On*, pages 1885–1892. IEEE.

Vidal, A. R., Rebecq, H., Horstschaefer, T., and Scaramuzza, D. (2017). Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios. *arXiv:1709.06310 [cs]*.

Vidal, A. R., Rebecq, H., Horstschaefer, T., and Scaramuzza, D. (2018). Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001.

Vissiere, D., Martin, A., and Petit, N. (2007). Using magnetic disturbances to improve IMU-based position estimation. In *Control Conference (ECC), 2007 European*, pages 2853–2858. IEEE.

Wagner, R., Birbach, O., and Frese, U. (2011). Rapid development of manifold-based graph optimization systems for multi-sensor calibration and SLAM. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3305–3312.

Weiss, S., Achtelik, M. W., Lynen, S., Chli, M., and Siegwart, R. (2012). Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference On*, pages 957–964. IEEE.

Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. (2012). Kintinuous: Spatially extended kinectfusion.

Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., and Davison, A. (2015). ElasticFusion: Dense SLAM without a pose graph. Robotics: Science and Systems.

Wu, K., Ahmed, A., Georgiou, G. A., and Roumeliotis, S. I. (2015). A Square Root Inverse Filter for Efficient Vision-aided Inertial Navigation on Mobile Devices. In *Robotics: Science and Systems*. Citeseer.

Yang, J., Nguyen, M. N., San, P. P., Li, X., and Krishnaswamy, S. (2015). Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *IJCAI*, pages 3995–4001.

Yang, N., Wang, R., Gao, X., and Cremers, D. (2017). Challenges in Monocular Visual Odometry: Photometric Calibration, Motion Bias and Rolling Shutter Effect. *arXiv:1705.04300 [cs]*.

Yang, Z. and Shen, S. (2016). Monocular Visual-Inertial State Estimation With Online Initialization and Camera-IMU Extrinsic Calibration. *IEEE Transactions on Automation Science and Engineering*, PP(99):1–13.

Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. (2016). LIFT: Learned Invariant Feature Transform. In *Proceedings of the European Conference on Computer Vision*, Amsterdam, Nederlands.

Yu, H. and Mourikis, A. I. (2015). Vision-aided inertial navigation with line features and a rolling-shutter camera. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference On*, pages 892–899. IEEE.

Yu, H. and Mourikis, A. I. (2017). Edge-based visual-inertial odometry. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6670–6677.

Zhang, J., Kaess, M., and Singh, S. (2014). Real-time depth enhanced monocular odometry. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference On*, pages 4973–4980. IEEE.

Zhang, J. and Singh, S. (2015). Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE.

Zhang, T., Wu, K., Su, D., Huang, S., and Dissanayake, G. (2017a). An Invariant-EKF VINS Algorithm for Improving Consistency. *arXiv:1702.07920 [cs]*.

Zhang, Z. (2000). A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22.

Zhang, Z., Liu, S., Tsai, G., Hu, H., Chu, C.-C., and Zheng, F. (2017b). PIRVS: An Advanced Visual-Inertial SLAM System with Flexible Sensor Fusion and Hardware Co-Design. *arXiv:1710.00893 [cs]*.

Zhang, Z., Suleiman, A. A., Carlone, L., Sze, V., and Karaman, S. (2017c). Visual-Inertial Odometry on Chip: An Algorithm-and-Hardware Co-design Approach.

Zheng, F., Tsai, G., Zhang, Z., Liu, S., Chu, C.-C., and Hu, H. (2018). PI-VIO: Robust and Efficient Stereo Visual Inertial Odometry using Points and Lines. *arXiv:1803.02403 [cs]*.

Zhu, A., Atanasov, N., and Daniilidis, K. (2017). Event-based visual inertial odometry. In *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog*, volume 3.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Amélioration des méthodes de navigation vision-inertiel par exploitation des perturbations magnétiques stationnaires de l'environnement
Mots clés : Fusion de capteurs, Navigation, Odométrie Visuelle, SLAM, Filtrage Non-linéaire

Résumé : Cette thèse s'intéresse au problème du positionnement (position et orientation) dans un contexte de réalité augmentée et aborde spécifiquement les solutions à base de capteurs embarqués.
Aujourd'hui, les systèmes de navigation vision-inertiel commencent à combler les besoins spécifiques de cette application. Néanmoins, ces systèmes se basent tous sur des corrections de trajectoire issues des informations visuelles à haute fréquence afin de pallier la rapide dérive des capteurs inertiels bas-coûts. Pour cette raison, ces méthodes sont mises en défaut lorsque l'environnement visuel est défavorable.
Parallèlement, des travaux récents menés par la société Sysnav ont démontré qu'il est possible de réduire la dérive de l'intégration inertielle en exploitant le champ magnétique, grâce à un nouveau type d'UMI bas-coût composée – en plus des accéléromètres et gyromètres traditionnels – d'un réseau de magnétomètres. Néanmoins, cette méthode est également mise en défaut si des hypothèses de non-uniformité et de stationarité du champ magnétique ne sont pas vérifiées localement autour du capteur.
Nos travaux portent sur le développement d'une solution de navigation à l'estime robuste combinant toutes ces sources d'information: magnétiques, visuelles et inertielles.
Nous présentons plusieurs approches pour la fusion de ces données, basées sur des méthodes de filtrage ou d'optimisation, et nous développons un modèle de prédiction du champ magnétique inspiré d'approximation proposées en inertiel et permettant d'intégrer efficacement des termes magnétiques dans les méthodes d'ajustement de faisceaux. Les performances de ces différentes approches sont évaluées sur des données réelles et nous démontrons le bénéfice de la fusion de données comparées aux solutions vision-inertielles ou magnéto-inertielles. Des propriétés théoriques de ces méthodes liées à la théorie de l'invariance des estimateurs sont également étudiées.

Title : Improving Visual-Inertial Navigation Using Stationary Environmental Magnetic Disturbances
Keywords : Sensor Fusion, Navigation, Visual Odometry, SLAM, Non-linear Filtering

Abstract : This thesis addresses the issue of positioning in 6-DOF that arises from augmented reality applications and focuses on embedded sensors based solutions.
Nowadays, the performance reached by visual-inertial navigation systems is starting to be adequate for AR applications. Nonetheless, those systems are based on position correction from visual sensors involved at a relatively high frequency to mitigate the quick drift of low-cost inertial sensors. This is a problem when the visual environment is unfavorable.
In parallel, recent works have shown it was feasible to leverage magnetic field to reduce inertial integration drift thanks to a new type of low-cost sensor, which includes – in addition to the accelerometers and gyrometers – a network of magnetometers. Yet, this magnetic approach for dead-reckoning fails if stationarity and non-uniformity hypothesis on the magnetic field are unfulfilled in the vicinity of the sensor.
We develop a robust dead-reckoning solution combining simultaneously information from all these sources: magnetic, visual, and inertial sensor. We present several approaches to solve for the fusion problem, using either filtering or non-linear optimization paradigm and we develop an efficient way to use magnetic error term in a classical bundle adjustment that was inspired from already used idea for inertial term. We evaluate the performance of these estimators on data from real sensors. We demonstrate the benefits of the fusion compared to visual-inertial and magneto-inertial solutions. Finally, we study theoretical properties of the estimators that are linked to invariance theory.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France