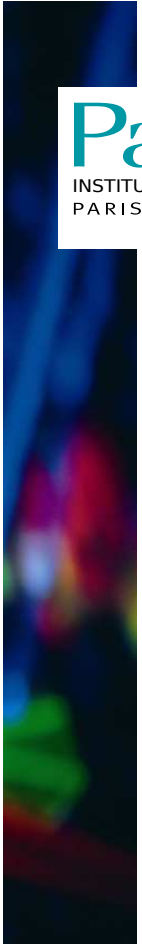
Augmented reality : the fusion of vision and navigation

Nadège Zarrouati-Vissière



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n°432: Sciences des Métiers de l'Ingénieur

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure des Mines de Paris

Spécialité « Mathématique et Automatique »

présentée et soutenue publiquement par

Nadège Zarrouati-Vissière

le 20 décembre 2013

La réalité augmentée: fusion de vision et navigation.

Augmented reality: the fusion of vision and navigation.

Directeur de thèse: **Pierre ROUCHON**

Jury

M. Tarek HAMEL, Professeur, I3S, Université de Nice-Sophia-Antipolis	Rapporteur
M. Gabriel PEYRE, Chargé de recherche, CEREMADE, Université Paris-Dauphine	Rapporteur
M. Hisham ABOU-KANDIL, Professeur, MOSS, ENS Cachan	Président du jury
M. Jean-Jacques SLOTINE, Professeur, NSL, MIT	Examinateur
Mme Karine BEAUCHARD, Chargée de recherches, CMLS, Ecole Polytechnique	Examinateur
M. Pierre ROUCHON, Professeur, CAS, MINES ParisTech	Examinateur
M. Mathieu HILLION, Docteur, SYSNAV	Examinateur

MINES ParisTech
Centre Automatique et Systèmes (CAS)
60, Bd Saint Michel - 75006 Paris - France

T H È S E

Nadège Zarrouati-Vissière

Direction Générale de l'Armement
7-9 rue des Mathurins 9220 Bagneux
France.

E-mail: nadege.zarrouati@mines-paristech.fr

Key words. - Augmented reality, navigation systems, vision, SLAM algorithms, bias estimation, depth, observability, differential geometry, MEMS sensors, RGBD data
Mots clés. - Réalité augmentée, systèmes de navigation, vision, algorithmes SLAM, estimation de biais, profondeur, observabilité, géométrie différentielle, capteurs MEMS, données RGBD

Il n'y a pas de vent favorable pour celui qui ne connaît pas son port.

Sénèque

Tell me and I forget, teach me and I may remember, involve me and I learn.

Benjamin Franklin

Remerciements

Je remercie Hisham Abou-Kandil président du jury, Gabriel Peyré et Tarek Hamel rapporteurs de ma thèse ainsi que les examinateurs Jean-Jacques Slotine, Karine Beauchard, Pierre Rouchon et Mathieu Hillion.

Je tiens à remercier tout particulièrement mon directeur de thèse Pierre Rouchon pour la confiance qu'il m'a faite en me confiant ces travaux de thèse, et pour son attention bienveillante au cours de ces trois années. Cela a été un honneur pour moi de pouvoir travailler à ses côtés et de bénéficier de son recul scientifique.

Je remercie Karine Beauchard pour son enthousiasme et sa réactivité pour répondre avec justesse et compétence à mes diverses questions. J'ai beaucoup apprécié nos séances de travail et nos nombreuses discussions, qu'elles soient d'ordre mathématique ou non.

Je remercie les membres du CAS, en particulier Philippe Martin et Laurent Praly, pour leurs conseils éclairés et leur gentillesse.

Je remercie Jean-Jacques Slotine, venu de loin pour participer au jury de ma thèse, pour m'avoir donné à Boston l'idée et l'envie de faire cette thèse au CAS.

Je remercie mon employeur la DGA pour m'avoir permis de réaliser cette formation par la recherche; je remercie en particulier ma future équipe qui s'est avant même mon arrivée intéressée à mon travail.

Je remercie Sysnav, qui a mis à ma disposition les moyens expérimentaux qui m'ont permis de valider mes travaux théoriques et bien plus encore. J'y ai trouvé une équipe soudée, passionnée par la technique et toujours prête à partager cet enthousiasme, dont les lumières m'ont été précieuses dans le domaine de la navigation inertielle. Je remercie en particulier Mathieu Hillion pour sa constante disponibilité pour répondre à mes questions, Pierre-Jean, Eric et Quentin pour avoir toujours été prêts à m'apporter de l'aide, et Georges pour ses anecdotes et sa longue expérience en navigation.

Je remercie les stagiaires que j'ai eu le plaisir d'encadrer pendant ma thèse, Mian, Rémi et surtout David pour sa gentillesse et sa persévérance.

Enfin, je remercie ma famille: mes parents qui m'ont appris à mener jusqu'au bout mes entreprises, mes soeurs qui m'ont aidée à ne pas les prendre trop au sérieux, et David qui m'a redonné la motivation et la confiance en moi qui parfois me manquaient pour surmonter les difficultés rencontrées.

Augmented reality: the fusion of vision and navigation.

Résumé

Cette thèse a pour objet l'étude d'algorithmes pour des applications de réalité visuellement augmentée. Plusieurs besoins existent pour de telles applications, qui sont traités en tenant compte de la contrainte d'indistinguabilité de la profondeur et du mouvement linéaire dans le cas de l'utilisation de systèmes monoculaires. Pour insérer en temps réel de manière réaliste des objets virtuels dans des images acquises dans un environnement arbitraire et inconnu, il est non seulement nécessaire d'avoir une perception 3D de cet environnement à chaque instant, mais également d'y localiser précisément la caméra. Pour le premier besoin, on fait l'hypothèse d'une dynamique de la caméra connue, pour le second on suppose que la profondeur est donnée en entrée: ces deux hypothèses sont réalisables en pratique. Les deux problèmes sont posés dans le contexte d'un modèle de caméra sphérique, ce qui permet d'obtenir des équations de mouvement invariantes par rotation pour l'intensité lumineuse comme pour la profondeur. L'observabilité théorique de ces problèmes est étudiée à l'aide d'outils de géométrie différentielle sur la sphère unité Riemanienne. Une implémentation pratique est présentée: les résultats expérimentaux montrent qu'il est possible de localiser une caméra dans un environnement inconnu tout en cartographiant précisément cet environnement.

Abstract

The purpose of this thesis is to study algorithms for visual augmented reality. Different requirements of such an application are addressed, with the constraint that the use of a monocular system makes depth and linear motion indistinguishable. The real-time realistic insertion of virtual objects in images of a real arbitrary environment yields the need for a dense Three dimensional (3D) perception of this environment on one hand, and a precise localization of the camera on the other hand. The first requirement is studied under an assumption of known dynamics, and the second under the assumption of known depth: both assumptions are practically realizable. Both problems are posed in the context of a spherical camera model, which yields $SO(3)$-invariant dynamical equations for light intensity and depth. The study of theoretical observability requires differential geometry tools for the Riemannian unit sphere. Practical implementation on a system is presented and experimental results demonstrate the ability to localize a camera in a unknown environment while precisely mapping this environment.

Contents

Introduction

Augmented reality

The emergence of mobile phones since the years 2000, and smartphones in the last few years led to an increasing demand of the public to quick, easy and wide access to personal as well as general information in any situation of everyday's life. For example, it is nowadays a basic requirement to be able to precisely locate oneself wherever in the world, even although the Global Positioning System (GPS), based on radio-frequency signals, is well known to be highly sensitive (among others) to the relative positions of the satellite constellation with respect to the user ([Kaplan and Hegarty, 2005]). Real-time localization is only one example of *augmented reality*, which can be defined as the addition of intangible information to our physical perception of the world: this can be achieved through sound, vision or even touch ([Azuma et al., 1997]). Numerous applications have been reached by augmented reality: a few examples are urban navigation ([Graham et al., 2013]), learning ([Gavish et al., 2013]), surgery or medical applications in general ([Speidel et al., 2013] or [Mousavi Hondori et al., 2013]), 3D modeling ([Brenner, 2005], [Yang et al., 2013] or [Newcombe and Davison, 2010]), robots teaching and learning ([Borrero and Marquez, 2013]), manufacturing ([Ong et al., 2008]), archeology ([Girbacia et al., 2013]), fine arts ([Bovcon et al., 2013]), gesture training ([de Sorbier et al., 2012]), astrophysics ([Vogt and Shingles, 2013]); but also virtual interior design ([Han and Seo, 2013]), fashion ([Ruzanka et al., 2013]), tourism ([Lin et al., 2013]), advertisement ([Shiva and Raajan, 2013]), e-commerce ([Li et al., 2013]) and, last but not least, gaming ([Santoso and Gook, 2012] or [Woun and Tan, 2012]).

The most natural and appealing way to augment reality is by the mean of visual addition, which yields some requirements, of software as well as of hardware types. The obvious hardware requirements are a capture (e.g a video camera, now available on most mobile phones [Khan et al., 2013]), and a display technology: the later has become the last fashion trend with the overwhelming (but still unavailable for the large public at the time of writing) Google Glass product ([Edwards, 2013] or [Ackerman, 2013]). The hidden hardware technology is the processing capability, which is driven by the software complexity. As for the software part, the possibility to add virtual objects to a real image in a realistic way requires the ability to localize the camera with respect to the visible environment, to inlay a 3D model of the object in the physical world, and to project this 3D view in the image. The first part of these requirements is generally solved by Simultaneous Localization and Mapping (SLAM) algorithms: from the

visual input (from one or multiple cameras), which can be aided by inertial inputs present in most portable technologies in the form of Micro-Electro-Mechanical Systems (MEMS) sensors, the camera is localized with respect to a 3D model of the environment. This model can be a complete input, for example a 3D model of the city ([Lothe et al., 2010]); a partial input, e.g fiducial known markers ([Lim and Lee, 2009]); or simply a real-time reconstruction of the environment without any input, which is the domain of application of Structure from Motion (SfM) algorithms ([Wei et al., 2013]). As for the second part of the software requirements, it leads to the necessity of a dense 3D modeling of the environment (at the very least a current camera-centered model) to be able to handle occlusions of the virtual object with the real world ([Kahn, 2013] or [Zollmann and Reitmayr, 2012]).

Dense 3D reconstruction of the environment

Environment reconstruction is tightly related to the SLAM problem [Smith et al., 1990], which can be addressed by nonlinear filtering of observed key feature locations (e.g. [Civera et al., 2008], [Montemerlo et al., 2003]), or by bundle adjustment ([Strasdat et al., 2010b, Strasdat et al., 2010a]). The problem of designing an observer to estimate the depth of single or isolated keypoints has raised a lot of interest, specifically in the case where the relative motion of the carrier is described by constant known ([Abdursul et al., 2004, Dahl et al., 2010]), constant unknown [Heyden and Dahl, 2009], or time varying known ([Dixon et al., 2003, Karagiannis and Astolfi, 2005, Luca et al., 2008, Sassano et al., 2010]) affine dynamics. From a different perspective, the seminal paper of [Matthies et al., 1989] performs incremental depth field refining for the whole field of view via iconic (pixel-wise) linear Extended Kalman filtering. However, estimating a sparse point cloud is insufficient for augmented reality applications, and none of these methods provide an accurate dense depth estimation in a general setting concerning the environment and the camera dynamics.

Motion estimation

For the specific problem of motion estimation and localization, the nature and the quality of the available measurements are critical, and delimit the reachable performances. In the 1950s, expensive Inertial Measurement Units (IMUs) were developed, as missile guidance and control required extremely accurate navigation data [Bezick et al., 2010, Chatfield, 1997, Faurre, 1971]. Tactical grade IMUs, less expensive, enable dead-reckoning techniques over short time periods, but require position fixes provided by GPS [Abbott and Powell, 1999], or combination through data fusion of other sensors outputs [Skog and Handel, 2009, Vissière et al., 2007]. As to recent low-cost IMUs using MEMS technologies, the cumulated error due to the bias of gyroscopes integrated over long time periods induces drift in orientation, which can be bounded. However, from biased accelerometers, only high frequency output can be relied on, since the double integration induces a position error quadratically growing in time; even more severe errors can be induced when accelerometers are used in the frame of attitude estimation. As odometers and velocimeters (e.g. Doppler radar [Uliana et al., 1997], Pitot tube, Electromagnetic (EM) log

sensor), are commonly available technologies in vehicles, mass market applications can combine their linear velocity outputs with angular velocity from low-quality IMUs. Unfortunately, Pitot probes and EM log sensors are known to only provide airspeed and Speed through the water (STW) instead of Speed over ground (SOG): real-time or beforehand bias estimation is necessary to make the measurements from these sensors exploitable. In a different category, the Kinect device has been a huge outbreak in the robotics and vision communities as it provides depth measurements registered at each pixel of a RGB image, at a relatively low cost: this output is named RGBD (Red Green Blue and Depth: refers to the aligned pair of an usual color image and a depth image (RGBD)). This sensor trivially solves the ambiguity issue in ego-motion estimation by a monocular system: while motion is generally estimated *up to as scale factor* (for example by homography estimation [Hamel et al., 2011]), RGBD data provide the input required to solve the SfM problem without resorting to stereo vision techniques [Pollefeys et al., 2004, Zhang and Faugeras, 1992].

Layout of the dissertation

Part I is dedicated to the study of real-time dense 3D reconstruction under the assumptions of known camera motion and known projection model for the on board monocular camera. These assumptions are motivated by the existence of a novel system developed by Sysnav company, providing linear and angular velocity of a system obtained by processing of measurements from gyrometers, accelerometers and magnetometers (see [Vissiere, 2008, Dorveaux, 2011] for a complete description and the historical development of this novel technique, and [Vissière et al., 2008] for a technical patented description). [1] We propose in this part a new estimation method for dense depth data, from known camera dynamics, namely six degrees of freedom (6-DOF) inputs. This method is based on a formulation of the optical flow constraint deriving from the spherical camera model introduced in [Bonnabel and Rouchon, 2009], where the camera dynamics is made explicit. This model is described in Chapter 1 and highlighted by a comparison with other camera models. The principal assumptions and notations are also introduced in this chapter. Variational methods applied to this constraint enable dense depth field estimation (Chapter 2). In addition, a so-called *geometrical flow constraint* can be exploited to design a non-linear observer based on optical flow or depth measurements (Chapter 3). These results are based on [Zarrouati et al., 2012a, Zarrouati et al., 2012b, Zarrouati et al., 2012c], and meet the needs of augmented reality for a real-time dense 3D perception of the visible part of the environment.

The requirement of visual augmented reality concerning the localization of the carrier with respect to its environment is addressed in Part II. We intend in this part to exploit simultaneously image and dense depth data, as provided by the Kinect device, for motion estimation. To bypass the nonlinearities of the system dynamics which usually force to

1. An extension of the distributed magnetometry approach was conducted by application of the method to visual measurements. It was shown in [Zarrouati et al., 2012d] that curvilinear velocity of a grounded rigid body can be estimated by the processing of low-cost images from a stereo vision system and gyrometers measurements. This work, at the stage of development, is not described in this dissertation.

study specific geometrical formulations (e.g the essential space [Soatto et al., 1994], the Plücker coordinates [Mahony and Hamel, 2005]), we lean on the spherical camera model and motion equations described in Chapter 1. In a first approach (Chapter 5), biased angular and linear velocity measurements are supposed to be available. Our work focuses on biases estimation: an important part of the study consists in determining the conditions of observability of such biases. These results appear in [Zarrouati-Vissiere et al., 2012] and have been completed by [Zarrouati-Vissière et al., 2013]. Next in Chapter 6, a novel odometry method based on the processing of dense depth images like the ones provided by the Kinect device is described. This method is named geometrical flow velocity estimation. This work has been the subject of a filed patent application [Zarrouati-Vissière et al., 2013b]. Finally, in Chapter 7, these geometrical flow velocity measurements are processed in addition to biased accelerometers, biased gyrometers and magnetometers to provide an estimation of acceleration and angular velocity biases, as well as a filtered estimate of the linear velocity.

In Part III, we describe a new implementation of a SLAM algorithm adapted for augmented reality. The proposed system includes magneto-inertial sensors (accelerometers, gyrometers and magnetometers), and a RGBD sensor. The algorithm is based on an implementation of Parallel Tracking and Mapping algorithm (PTAM), which was adapted to process RGBD and magneto-inertial data instead of the sole monocular RGB data. The vision algorithm, based on pose optimization and parallel bundle adjustment to optimize the map, is made more robust and scaled by the constraints of depth data. In addition, the fusion of visual position, inertial motion and geometrical flow velocity measurements enable a precise continuous localization.

Acronyms

2D Two dimensional.

3D Three dimensional.

6-DOF six degrees of freedom.

EKF Extended Kalman Filter.

EM Electromagnetic.

FAST Features from Accelerated Segment Test.

FSBI Fast Small Blurry Image.

GPS Global Positioning System.

GPU Graphical Processing Unit.

ICP Iterative Closest Point.

IMU Inertial Measurement Unit.

MEMS Micro-Electro-Mechanical Systems.

PDE Partial Differential Equation.

PTAM Parallel Tracking and Mapping algorithm.

PVS Potentially Visible Set.

RANSAC RANdom SAmple Consensus.

RGBD Red Green Blue and Depth: refers to the aligned pair of an usual color image and a depth image.

SBI Small Blurry Image.

SfM Structure from Motion.

SLAM Simultaneous Localization and Mapping.

SOG Speed over ground.

STW Speed through the water.

Part I

From motion to depth

Du mouvement à la profondeur

Chapter 1

Camera models and motion equations

Formation des images et influence du mouvement

La définition la plus générale d'une caméra pourrait être : un système par lequel le monde tridimensionnel est projeté sur une surface bidimensionnelle. La difficulté de la définition réside dans l'estimation de l'opérateur de projection : pendant des siècles, le principal objectif de l'Optique a été d'établir des modèles traduisant la formation d'images ; dans les dernières décades, les scientifiques se sont concentrés sur le problème de calibration de caméras par rapport à un modèle. Dans ce chapitre, différents modèles de caméras sont décrits, du plus simple et largement utilisé sténopé à des modèles plus élaborés. Le modèle sphérique, adopté dans les Parties I et II, peut être considéré comme un modèle unificateur. Les équations de mouvement associées à ce modèle sont explicitées.

The most general definition of a camera system could be: a system projecting the three-dimensional world onto a two-dimensional surface. However, the estimation of the projection operator for any system is far from obvious: for many centuries, it has been the main purpose of optics to establish models representing image formation; in the past decades, scientists focused on the problem of calibration of systems according to a model. In this chapter, different camera models are described, from the simplest and most used pinhole camera model to more sophisticated ones. The spherical model, adopted in Parts I and II can be seen as a unifying model. The motion equations satisfied by such a spherical camera are then developed.

1.1 Camera models

1.1.1 Pinhole model

The pinhole camera model is the simplest as well as the oldest camera model, as it derives from the *camera obscura* (the latin for "darkened room"). Known by Aristotle and Kepler [Lefèvre, 2007] to observe stars and sun eclipses, its lightest implementation consists of a dark box with a small aperture (a pinhole) on one of the sides. It enabled the discovery that light

travels across straight lines since an inverted image of the surrounding environment is projected onto the back side of the box. More complex devices including mirrors, lenses or translucent screens can rectify the image. Apart from its scientific interest, it used to be an entertainment object at a time where people were unfamiliar with two-dimensional views of the world. In addition, although there is no documentary evidence of this idea apart from the paintings themselves ([Gaskell, 1998]), it is widely admitted that Dutch painters such as Vermeer used *camerae obscurae* as an aid to render unusual perspective views (Figure 1.1).

Figure 1.1: *Woman with a pearl necklace*, Johannes Vermeer, Gemäldegalerie, Berlin.

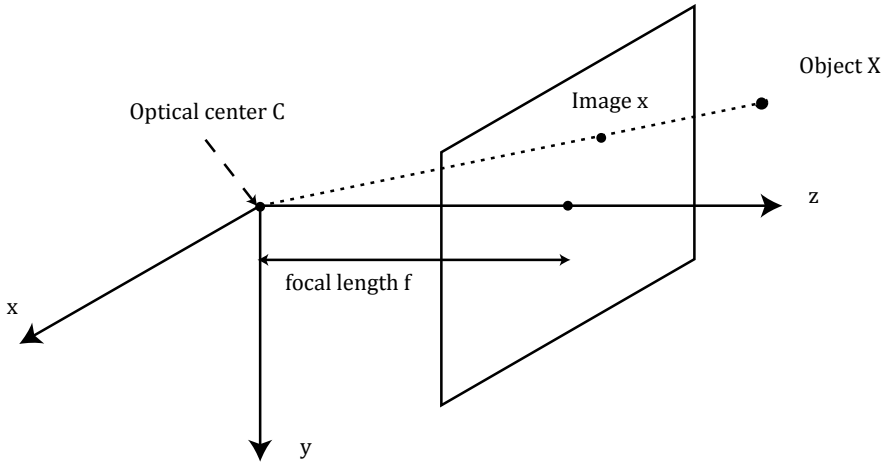


Figure 1.2: Pinhole camera model

The pinhole camera model is a schematic representation of the *camera obscura*: the image is the projection of the world onto a plane (the focal plane) located in front of the optical center of the system (the image in that case is *virtual* and not inverted). The projection is central, so that every light ray travels across the optical center (Figure 1.2). Let us define a direct frame of reference $\mathcal{R}_{cam}$ attached to the camera and centered at the the optical center C: the optical axis (or principal axis), orthogonal to the image plane, is the z-axis, and the x and y axis have horizontal and vertical orientations, pointing in the right and down directions respectively. Let us consider a punctual object, with coordinates (x, y, z) with respect to the camera frame $\mathcal{R}_{cam}$. According to the pinhole camera model, the image of this object is located at the intersection of the image plane (the equation of this plane is $z = f$, where f is the focal length) and on the line (CX). The relation between the object coordinates and the image coordinates, both expressed in $\mathcal{R}_{cam}$, is:

$$(x, y, z)^T \mapsto f(\frac{x}{z}, \frac{y}{z})^T. \tag{1.1}$$

From this relation, the perspective phenomenon can be seen in different ways: first, different punctual objects aligned on the same light ray are all projected on the same punctual image; secondly, the closer extended objects are from the camera center, the bigger they appear on the image; thirdly, the relation between points and image coordinates is non-linear. These considerations led to the introduction of the so-called homogeneous coordinates, consistent with the perspective projection: since all objects on a line that crosses the optical center are equivalent by perspective, their homogeneous coordinates are equal. The homogeneous coordinates of a point X in 3-D space is a vector with 4 components, denoted $\widetilde{X} = (x_1, x_2, x_3, x_4)^T$. Consistency

with perspective yields $\widetilde{X} \sim \lambda\widetilde{X}$ for any non-zero multiplicative factor λ if $\widetilde{X}$ is expressed in $\mathcal{R}_{cam}$. More precisely, $\widetilde{X}$ is the homogeneous representation of X as soon as $x_4 \neq 0$ and

$$x = \frac{x_1}{x_4}, \quad y = \frac{x_2}{x_4} \quad \text{and } z = \frac{x_3}{x_4};$$

the simplest homogeneous representation of X is $(x, y, z, 1)^T$. Homogeneous coordinates are also possible for $2D$–points. These coordinates enable to write the non-linear relation (1.1) as matrix multiplication:

$$\widetilde{X} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \widetilde{x} = \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = P\widetilde{X} \tag{1.2}$$

where P is the matrix projection. This basic projection model can be improved and augmented in numerous ways to better represent image formation in a real camera: this is the purpose of camera calibration. A brief summary of possible extensions is given in the following. Details can be found in the excellent textbook [Hartley and Zisserman, 2000] (chapters 4 and 6); a good (french) review is given in [Boutteau, 2010]; issues about (self-)calibration are addressed in [Pollefeys, 1999]; methods and ready-to-use algorithms for camera calibration were developped by Zhengyou Zhang [Zhang, 2000, Zhang, 2001].

Intrinsic calibration

The intrinsic calibration of a camera is the relation between the coordinates of a $3D$–point $\widetilde{X}$ and its $2D$ projection $\widetilde{x}$, both expressed in the camera reference frame $\mathcal{R}_{cam}$.

- **Principal point offset** In equation (1.2), it is assumed that $\mathcal{R}_{cam}$ is centered at the optical center of the camera, so that the principal point (the intersection between the image plane and the principal axis) is the origin of coordinates in the image plane. In practice, pixels are labeled with positive integer values: the origin of coordinates is located in a corner of the frame (usually the upper-left corner to be consistent with the camera frame definition). Let us denote p_x and p_y the coordinates of the principal point: the projection relation is now

$$\widetilde{X} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \widetilde{x} = \begin{pmatrix} fx + zp_x \\ fy + zp_y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \tag{1.3}$$

- **Pixel coordinates** To express the image coordinates in pixels, instead of meters, let us introduce m_x and m_y the number of pixels per meter in the horizontal and vertical directions, respectively: in particular cases, $m_x \neq m_y$ which results in rectangular pixels; but in most cases, pixels are square ($m_x = m_y$). This yields

$$\widetilde{X} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \widetilde{x}_p = \begin{pmatrix} m_x(fx + zp_x) \\ m_y(fy + zp_y) \\ z \end{pmatrix} = \begin{pmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1.4}$$

where $\alpha_x := m_x f$ and $\alpha_y := m_y f$, are the focal lengths expressed in pixels, and $x_0 := m_x p_x$ and $y_0 := m_y p_y$ are the principal point coordinates expressed in pixels. s is the skew parameter: it describes the orthogonality default of horizontal and vertical axis. Since in practice, internal calibrations can only be performed using pixel coordinates, α_x, α_y, x_0 and y_0 are the values actually estimated, and then f, $p_{x,y}$, $m_{x,y}$ are deduced from the previous relations. Other parameters are useful when performing an internal calibration: for example, resolution (the number of pixels actually seen in each direction, 640 by 480 for a VGA format), sensor size or field-of-view (denoted FOV, the angular section actually seen in each direction) are very meaningful in the choice of a camera objective. Simple relations exist between these parameters (for example, $FOV = 2\tan\left(\frac{resolution}{2\alpha_x}\right)^{-1}$), and various combinations of these parameters can be used to express the projection matrix P.

– **Distortion** In real cameras, perspective is not the only cause of non-linearity: low-cost lenses and optical devices can result in distortions. Distortions can be defined as an additional transformation applied to the "ideal" (undistorted) image coordinates $\widetilde{x}_u$ as expressed in (1.2) (the principal point is the origin of coordinates). The radial distortion function R maps the ideal radius $r_u = \sqrt{x_u^2 + y_u^2}$ onto the real (distorted) radius r_d of image coordinates $\widetilde{x}_d$: $\sqrt{x_d^2 + y_d^2} = r_d = R(r_u)$. Given the coordinates $\widetilde{x}_u$ and the mapping R, the distorted image coordinates are

$$x_d = x_u \frac{R(r_u)}{r_u}, \quad y_d = y_u \frac{R(r_u)}{r_u}. \tag{1.5}$$

Then, translation and conversion to pixels are applied similarly to (1.3) and (1.4). Let us underline that the distortion center and the principal point may be different points. Details on distortion models and calibration procedures can be found in [Devernay and Faugeras, 2001]. Let us cite two main radial distortion mappings:

1. Polynomial distortion model: the inverse mapping R^{-1} is a polynomial function of r_d, whose degree is one of the parameters to estimate during calibration:

$$r_u = R^{-1}(r_d) = r_d(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4 + \dots).$$

The direct mapping R is computed solving a polynomial equation: as the closed-loop solution does not necessarily exist, numerical computations can be considered.

2. Fish-eye model: the direct mapping only depends on one parameter ω (corresponding to the field-of-view of the ideal lens):

$$r_d = R(r_u) = \frac{1}{\omega} \arctan\left(2 r_u \tan\left(\frac{\omega}{2}\right)\right). \tag{1.6}$$

Extrinsic calibration

The extrinsic calibration of a camera is the relation between the coordinates of a $3D$–point $\widetilde{X}_W$ expressed in a fixed external reference frame (usually called *the world* and denoted $\mathcal{R}_W$) and its $2D$ projection $\widetilde{x}$ expressed in the camera reference frame $\mathcal{R}_{cam}$. It depends on the (possibly

varying) orientation and position of the camera with respect to the world. Let us introduce $R(t)$ the rotation matrix representing the orientation of the camera frame $\mathcal{R}_{cam}$ with respect to the world reference frame $\mathcal{R}_W$, and $C(t)$ the position of the optical center of the camera with respect to $\mathcal{R}_W$. Then, the (inhomogeneous) coordinates X_{cam} in $\mathcal{R}_{cam}$ of a $3D-$point with coordinates X_W in $\mathcal{R}_W$ are given by

$$X_{cam} = R(t)(X_W - C(t)).$$

This is linearly written in homogeneous coordinates as

$$\widetilde{X}_{cam} = \begin{pmatrix} R(t) & -R(t)C(t) \\ 0_{1\times 3} & 1 \end{pmatrix} \widetilde{X}_W = K\widetilde{X}_W. \tag{1.7}$$

Sometimes, the translation term $-R(t)C(t)$ in the matrix K is simply replaced by T: one must always keep in mind that this corresponds indeed to the translation of the camera center with respect to a fixed reference frame, but expressed *in the current camera frame*. For a regular camera (pinhole model, no distortion considered) , the relation between the coordinates $\widetilde{X}_W$ of a $3D$ point and its $2D$ projection $\widetilde{x}$ is simply the multiplication of the extrinsic and intrinsic calibration matrices (K and P respectively). The extrinsic matrix K is a member of the *special Euclidean group* $SE(3)$ which is the set of rigid body transformations in three dimensions. More details on $SE(3)$ and more generally on representation of 3D rigid-body kinematics can be found in Appendix C.

When the acquisition system consists of a set of camera, as it happens for a stereo vision system (a stereo rig), extrinsic calibration can refer to the estimation of position and rotation of cameras with respect to each other. This is a crucial step in an algorithm that aims to reconstruct the geometry of a scene by triangulation of matched point of interests between stereoscopic views. In strict triangulation, the baseline (the distance between the optical centers of the cameras) is in a sense the unit of length of the entire map: it is applied as a scale factor in the $3D$ reconstruction process. Stereo Vision and its extension to SfM have been thoroughly studied as a rich field of Computer Vision: description and comparison of existing algorithms can be found in [Faugeras, 1993], [Szeliski, 2010], [Hartley and Zisserman, 2000], and state-of-the-art applications are described in [Nistér et al., 2004, Zhang and Faugeras, 1992].

1.1.2 Other models

In the regular pinhole model, light rays travel across lenses: even when distortion is involved, only *transmission* is considered. Other optical devices, known as catadioptric systems, involve mirrors, which can be very useful to increase the field-of-view of the camera. Consequently, the effect of *reflection* on light rays must be included in the projection model of such a camera. Several classes of catadioptric systems exist, depending on the shape of the reflective systems: planar, conical, spherical, ellipsoidal, hyperboloidal and paraboloidal mirrors have been studied. For any of them, the parameters describing the overall geometry of the surface must satisfy the *single effective viewpoint constraint*: this constraint ensures that the camera system measures the intensity of light rays that all meet in a single point, also called the projection center. The expression of this constraint obviously depends on the class of the catadioptric system: it

was first derived by Baker and Nayar in [Baker and Nayar, 1998] and [Baker and Nayar, 1999]. Then, Geyer showed that all classes of central catadioptric systems were equivalent in the sense that their projection mapping is isomorphic to projective mapping from the sphere to a plane with a projection center on the perpendicular to the plane ([Geyer and Daniilidis, 2000]). Recently, Barreto proposed a unifying geometric representation of central projection systems ([Barreto, 2006]): it covers perspective, catadioptric camera models and includes distortion. In parallel, a general model for imaging systems (including multiple viewpoints systems) was studied ([Grossberg and Nayar, 2001]). These developments suggest that image formation in a generic system is mainly governed by one (or many) viewpoint(s) and a surface of projection, which can be arbitrarily chosen. Once the generic projection model corresponding to these fixed parameters is known, the projection model (either linear or non-linear) of any specific camera, as long as this camera ensures uniqueness of an imaging point, is developed as a mapping between the generic and the specific model.

1.2 Spherical model: notations

The generic model chosen in the following (Parts I and II) is a spherical model: the projection surface is a unit three dimensional sphere ($\mathbb{S}^2$) and the projection center its center, denoted by $C(t)$ as previously. The coordinates of $C(t)$ are expressed in the reference frame $\mathcal{R}_W$. Orientation of the camera is given by the quaternion $q(t)$: any vector ς in the camera frame $\mathcal{R}_{cam}$ corresponds to the vector $q\varsigma q^\star$ in the reference frame $\mathcal{R}_W$ using the identification of vectors as imaginary quaternions (see Appendix C.1.2). A pixel is labeled by the unit vector $\boldsymbol{\eta}$ in the camera frame: $\boldsymbol{\eta}$ belongs to the sphere $\mathbb{S}^2$ and receives the brightness $y(t, \boldsymbol{\eta})$. Thus at each time t, the image produced by the camera is described by the scalar field $\mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto y(t, \boldsymbol{\eta}) \in \mathbb{R}$. The notations of this model are represented in Figure 1.3. In this model, no particular axis is preferred to another, as it is the case in the pinhole model where the optical axis, orthogonal to the image plane partially defines orientation and direction of the camera reference frame.

1.3 Motion equations

1.3.1 Spherical model: the camera in a fixed environment

The motion of the camera is given through the linear and angular velocities $\boldsymbol{v}(t)$ and $\boldsymbol{\omega}(t)$ expressed in the camera reference frame $\mathcal{R}_{cam}$. Linear and angular velocities are related to the time derivatives of position and orientation through the motion equations (see Appendix C and [Radix, 1980])

$$\dot{q} = \frac{1}{2} q\boldsymbol{\omega} \text{ and } \dot{C}(t) = q\boldsymbol{v}q^\star. \tag{1.8}$$

The model adopted for the scene is based on two main assumptions:
- (H1): the scene is a closed, C^1 and convex surface Σ of $\mathbb{R}^3$, diffeomorphic to $\mathbb{S}^2$;
- (H2): the scene is a Lambertian surface;
- (H3): the scene is static.

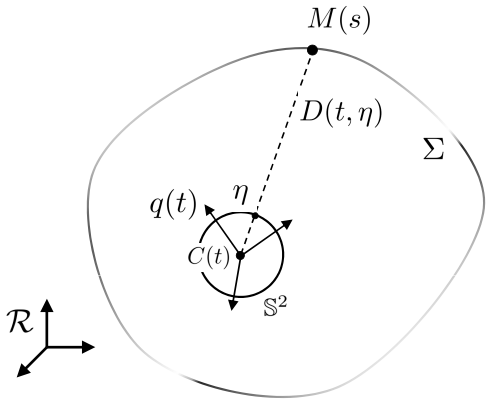


Figure 1.3: Model and notations of a spherical camera in a static environment [Bonnabel and Rouchon, 2009, Zarrouati et al., 2012c].

To be consistent with assumption (H1), the camera is inside the domain $\Omega \subset \mathbb{R}^3$ delimited by $\Sigma = \partial\Omega$.

The first assumption implies that to a point $M \in \Sigma$ corresponds one and only one camera pixel. At each time t, there is a bijection between the position of the pixel given by $\boldsymbol{\eta} \in \mathbb{S}^2$ and the point $M \in \Sigma$. Since the point M is labeled by $s \in \mathbb{S}^2$, this means that for each t, exist two mappings $\mathbb{S}^2 \ni s \mapsto \boldsymbol{\eta} = \phi(t,s) \in \mathbb{S}^2$ and $\mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto s = \psi(t,\boldsymbol{\eta}) \in \mathbb{S}^2$ with $\phi(t,\psi(t,\boldsymbol{\eta})) \equiv \boldsymbol{\eta}$ and $\psi(t,\phi(t,s)) \equiv s$, for all $\boldsymbol{\eta}, s \in \mathbb{S}^2$. To summarize we have:

$$\boldsymbol{\eta} = \phi(t,s) \text{ and } s = \psi(t,\boldsymbol{\eta}) \tag{1.9}$$

where $\psi(t,.)$ and $\phi(t,.)$ are diffeomorphisms of $\mathbb{S}^2$ for every $t > 0$. Moreover, ϕ is implicitly defined by the relation

$$\boldsymbol{\eta} = q(t)^* \frac{\overrightarrow{C(t)M(s)}}{\|\overrightarrow{C(t)M(s)}\|} q(t) \tag{1.10}$$

The density of light emitted by a point $M(s) \in \Sigma$ does not depend on the direction of emission: this results from the second assumption (H2) on the scene. In addition, it is independent of t (H3). This means that $y(t,\boldsymbol{\eta})$ depends only on s: there exists a function $y_\Sigma(s)$ such that

$$y(t,\boldsymbol{\eta}) = y_\Sigma(\psi(t,\boldsymbol{\eta})). \tag{1.11}$$

In other words,

$$\left.\frac{\partial y}{\partial t}\right|_s = 0. \tag{1.12}$$

The distance $\|\overrightarrow{C(t)M(s)}\|$ between the optical center and the object seen in the direction $\boldsymbol{\eta} = \phi(t,s)$ is denoted by $D(t,\boldsymbol{\eta})$.

$$D(t,\boldsymbol{\eta}) := \|\overrightarrow{C(t)M(\psi(t,\boldsymbol{\eta}))}\|. \tag{1.13}$$

Let us consider the time derivative of the distance D attached to a specific point $M(s)$:

$$\begin{aligned}
\left.\frac{\partial D}{\partial t}\right|_s &= \left.\frac{\partial \|\overrightarrow{C(t)M(s)}\|}{\partial t}\right|_s \\
&= \left.\frac{\partial \overrightarrow{C(t)M(s)}}{\partial t}\right|_s \cdot \frac{\overrightarrow{C(t)M(s)}}{\|\overrightarrow{C(t)M(s)}\|} \\
&= -q\boldsymbol{v}q^* \cdot q\boldsymbol{\eta}q^* \quad \text{using (1.8) and (1.10)} \\
&= -\boldsymbol{v}\cdot\boldsymbol{\eta}. \qquad (1.14)
\end{aligned}$$

The last relation holds since a dot product does not depend on the frame of reference in which it is computed. The meaning of equation (1.14) is simple: the distance between the camera and a specific point decreases as the camera moves towards that point.

On the other hand, the direction of observation $\boldsymbol{\eta}$ of a specific object $M(s)$ in the camera frame evolves during the trajectory of the camera according to:

$$\begin{aligned}
\left.\frac{\partial \boldsymbol{\eta}}{\partial t}\right|_s &= \left.\frac{\partial}{\partial t}\left(q^* \frac{\overrightarrow{C(t)M(s)}}{\|C(t)M(s)\|} q\right)\right|_s \text{ using (1.10)} \\
&= q^* \left.\frac{\partial}{\partial t}\left(\frac{\overrightarrow{C(t)M(s)}}{\|C(t)M(s)\|}\right)\right|_s q + \boldsymbol{\eta}\times\boldsymbol{\omega} \text{ using (1.8) and properties of the quaternion product} \\
&= q^*\left(q\left(-\frac{\boldsymbol{v}}{D} + \boldsymbol{\eta}\frac{\boldsymbol{v}\cdot\boldsymbol{\eta}}{D}\right)q^*\right)q + \boldsymbol{\eta}\times\boldsymbol{\omega} \text{ using (1.8)} \\
&= \boldsymbol{\eta}\times\left(\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta}\times\boldsymbol{v}\right). \qquad (1.15)
\end{aligned}$$

In a pure rotational motion ($\boldsymbol{v} = 0$), this equation means that all objects move in the same way in the image: in other words, all pixels are relabeled with a different origin, which is consistent with the spherical geometry of the problem. If the trajectory of the camera is a translation ($\boldsymbol{v} \neq 0$), the objects move differently depending on their position with respect to the camera. Exactly as for a regular perspective camera, there is an intrinsic ambiguity in the projection of the trajectory of an object, due to the multiplicative term $\frac{\boldsymbol{v}}{D}$: a monocular camera system can not capture the scale factor of the perceived environment.

Tracking specific well-defined objects like we just did to obtain equations (1.12), (1.14) and (1.15), is somehow a Lagrangian approach. But these equations can be combined all together in a Eulerian approach, using chain rule differentiation: for any mapping $\mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto h(t,\boldsymbol{\eta}) \in \mathbb{R}$,

$$\left.\frac{\partial h}{\partial t}\right|_s = \left.\frac{\partial h}{\partial t}\right|_{\boldsymbol{\eta}} + \left.\frac{\partial h}{\partial \boldsymbol{\eta}}\right|_t \cdot \left.\frac{\partial \boldsymbol{\eta}}{\partial t}\right|_s.$$

$\left.\frac{\partial h}{\partial \boldsymbol{\eta}}\right|_t$ is the gradient of h with respect to the Riemannian metric of $\mathbb{S}^2$ (see Section A.2), denoted ∇h in the following. Applied to the intensity y and the depth D, it yields

$$\left.\frac{\partial y}{\partial t}\right|_{\boldsymbol{\eta}} = -\nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})), \tag{1.16}$$

$$\left.\frac{\partial D}{\partial t}\right|_{\boldsymbol{\eta}} = -\nabla D \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})) - \boldsymbol{v} \cdot \boldsymbol{\eta}, \tag{1.17}$$

where $\left.\frac{\partial y}{\partial t}\right|_{\boldsymbol{\eta}}$ and $\left.\frac{\partial D}{\partial t}\right|_{\boldsymbol{\eta}}$ stand for partial derivatives of y and D with respect to t: they are denoted in the following $\partial_t y$ and $\partial_t D$ for sake of simplicity.

The first Partial Differential Equation (PDE) (1.16) is well-known in the literature, under a slightly different form, as the *optical flow constraint* (see [Horn and Schunck, 1981] and Part I for more details on optical flow). The second PDE (1.17) was introduced in [Bonnabel and Rouchon, 2009]: the transport term is the same than in (1.16), but a second term breaks the ambiguity of the scale factor. This very interesting property enables to solve many issues encountered in image processing: this will be the main topic of Parts I and II. By analogy with the optical flow constraint (1.16), this equation (1.17) is named in the following *geometrical flow constraint*. Equations (1.16) and (1.17) are $SO(3)$-invariant: they remain unchanged by any rotation described by the quaternion σ and changing $(\boldsymbol{\eta}, \boldsymbol{\omega}, \boldsymbol{v})$ to $(\sigma\boldsymbol{\eta}\sigma^*, \sigma\boldsymbol{\omega}\sigma^*, \sigma\boldsymbol{v}\sigma^*)$.

1.3.2 Pinhole-adapted model

To use this model with camera data, one needs to write the invariant equations (1.16) and (1.17) with local coordinates on $\mathbb{S}^2$ corresponding to a real camera model. We detail here the computations for a pinhole camera model, but as explained in Chapter 1, any single viewpoint model could be considered, as long as the mapping between a 3D point coordinates in the camera frame and its projection on the image plane is known. For the pinhole camera model, the pixel of coordinates (z_1, z_2) corresponds to the unit vector $\boldsymbol{\eta} \in \mathbb{S}^2$ of coordinates in $\mathbb{R}^3$:

$$\boldsymbol{\eta} = \frac{1}{\sqrt{1 + z_1^2 + z_2^2}} \begin{pmatrix} z_1 \\ z_2 \\ 1 \end{pmatrix}.$$

The optical axis of the camera(z-axis in the notations introduced in Subsection 1.1.1) corresponds here to the direction z_3. Directions 1 and 2 correspond respectively to the horizontal axis from left to right and to the vertical axis from top to bottom on the image frame (x-axis and y-axis respectively).

The gradients ∇y and $\nabla \Gamma$ must be expressed with respect to z_1 and z_2. Let us detail this derivation for y. We use two properties of the Riemannian gradient:

- ∇y is tangent to $\mathbb{S}^2$, thus $\nabla y \cdot \boldsymbol{\eta} = 0$;
- the differential dy corresponds on one hand to $\nabla y \cdot d\boldsymbol{\eta}$ and on the other hand to $\frac{\partial y}{\partial z_1} dz_1 + \frac{\partial y}{\partial z_2} dz_2$.

The computation of $d\boldsymbol{\eta}$ yields

$$d\boldsymbol{\eta} = \frac{1}{(1 + z_1^2 + z_2^2)^{3/2}} \begin{pmatrix} 1 + z_2^2 \\ -z_1 z_2 \\ -z_1 \end{pmatrix} dz_1 + \frac{1}{(1 + z_1^2 + z_2^2)^{3/2}} \begin{pmatrix} -z_1 z_2 \\ 1 + z_1^2 \\ -z_2 \end{pmatrix} dz_2. \tag{1.18}$$

Then, by identification of ∇y with a vector of $\mathbb{R}^3$ expressed in the camera frame, one writes

$$\nabla y = \begin{pmatrix} \nabla y_1 \\ \nabla y_2 \\ \nabla y_3 \end{pmatrix}.$$

This yields for dy

$$\nabla y \cdot d\boldsymbol{\eta} = \frac{1}{(1+z_1^2+z_2^2)^{3/2}}[(\nabla y_1(1+z_2^2) - \nabla y_2(z_1 z_2) - \nabla y_3(z_1))dz_1 + (-\nabla y_1(z_1 z_2) + \nabla y_2(1+z_1^2) - \nabla y_3(z_2))dz_2]. \quad (1.19)$$

Then, $\nabla y \cdot \boldsymbol{\eta} = 0$ yields

$$\nabla y_3 = -\nabla y_1 z_1 - \nabla y_2 z_2. \quad (1.20)$$

Equation (1.20) is plugged in (1.19), then by identification of dy with $\frac{\partial y}{\partial z_1}dz_1 + \frac{\partial y}{\partial z_2}dz_2$, one gets:

$$\nabla y = \sqrt{1+z_1^2+z_2^2}\begin{pmatrix} \frac{\partial y}{\partial z_1} \\ \frac{\partial y}{\partial z_2} \\ -z_1\frac{\partial y}{\partial z_1} - z_2\frac{\partial y}{\partial z_2} \end{pmatrix}. \quad (1.21)$$

Similarly we get the three coordinates of $\nabla\Gamma$. Injecting these expressions in (1.16) and (1.17) yields the following PDEs corresponding to (1.16) and (1.17) in local pinhole coordinates:

$$\partial_t y = -\frac{\partial y}{\partial z_1}\left(z_1 z_2\omega_1 - (1+{z_1}^2)\omega_2 + z_2\omega_3 + \frac{1}{D}\sqrt{1+z_1^2+z_2^2}(-v_1 + z_1 v_3)\right) - \frac{\partial y}{\partial z_2}\left((1+{z_2}^2)\omega_1 - z_1 z_2\omega_2 - z_1\omega_3 + \frac{1}{D}\sqrt{1+z_1^2+z_2^2}(-v_2 + z_2 v_3)\right) \quad (1.22)$$

$$\partial_t D = -\frac{\partial D}{\partial z_1}\left(z_1 z_2\omega_1 - (1+{z_1}^2)\omega_2 + z_2\omega_3 + \frac{1}{D}\sqrt{1+z_1^2+z_2^2}(-v_1 + z_1 v_3)\right) - \frac{\partial D}{\partial z_2}\left((1+{z_2}^2)\omega_1 - z_1 z_2\omega_2 - z_1\omega_3 + \frac{1}{D}\sqrt{1+z_1^2+z_2^2}(-v_2 + z_2 v_3)\right) - (z_1 v_1 + z_2 v_2 + v_3) \quad (1.23)$$

where v_1, v_2, v_3, ω_1, ω_2, ω_3 are the components of linear and angular velocities in the camera frame.

Chapter 2

Formulation as a minimization problem

La profondeur, solution d'un problème inverse

Dans ce chapitre sont décrites les méthodes variationnelles et leurs applications à l'estimation du flot optique. Les plus connues de ces techniques d'estimation, la méthode d'Horn-Schunck et la méthode TV-L^1, sont détaillées. De telles méthodes peuvent être adaptées à l'estimation de profondeur par la minimisation de notre contrainte de flot optique spécifique (1.16) plutôt que de la contrainte de flot optique classique : le mouvement apparaissant explicitement, le problème est bien posé au sens où, à chaque pixel de l'image, une équation scalaire est associée à l'unique inconnue du problème, la profondeur. Ces méthodes ont été présentées dans [Zarrouati et al., 2012a, Zarrouati et al., 2012b, Zarrouati et al., 2012c].

In this chapter, variational methods and their application to optical flow estimation are described. The most well-known, the Horn and Schunck and the TV-L^1 optical flow estimation methods are detailed. Such methods can be adapted to depth estimation by the minimization of our invariant optical flow constraint (1.16) where motion is made explicit, instead of the regular optical flow constraint: the problem is well-posed and geometrically constrained in the sense that at each pixel, one scalar equation is associated to the only unknown of the problem, the depth. These methods were presented in [Zarrouati et al., 2012a, Zarrouati et al., 2012b, Zarrouati et al., 2012c].

2.1 Variational methods applied to optical flow estimation

2.1.1 Optical flow

In [Horn and Schunck, 1981], Horn and Schunck described a method to compute the optical flow, defined as "the distribution of apparent velocities of movement of brightness patterns in an image". The entire method is based on the assumption that the light intensity emitted by a specific object does not depend on time or direction of observation (assumptions (H2) and (H3) of the Section 1.3.1 and the consequent equation (1.12)). Provided that the same physical point

is observed in two different images y_1 and y_2, the following equality holds:

$$y_1(z_1, z_2) = y_2(z_1 + \delta z_1, z_2 + \delta z_2)$$

where $z := (z_1, z_2)$ is the position of the point in the first image, and $(\delta z_1, \delta z_2)$ is the apparent displacement of the point between the two images. If the images are only separated by a short time interval δt, it is natural to introduce an apparent velocity vector $(V_1, V_2) := \frac{1}{\delta t}(\delta z_1, \delta z_2)$. Extending this apparent velocity concept to the whole image yields a velocity field $V(t, z) := (V_1, V_2)(t, z_1, z_2)$ named optical flow. First order differentiation in both time and space yields the so-called optical flow constraint:

$$\frac{\partial y}{\partial t} + V_1 \frac{\partial y}{\partial z_1} + V_2 \frac{\partial y}{\partial z_2} = 0. \tag{2.1}$$

This equation can be identified with equation (1.22): an explicit expression of V_1 and V_2 with respect to velocity and depth is given in the following (see equations (2.10) and (2.11)).

2.1.2 Variational methods applied to inverse problems

An important field of image processing is devoted to image restoration: the object is to improve the quality of a given damaged image. The term "damage" covers a wide range of effects, from additional noise to missing pixel data or blur. The usual formulation of restoration is the following: one aims to reconstruct the ideal input y_0 given an output y (the image) related by

$$y = \mathcal{K} y_0 + \nu$$

where ν designates noise and $\mathcal{K}$ is a linear operator that describes the "damaging" effect. For example, to restore

- a noisy image: $\mathcal{K}$ is simply the identity (noise is already taken into account in ν);
- a blurred image: $\mathcal{K}$ is the convolution with a Gaussian kernel;
- an image where pixel data are missing: $\mathcal{K}$ is the characteristic function $\mathbf{1}_S$ of a discrete set S.

Variational methods (specifically variational minimization) apply to inverse problems as the solution is claimed to be the argument of the following minimized functional

$$\int \|y - \mathcal{K} y_0\| + \alpha^2 J(y)$$

where

- J is a *prior*, in other words a weighting function of the image,
- $\int \|y - \mathcal{K} y_0\|$ is the *data fidelity term*, usually the sum over the whole rectangular frame occupied by the image of the *residuals*, and
- α^2 is a strictly positive parameter allowing to scale the influence of the prior term in the minimization task.

Many priors have been proposed in the recent years, and generally each one is very well adapted to a specific class of images. The most well known are the Sobolev prior and the total variation prior. The data fidelity term usually takes into play the Euclidean norm (L^2) but can be in some cases more sophisticated. For more details on variational minimization applied to inverse problems and the links between variational methods and PDEs methods, see [Peyré, 2013], [Grasmair, 2010], or [Guichard and Morel, 2004]. Variational methods have also been applied to other estimation problems, and in particular to optical flow estimation, for which two formulations are detailed in the following.

2.1.3 Variational method with Sobolev prior

For optical flow estimation, Horn and Schunck proposed the data fidelity term to be replaced by the optical flow constraint (2.1) in L^2 norm. The prior used for J is the Sobolev prior, defined as the L^2 norm of the gradient, applied to optical flow:

$$J(V_1, V_2) = \|\nabla V_1\|_{L^2}^2 + \|\nabla V_2\|_{L^2}^2.$$

This yields that the apparent velocity field $V = (V_1, V_2)$ is estimated for each time t by minimizing versus $W := (W_1, W_2)$ the following cost (the image $\mathcal{I}$ is the rectangle of $\mathbb{R}^2$ occupied by the image)

$$I(W) = \iint_{\mathcal{I}} \left(\left(\frac{\partial y}{\partial t} + W_1 \frac{\partial y}{\partial z_1} + W_2 \frac{\partial y}{\partial z_2} \right)^2 + \alpha^2 (\nabla {W_1}^2 + \nabla {W_2}^2) \right) dz_1 dz_2 \tag{2.2}$$

where ∇ is the gradient operator in the Euclidean plane (z_1, z_2) and the partial derivatives $\frac{\partial y}{\partial t}$, $\frac{\partial y}{\partial z_1}$ and $\frac{\partial y}{\partial z_2}$ are assumed to be known. The Sobolev prior is a regularizing term, and $\alpha > 0$ is a regularization parameter: when used for image restoration, it yields to a formulation where the solution of the minimization problem is the solution of the heat equation and can be explicitly deduced from the image y by convolution with a Gaussian kernel.

Notice that for optical flow estimation, the prior term is necessary to solve a fundamentally ill-posed problem where for each position in an image, two components of a vector are estimated from only one scalar equation. Here, the Sobolev prior comes from the assumption that the velocity field is only submitted to small spatial variations: this is a reasonable assumption in an environment where the depth D is continuous, regular and lower-bounded as shown by the following identification ((2.10) and (2.11)).

Such Horn-Schunk estimation of V at time t is denoted by

$$V_{\text{HS}}(t, z) := (V_{\text{HS}1}(t, z), V_{\text{HS}2}(t, z)).$$

For each time t, minimization of the cost I is computed using usual calculus of variation, as the velocity field solution V_{HS} satisfies the system of Euler-Lagrange equations obtained as first-order stationary equations for I:

$$\begin{aligned}
\frac{\partial y}{\partial z_1}\frac{\partial y}{\partial t} + \left(\frac{\partial y}{\partial z_1}\right)^2 V_{\text{HS}1} + \frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2} V_{\text{HS}2} - \alpha^2 \Delta V_{\text{HS}1} = 0 \\
\frac{\partial y}{\partial z_2}\frac{\partial y}{\partial t} + \frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2} V_{\text{HS}1} + \left(\frac{\partial y}{\partial z_2}\right)^2 V_{\text{HS}2} - \alpha^2 \Delta V_{\text{HS}2} = 0
\end{aligned}$$

with boundary conditions $\frac{\partial V_{\text{HS1}}}{\partial n} = \frac{\partial V_{\text{HS2}}}{\partial n} = 0$ (n the normal to $\partial\mathcal{I}$). Here Δ is the Laplacian operator in the Euclidean space (z_1, z_2). The numerical resolution is usually based on

1. **Computations of $\frac{\partial y}{\partial z_1}$, $\frac{\partial y}{\partial z_2}$ and $\frac{\partial y}{\partial t}$ via differentiation filters (e.g. Sobel filtering) directly from the image data at different times around t.** For example, $\frac{\partial y}{\partial z_1}$ can be approximated using a central finite difference scheme (among others) by

$$\frac{\partial y}{\partial z_1}(z_1, z_2, t) \approx \frac{y(z_1 + \delta, z_2, t) - y(z_1 - \delta, z_2, t)}{2\delta} := Dy_{z_1}(z_1, z_2, t). \tag{2.3}$$

To minimize the impact of noise on this differentiation scheme, a slight smoothing of the resulting derivative along the orthogonal direction (z_2 in this example) is required. Using the above definition (2.3) yields

$$\frac{\partial y}{\partial z_1}(z_1, z_2, t) \approx \frac{1}{4}\left(2Dy_{z_1}(z_1, z_2, t) + Dy_{z_1}(z_1, z_2 - \delta, t) + Dy_{z_1}(z_1, z_2 - \delta, t)\right). \tag{2.4}$$

When δ is the length of the square pixels of the grid where image data are available, (2.4) can be expressed as a convolution operator: an approximation of $\frac{\partial y}{\partial z_1}(z_1, z_2, t)$ is obtained by convolution of y with the following kernel:

$$\frac{1}{8\delta}\begin{pmatrix} 1, & 0, & -1 \\ 2, & 0, & -2 \\ 1, & 0, & -1 \end{pmatrix}.$$

2. **Approximation of ΔV_{HS1} and ΔV_{HS2} by the difference between the weighted mean of V_{HS1} and V_{HS2} on the neighboring pixels and their values at the current pixel**. The most common approximation of the discrete Laplacian $\Delta_D f$ of a function $f(z_1, z_2)$ is given by

$$\Delta_D f \approx \frac{f(z_1 - \delta, z_2) + f(z_1 + \delta, z_2) + f(z_1, z_2 - \delta) + f(z_1, z_2 + \delta) - 4f(z_1, z_2)}{\delta^2}$$

in the case where f is known at every vertex of a square grid with length spacing δ. Thus, a fast computation of $\Delta_D f$ is $\frac{\bar{f}_n - nf}{\delta^2}$ (here $n = 4$) where $\bar{f}_n$ is computed as the convolution of f with K_n, here

$$K_4 = \begin{pmatrix} 0, & 1, & 0 \\ 1, & 0, & 1 \\ 0, & 1, & 0 \end{pmatrix}.$$

This yields the following system of linear equations in $(V_{\text{HS1}}, V_{\text{HS2}})$ at each pixel of the image:

$$\begin{cases} \left(\left(\frac{\partial y}{\partial z_1}\right)^2 + n\frac{\alpha^2}{\delta^2}\right) V_{\text{HS1}} + \left(\frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2}\right) V_{\text{HS2}} = \frac{\alpha^2}{\delta^2}(K_n * V_{\text{HS1}}) - \frac{\partial y}{\partial z_1}\frac{\partial y}{\partial t} \\ \left(\frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2}\right) V_{\text{HS1}} + \left(\left(\frac{\partial y}{\partial z_2}\right)^2 + n\frac{\alpha^2}{\delta^2}\right) V_{\text{HS2}} = \frac{\alpha^2}{\delta^2}(K_n * V_{\text{HS2}}) - \frac{\partial y}{\partial z_2}\frac{\partial y}{\partial t}. \end{cases} \tag{2.5}$$

3. **Iterative resolution (Jacobi scheme) of the linear system (2.5) in $V_{\mathrm{HS}1}$ and $V_{\mathrm{HS}2}$.** Defining $\overline{V_{\mathrm{HS}1}} = \frac{1}{n} K_n * V_{\mathrm{HS}1}$ and $\overline{V_{\mathrm{HS}2}} = \frac{1}{n} K_n * V_{\mathrm{HS}2}$, the inversion of (2.5) at each pixel writes

$$\begin{cases} \left(\left(\frac{\partial y}{\partial z_1}\right)^2 + \left(\frac{\partial y}{\partial z_2}\right)^2 + n\frac{\alpha^2}{\delta^2}\right) V_{\mathrm{HS}1} = \left(n\frac{\alpha^2}{\delta^2} + \left(\frac{\partial y}{\partial z_2}\right)^2\right)\overline{V_{\mathrm{HS}1}} - \left(\frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2}\right)\overline{V_{\mathrm{HS}2}} - \frac{\partial y}{\partial z_1}\frac{\partial y}{\partial t} \\ \left(\left(\frac{\partial y}{\partial z_1}\right)^2 + \left(\frac{\partial y}{\partial z_2}\right)^2 + n\frac{\alpha^2}{\delta^2}\right) V_{\mathrm{HS}2} = \left(n\frac{\alpha^2}{\delta^2} + \left(\frac{\partial y}{\partial z_1}\right)^2\right)\overline{V_{\mathrm{HS}2}} - \left(\frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2}\right)\overline{V_{\mathrm{HS}1}} - \frac{\partial y}{\partial z_2}\frac{\partial y}{\partial t}. \end{cases} \tag{2.6}$$

However, as $\overline{V_{\mathrm{HS}1}}$ and $\overline{V_{\mathrm{HS}2}}$ implicitly depend on $V_{\mathrm{HS}1}$ and $V_{\mathrm{HS}2}$, the closed form solution of the problem (2.6) would require to invert a system of equations whose length is twice the size of an image, which is computationally not optimal. Alternatively, an iterative computation of the solution of this system appears, rewriting (2.6) as

$$\begin{cases} V_{\mathrm{HS}1} = \overline{V_{\mathrm{HS}1}} - \dfrac{\left(\frac{\partial y}{\partial z_1}\right)^2\overline{V_{\mathrm{HS}1}} + \left(\frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2}\right)\overline{V_{\mathrm{HS}2}} + \frac{\partial y}{\partial z_1}\frac{\partial y}{\partial t}}{\left(\left(\frac{\partial y}{\partial z_1}\right)^2 + \left(\frac{\partial y}{\partial z_2}\right)^2 + n\frac{\alpha^2}{\delta^2}\right)} = \mathcal{V}_1(V_{\mathrm{HS}1}, V_{\mathrm{HS}2}, y) \\ V_{\mathrm{HS}2} = \overline{V_{\mathrm{HS}2}} - \dfrac{\left(\frac{\partial y}{\partial z_2}\right)^2\overline{V_{\mathrm{HS}2}} + \left(\frac{\partial y}{\partial z_1}\frac{\partial y}{\partial z_2}\right)\overline{V_{\mathrm{HS}1}} + \frac{\partial y}{\partial z_2}\frac{\partial y}{\partial t}}{\left(\left(\frac{\partial y}{\partial z_1}\right)^2 + \left(\frac{\partial y}{\partial z_2}\right)^2 + n\frac{\alpha^2}{\delta^2}\right)} = \mathcal{V}_2(V_{\mathrm{HS}1}, V_{\mathrm{HS}2}, y) \end{cases}. \tag{2.7}$$

This form shows that $V_{\mathrm{HS}1}$ and $V_{\mathrm{HS}2}$ can be iteratively computed, applying the operators $\mathcal{V}_1$ and $\mathcal{V}_2$ to the result of the previous iteration.

The convergence of this numerical method of resolution was proven in [Mitiche and reza Mansouri, 2004]. Three parameters have a direct impact on the speed of convergence and on the precision: the regularization parameter α, the number of iterations for the Jacobi scheme and the initial values of $V_{\mathrm{HS}1}$ and $V_{\mathrm{HS}2}$ at the beginning of this iteration step. To be specific, α should neither be too small in order to filter noise appearing in differentiation filters applied on y, nor too large in order to have V_{HS} close to V when $\nabla V \neq 0$.

2.1.4 Variational method with total variation prior

In their seminal article [Rudin et al., 1992], Rudin, Osher and Fatemi proposed for image denoising the total variation prior defined by the sum of the gradients in L^1 norm. Its main advantage resides in the fact that edges are better conserved: the smoothing/regularizing effect related to diffusion in the heat equation is attenuated. This prior enables the extension of variational methods to non-smooth functions, and is well-adapted to piecewise constant images (cartoon images). It can however sometimes lead to a staircase effect. The main issue with total variation prior comes from the non differentiability of this prior at point 0. A solution to this problem consists in the following approximation:

$$\int \|\nabla f\| \approx \int \sqrt{\epsilon^2 + \|\nabla f\|^2}$$

where ϵ is a small positive constant, which is differentiable. Another solution is based on dual formulation of the minimization task and was introduced by Chambolle in [Chambolle, 2004]

along with a convergent iterative scheme summarized in the following. The solution of the minimization task

$$\min_y \int ||\nabla y|| + \beta^2 ||y - y_d||^2$$

is given by

$$y = y_d + \frac{1}{2\beta^2} \nabla \cdot \mathbf{p} \tag{2.8}$$

where $\mathbf{p} = (p_1, p_2)$ is the dual variable and $\nabla \cdot \mathbf{p}$ denotes the Euclidean divergence of the vector field $\mathbf{p}$. The dual variable is computed iteratively by

$$\tilde{\mathbf{p}}_n = \mathbf{p}_{n-1} + 2\beta^2 \tau \left(\nabla \left(y_d + \frac{1}{2\beta^2} \nabla \cdot \mathbf{p}_{n-1} \right) \right)$$

and then

$$\mathbf{p}_n = \frac{\tilde{\mathbf{p}}_n}{\max(1, |\tilde{\mathbf{p}}_n|)}$$

where $\mathbf{p}_0 = 0$ and the time step $\tau \leq 1/4$.

When, in addition, the data fidelity term $||y - y_d||$ also involves L^1 norm, the method is known as TV-L^1 minimization, and a solution consists of dividing the problem in two minimization subtasks. An auxiliary variable y_a is introduced, and the functional to minimize is approximated by

$$F_\beta = \int ||\nabla y|| + \alpha^2 ||y_a - y_d|| + \beta^2 ||y - y_a||^2$$

where β is a small positive constant. Then, the minimization process of F_β consists of alternating

1. $$\min_y \int ||\nabla y|| + \beta^2 ||y - y_a||^2$$

   for which a solution is given by (2.8), and

2. $$\min_{y_a} \int \alpha^2 ||\rho(y_a)|| + \beta^2 ||y - y_a||^2$$

where $\rho(y_a)$ designates the residuals associated to the auxiliary variable y_a. The second step can be solved point-wise if the residual $\rho(y_a)$ does not depend on spatial derivatives of y_a.

This method was applied to optical flow estimation in [Chambolle and Pock, 2010] and [Wedel et al., 2009]. In that case, the residual $\rho(y)$ is simply the optical flow constraint, which is *linear* with respect to the optical flow components. This yields that the second step optimization is performed using regular optimization methods (e.g. Karush-Kuhn-Tucker conditions) combined with thresholding steps (see [Wedel et al., 2008]):

$$y_a = y + \begin{cases} \frac{\alpha^2}{2\beta^2} \nabla y & \text{if } \rho(y) < -\frac{\alpha^2}{2\beta^2} ||\nabla y||^2 \\ -\frac{\alpha^2}{2\beta^2} \nabla y & \text{if } \rho(y) > \frac{\alpha^2}{2\beta^2} ||\nabla y||^2 \\ -\rho(y) \frac{\nabla y}{||\nabla y||^2} & \text{if } |\rho(y)| \leq \frac{\alpha^2}{2\beta^2} ||\nabla y||^2 \end{cases} . \tag{2.9}$$

2.2 Variational methods applied to depth estimation

We described in Chapter 1.3 a slightly different form of optical flow constraint, where linear and angular velocities appear explicitly and optical flow as a vector field is implicit. Expressing (1.16) in pinhole coordinates (see (1.22)) yields a precise identification of optical flow components with respect to velocities $\boldsymbol{v}$ and $\boldsymbol{\omega}$ and depth D:

$$\begin{aligned} V_1(t,z) &= f_1(z,\boldsymbol{\omega}(t)) + \frac{1}{D(t,z)} g_1(z,\boldsymbol{v}(t)) \\ V_2(t,z) &= f_2(z,\boldsymbol{\omega}(t)) + \frac{1}{D(t,z)} g_2(z,\boldsymbol{v}(t)) \end{aligned} \tag{2.10}$$

with

$$\begin{aligned} f_1(z,\boldsymbol{\omega}) &:= z_1 z_2 \omega_1 - (1+{z_1}^2)\omega_2 + z_2\omega_3 \\ g_1(z,\boldsymbol{v}) &:= \sqrt{1+z_1^2+z_2^2}(-v_1 + z_1 v_3) \\ f_2(z,\boldsymbol{\omega}) &:= (1+{z_2}^2)\omega_1 - z_1 z_2 \omega_2 - z_1\omega_3 \\ g_2(z,\boldsymbol{v}) &:= \sqrt{1+z_1^2+z_2^2}(-v_2 + z_2 v_3) \end{aligned} \tag{2.11}$$

where $\boldsymbol{\omega} = (\omega_1,\omega_2,\omega_3)^T$ and $\boldsymbol{v} = (v_1,v_2,v_3)^T$ are expressed in the camera reference frame $\mathcal{R}_{cam}$ (see Section 1.3.2 for notations). Assuming that linear and angular velocities are known inputs, the structure of optical flow is well determined: in addition, it is linear with respect to $\Gamma := 1/D$ the inverse of depth. We will describe in the following how one can take advantage of this linear dependence and the knowledge of motion to adapt usual variational methods of optical flow estimation to depth estimation. Let us highlight the fact that depth only appears in the expressions (2.10) multiplied by the terms g_1 and g_2 which directly depend on the linear velocity: in others words, linear motion is necessary to estimate depth. In addition, the quality of depth estimation directly depends on the magnitude of linear velocity. This a well-known effect in stereo-vision where the choice of the *baseline* between the cameras is the result of a trade off between performance and practical considerations.

2.2.1 Estimation of depth with Sobolev priors

Instead of minimizing the cost I given by (2.2) with respect to the optical flow components W_1 and W_2, let us define a new invariant cost J_{HS} associated to our optical flow constraint (1.16),

$$J_{HS}(\Upsilon) = \iint_{\mathcal{I}} \left(\left(\frac{\partial y}{\partial t} + \nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \Upsilon \boldsymbol{\eta} \times \boldsymbol{v})) \right)^2 + \alpha^2 \nabla \Upsilon^2 \right) d\sigma_\eta \tag{2.12}$$

and minimize it with respect to any inverse depth profile $\mathcal{I} \ni \boldsymbol{\eta} \mapsto \Upsilon(t,\boldsymbol{\eta}) \in \mathbb{R}$. The time t is fixed here and $d\sigma_\eta$ is the Riemannian infinitesimal surface element on $\mathbb{S}^2$. $\mathcal{I} \subset \mathbb{S}^2$ is the domain where y is measured and $\alpha > 0$ is the regularization parameter.

The first order stationary condition of J_{HS} with respect to any variation of Υ yields the following invariant PDE characterizing the resulting estimation Γ_{HS} of Γ in pinhole cordinates:

$$\alpha^2 \Delta\Gamma_{\text{HS}} = \left(\frac{\partial y}{\partial t} + \nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \Gamma_{\text{HS}} \boldsymbol{\eta} \times \boldsymbol{v}))\right)(\nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}))) \qquad \text{on } \mathcal{I} \tag{2.13}$$

$$\frac{\partial \Gamma_{\text{HS}}}{\partial \mathbf{n}} = 0 \qquad \text{on } \partial\mathcal{I} \tag{2.14}$$

where $\Delta\Gamma_{\text{HS}}$ is the Laplacian of Γ_{HS} on the Riemannian sphere $\mathbb{S}^2$ and $\partial\mathcal{I}$ is the boundary of $\mathcal{I}$, assumed to be piece-wise smooth and with unit normal vector $\mathbf{n}$.

In pinhole coordinates, recall from Section 1.3.2 that

$$\begin{aligned}
d\sigma_\eta &= \left(1 + z_1^2 + z_2^2\right)^{-3/2} dz_1 dz_2 \\
\nabla\Upsilon^2 &= \left(1 + z_1^2 + z_2^2\right)\left(\frac{\partial \Upsilon}{\partial z_1}^2 + \frac{\partial \Upsilon}{\partial z_2}^2 + (z_1 \frac{\partial \Upsilon}{\partial z_1} + z_2 \frac{\partial \Upsilon}{\partial z_2})^2\right) \\
&\left(\frac{\partial y}{\partial t} + \nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \Upsilon \boldsymbol{\eta} \times \boldsymbol{v}))\right)^2 = (F + \Upsilon G)^2
\end{aligned}$$

where

$$\begin{aligned}
F &\coloneqq \frac{\partial y}{\partial t} + f_1(z, \boldsymbol{\omega}) \frac{\partial y}{\partial z_1} + f_2(z, \boldsymbol{\omega}) \frac{\partial y}{\partial z_2} \\
G &\coloneqq g_1(z, \boldsymbol{v}) \frac{\partial y}{\partial z_1} + g_2(z, \boldsymbol{v}) \frac{\partial y}{\partial z_2}.
\end{aligned} \tag{2.15}$$

Consequently, the first order stationary condition (2.13) reads in (z_1, z_2) coordinates:

$$\begin{aligned}
\Gamma_{\text{HS}} G^2 + FG = \alpha^2 (1 + z_1^2 + z_2^2)^{3/2} \Bigg[& \frac{\partial}{z_1}\left(\frac{1 + z_1^2}{\sqrt{1 + z_1^2 + z_2^2}} \frac{\partial \Gamma_{\text{HS}}}{\partial z_1}\right) \\
& + \frac{\partial}{z_2}\left(\frac{1 + z_2^2}{\sqrt{1 + z_1^2 + z_2^2}} \frac{\partial \Gamma_{\text{HS}}}{\partial z_2}\right) \\
& + \frac{\partial}{z_2}\left(\frac{z_1 z_2}{\sqrt{1 + z_1^2 + z_2^2}} \frac{\partial \Gamma_{\text{HS}}}{\partial z_1}\right) \\
& + \frac{\partial}{z_1}\left(\frac{z_1 z_2}{\sqrt{1 + z_1^2 + z_2^2}} \frac{\partial \Gamma_{\text{HS}}}{\partial z_2}\right)\Bigg]
\end{aligned} \tag{2.16}$$

on the rectangular domain $\mathcal{I} \coloneqq [-\bar{z}_1, \bar{z}_1] \times [-\bar{z}_2, \bar{z}_2]$ ($\bar{z}_1, \bar{z}_2 > 0$ with $\bar{z}_1^2 + \bar{z}_2^2 < 1$).

The right term of (2.16) corresponds to the Laplacian operator on the Riemannian sphere $\mathbb{S}^2$ multiplied by α^2 in pinhole coordinates by identification with (2.13). At the center of the image, it is equivalent to the Euclidean Laplacian, but as the direction of observation $\boldsymbol{\eta}$ moves towards the corners of the rectangular matrix of pixels, (the absolute value of pixel coordinates z_1 and

z_2 increase), the correction due to the reprojection of the equation on $\mathbb{S}^2$ back to a regular plane can not be neglected. As a numerical example, let us consider that the visible portion of the environment is a plane, orthogonal to the optical axis. The equation describing this plane is

$$\Gamma(z_1, z_2) = \frac{\Gamma_0}{\sqrt{1 + z_1^2 + z_2^2}}.$$

The Euclidean Laplacian $\Delta_E\Gamma$ and the Riemannian Laplacian $\Delta_R\Gamma$ of this function write respectively

$$\Delta_E\Gamma(z_1, z_2) = \frac{\Gamma_0(z_1^2 + z_2^2 - 2)}{(1 + z_1^2 + z_2^2)^{5/2}}, \quad \Delta_R\Gamma(z_1, z_2) = -\frac{2\Gamma_0}{\sqrt{1 + z_1^2 + z_2^2}}.$$

Notice that the Riemannian Laplacian $\Delta_R\Gamma$ can be computed by two different methods: first, using the identification of the Laplacian with the right term of the expression (2.16); secondly, using the differential calculus expression (A.4) applied to $\Gamma(\boldsymbol{\eta}) = \Gamma_0\boldsymbol{\eta} \cdot (0, 0, 1)^T$.

Figure 2.1 graphically represents the relative difference $\frac{|\Delta_E\Gamma - \Delta_R\Gamma|}{|\Delta_E\Gamma|}$ over the whole image: the relative error reaches 200% in the corners in the image, for a camera with a total field of view of 60° by 50° (horizontally and vertically, respectively).

The numerical resolution of this scalar diffusion equation providing the estimation Γ_{HS} of Γ is similar to the one used for the Horn-Schunck estimation V_{HS} of V, except for the evaluation of the Laplacian, which is adapted to the spherical model.

2.2.2 Estimation of depth by total variation minimization

For total variation minimization as in subsection 2.1.4, both our optical flow constraint and the regularization term are taken into account in L^1 norm in the functional $J_{TV\text{-}L^1}$ to minimize:

$$J_{TV\text{-}L^1}(\Gamma) = \int_{\mathcal{I}} \left(\alpha^2 \| \frac{\partial y}{\partial t} + \nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \Gamma\boldsymbol{\eta} \times \boldsymbol{v})) \| + \|\nabla\Gamma\| \right) d\sigma_\eta \tag{2.17}$$

where $d\sigma_\eta$ is the Riemannian infinitesimal surface element on $\mathbb{S}^2$ and α is a parameter that weights between the data fidelity and the regularization terms. As described for optical flow estimation, the minimization strategy used is divided in two tasks:

1. For Λ being fixed, find the argument Γ of the minimum of

$$J_1(\Gamma) = \int_{\mathcal{I}} \left(\|\nabla\Gamma\| + \beta^2(\Gamma - \Lambda)^2 \right) d\sigma_\eta \tag{2.18}$$

2. For Γ being fixed, find the argument Λ of the minimum of

$$J_2(\Lambda) = \int_{\mathcal{I}} \left(\alpha^2 \|\rho(\Lambda)\| + \beta^2(\Gamma - \Lambda)^2 \right) d\sigma_\eta \tag{2.19}$$

where $\rho(\Lambda) = \frac{\partial y}{\partial t} + \nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \Lambda\boldsymbol{\eta} \times \boldsymbol{v}))$ is the depth residual.

The main algorithm steps are given by the following:

– Initialization: $\Gamma^0 = 0$, $\Lambda^0 = 0$, $\bar{\Gamma}^0 = 0$ $\mathbf{p}^0 = 0$, $\tau \leq 1/4$

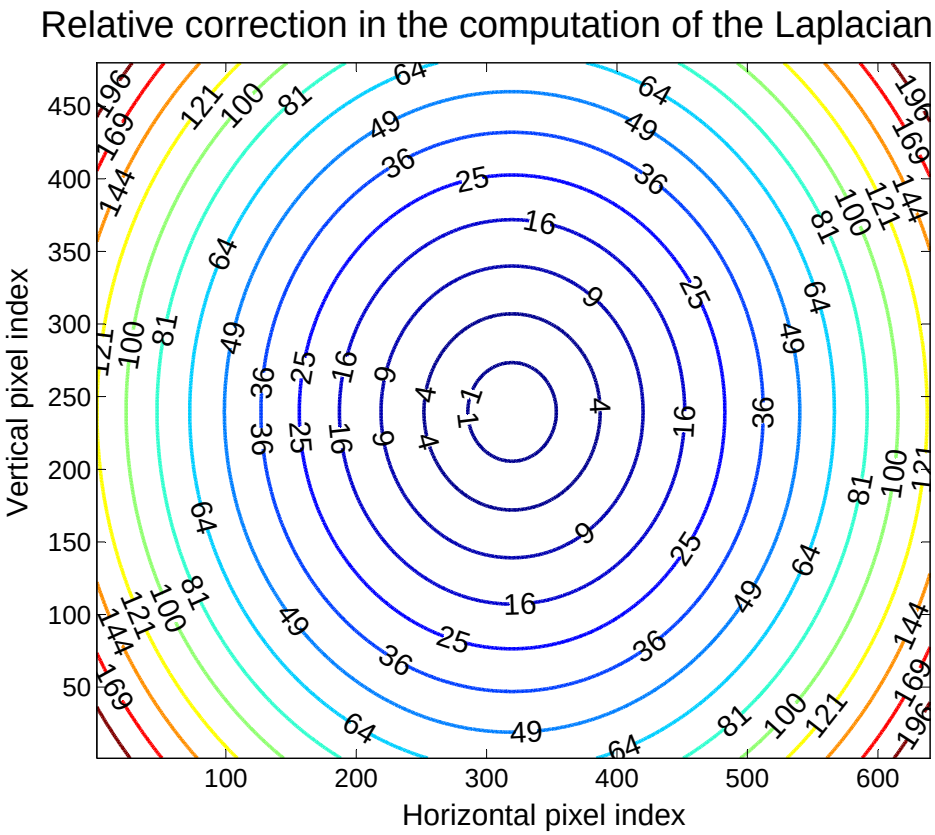


Figure 2.1: Isolines of relative difference between the Riemannian and the Euclidean Laplacian of the depth field when the visible environment is a plane orthogonal to optical axis, for a total field of view of 60° by 50°(horizontally and vertically respectively).

– Iterations:

1. $\mathbf{p}^{n+1} = \mathbf{p}^n + 2\beta^2\tau\nabla\bar{\Gamma}^n$
2. $\mathbf{p}^{n+1} = \frac{\mathbf{p}^{n+1}}{\max(1,\|\mathbf{p}^{n+1}\|)}$
3. $\Gamma^{n+1} = \Gamma^n + \frac{1}{2\beta^2}\nabla\cdot\mathbf{p}^{n+1}$
4. $\Lambda^{n+1} = \Lambda^n + \begin{cases} -\frac{\rho(\Gamma)}{G}, \text{if } \|\rho(\Gamma)\| \le \frac{\alpha^2}{2\beta^2}G^2 \\ \frac{\alpha^2}{2\beta^2}G, \text{if } \rho(\Gamma) < -\frac{\alpha^2}{2\beta^2}G^2 \\ -\frac{\alpha^2}{2\beta^2}G, \text{if } \rho(\Gamma) > \frac{\alpha^2}{2\beta^2}G^2 \end{cases}$
5. $\bar{\Gamma}^{n+1} = 2\Lambda^{n+1} - \Lambda^n$

where $\mathbf{p} = (p_1, p_2)$ and G is defined as in (2.15). The first three steps of the iteration process result from the minimization of J_1, the fourth step is the thresholding method for minimizing J_2 applied to our specific residual and the fifth step is a low-pass filter. More precisely, the fourth step is based on the first order stationary condition in the cases where $\rho(\Lambda) = 0$, $\rho(\Lambda) < 0$ and $\rho(\Lambda) > 0$, respectively. For example, when $\rho(\Lambda) > 0$, the minimum may only be reached if Λ satisfies

$$\alpha^2 G - 2\beta^2(\Gamma - \Lambda) = 0 \tag{2.20}$$

which yields $\Lambda = \Gamma - \frac{\alpha^2}{2\beta^2}G$. Thus, $\rho(\Lambda) > 0$ reads $\rho(\Gamma) > \frac{\alpha^2}{2\beta^2}G^2$.

Chapter 3

Reconstruction of the depth

Filtrage de la profondeur par la contrainte de flot géométrique

Dans ce chapitre, nous montrons comment notre contrainte de flot géométrique invariante (1.17) peut être mise à profit pour la création d'observateurs asymptotiques non-linéaires : les mesures de flot optique ou de profondeur ainsi que la dynamique de la caméra peuvent être utilisées comme entrées de l'observateur de manière à améliorer progressivement l'estimation de profondeur. La convergence de ces observateurs est étudiée, et repose sur des hypothèses géométriques sur l'environnement et la dynamique de la caméra. Ces observateurs et les preuves de convergence ont été présentés dans [Zarrouati et al., 2012a, Zarrouati et al., 2012c].

In this chapter, we show how our invariant geometrical flow constraint (1.17) can be of great benefit in the frame of an asymptotic non-linear observer: either optical flow or depth measurements can be used as inputs, along with camera dynamics, to incrementally refine depth estimation. Convergence of such observers, relying on geometric assumptions concerning the camera dynamics and the environment, is addressed. These observers and the proofs of convergence were presented in [Zarrouati et al., 2012a, Zarrouati et al., 2012c].

3.1 Asymptotic observer based on optical flow measures

From any optical flow estimation, such as V_{HS}, it is reasonable to assume that we have access for each time t, to the components in pinhole coordinates of the vector field

$$\varpi_t : \ \mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto \varpi_t(\boldsymbol{\eta}) = \boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v}) \in T_\eta$$

appearing in (1.17). T_η is the tangent space of the unit sphere $\mathbb{S}^2$ at point $\boldsymbol{\eta}$ (see Appendix A). This vector field can be considered as a measured output for (1.17), expressed as $\varpi_t(\boldsymbol{\eta}) =$

$\boldsymbol{f_t}(\boldsymbol{\eta}) + \frac{1}{D(t,\boldsymbol{\eta})}\boldsymbol{g_t}(\boldsymbol{\eta})$, where $\boldsymbol{f_t}$ and $\boldsymbol{g_t}$ are the vector fields

$$\boldsymbol{f_t}: \ \mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto \boldsymbol{f_t}(\boldsymbol{\eta}) = \boldsymbol{\eta} \times \boldsymbol{\omega} \in T_\eta \tag{3.1}$$

$$\boldsymbol{g_t}: \ \mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto \boldsymbol{g_t}(\boldsymbol{\eta}) = \boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}) \in T_\eta. \tag{3.2}$$

Since D obeys to $\partial_t D = -\nabla D \cdot \varpi_t - \boldsymbol{v} \cdot \boldsymbol{\eta}$, we propose the following asymptotic observer for D, where $\widehat{D}$ denotes the estimation of D:

$$\frac{\partial \widehat{D}}{\partial t} = -\nabla \widehat{D} \cdot \varpi_t - \boldsymbol{v} \cdot \boldsymbol{\eta} + k\boldsymbol{g_t} \cdot (\widehat{D}\boldsymbol{f_t} + \boldsymbol{g_t} - \widehat{D}\varpi_t) \tag{3.3}$$

where ϖ_t, $\boldsymbol{f_t}$ and $\boldsymbol{g_t}$ are known time-varying vector fields on $\mathbb{S}^2$ and $k > 0$ is a tuning parameter. This observer is trivially $SO(3)$ invariant and reads in pinhole coordinates:

$$\frac{\partial \widehat{D}}{\partial t} = -\frac{\partial \widehat{D}}{\partial z_1}V_1 - \frac{\partial \widehat{D}}{\partial z_2}V_2 - (z_1 v_1 + z_2 v_2 + v_3) + k\big(g_1(\widehat{D}f_1 + g_1 - \widehat{D}V_1) + g_2(\widehat{D}f_2 + g_2 - \widehat{D}V_2)\big) \tag{3.4}$$

where V is given by any optical flow estimation and (f_1, f_2, g_1, g_2) are defined by (2.11).

As assumed in section 1.3.1, for each time t, there is a one to one smooth mapping between $\boldsymbol{\eta} \in \mathbb{S}^2$ attached to the camera pixel and the scene point $M(s)$ corresponding to this pixel. This means that, for any $t \geq 0$, the flow $\phi(t, s)$ defined by

$$\left.\frac{\partial \phi}{\partial t}\right|_{(t,s)} = \varpi_t(\phi(t, s)), \quad \phi(0, s) = s \in \mathbb{S}^2$$

defines a time varying diffeomorphism on $\mathbb{S}^2$. As in 1.3.1, ψ denotes the inverse diffeomorphism: $\phi(t, \psi(t, \boldsymbol{\eta})) \equiv \boldsymbol{\eta}$. Assume that $\Gamma(t, \boldsymbol{\eta}) > 0$, $\boldsymbol{v}(t)$ and $\boldsymbol{\omega}(t)$ are uniformly bounded for $t \geq 0$ and $\boldsymbol{\eta} \in \mathbb{S}^2$. This means that the trajectory of the camera center $C(t)$ remains strictly inside the convex surface Σ with minimal distance to Σ. These considerations motivate the assumptions used in the following theorem.

Theorem 1. *Consider $D(t, \boldsymbol{\eta})$ associated to the motion of the camera inside the domain Ω delimited by the scene Σ, a C^1, convex and closed surface as explained in section 1.3.1. Assume that exist $\bar{v} > 0$, $\bar{\omega} > 0$, $d_m > 0$ and $D_M > 0$ such that*

$$\forall t \geq 0, \ \forall \boldsymbol{\eta} \in \mathbb{S}^2, \ |\boldsymbol{v}(t)| \leq \bar{v}, \ |\boldsymbol{\omega}(t)| \leq \bar{\omega}, \ d_m \leq D(t, \boldsymbol{\eta}) \leq D_M.$$

Then, for $t \geq 0$, $D(t, \boldsymbol{\eta})$ is a C^1 solution of (1.17). Consider the observer (3.3) with a C^1 initial condition versus $\boldsymbol{\eta}$, $\widehat{D}(0, \boldsymbol{\eta})$. Then we have the following implications:

1. *$\forall t \geq 0$, the solution $\widehat{D}(t, \boldsymbol{\eta})$ of (3.3) exists, is unique and remains C^1 versus $\boldsymbol{\eta}$. Moreover*

$$t \mapsto \|\widehat{D}(t, \cdot) - D(t, \cdot)\|_{L^\infty} = \max_{\boldsymbol{\eta} \in \mathbb{S}^2} \left|\widehat{D}(t, \boldsymbol{\eta}) - D(t, \boldsymbol{\eta})\right|$$

is decreasing (L^∞ stability);

2. *if additionally for all* $s \in \mathbb{S}^2$, $\int_0^{+\infty} \|g_\tau(\phi(\tau,s))\|^2 d\tau = +\infty$, *then we have for all* $p > 0$,

$$\lim_{t\mapsto+\infty} \int_{\mathbb{S}^2} \left|\widehat{D}(t,\boldsymbol{\eta}) - D(t,\boldsymbol{\eta})\right|^p d\sigma_\eta = 0$$

(convergence in any L^p *topology);*

3. *if additionally there is* $\lambda > 0$ *and* $T > 0$ *such that, for all* $t \geq T$ *and* $s \in \mathbb{S}^2$, $\int_0^t \|g_\tau(\phi(\tau,s))\|^2 d\tau \geq \lambda t$, *then we have, for all* $t \geq T$,

$$\|\widehat{D}(t,\boldsymbol{.}) - D(t,\boldsymbol{.})\|_{L^\infty} \leq e^{-k\frac{1}{D_M}\lambda t} \|\widehat{D}(0,\boldsymbol{.}) - D(0,\boldsymbol{.})\|_{L^\infty}$$

(exponential convergence in L^∞ *topology).*

Remark 1. *Assumptions on* $\int \|g_t(\phi(t,s))\|^2 dt$ *can be seen as a condition of persistent excitations. It should be satisfied for generic motions of the camera.*

Proof. The facts that $\boldsymbol{v}$, $\boldsymbol{\omega}$ and D are bounded and that the scene surface Σ is C^1, closed and convex, ensure that the mapping $\boldsymbol{\eta} = \phi(t,s)$ and its inverse $s = \psi(t,\boldsymbol{\eta})$ are C^1 diffeomorphism on $\mathbb{S}^2$ with bounded derivatives versus s and $\boldsymbol{\eta}$ for all time $t > 0$. Therefore, D is also a function of (t,s). Set $\overline{D}(t,s) = D(t,\phi(t,s))$: in the (t,s) independent variables the PDE (1.17) becomes a set of ordinary differential equations indexed by s:

$$\left.\frac{\partial \overline{D}}{\partial t}\right|_s = -\boldsymbol{v}(t)\cdot\phi(t,s).$$

Thus

$$\overline{D}(t,s) - \overline{D}(0,s) = -\int_0^t \boldsymbol{v}(\tau)\cdot\phi(\tau,s)\, d\tau.$$

Consequently, $1/\overline{D}$ is C^1 versus s and thus $1/D$ is C^1 versus $\boldsymbol{\eta}$.

Set $\overline{\widehat{D}}(t,s) = \widehat{D}(t,\phi(t,s))$. Then

$$\left.\frac{\partial \overline{\widehat{D}}}{\partial t}\right|_s = -\boldsymbol{v}(t)\cdot\phi(t,s) + k\|\boldsymbol{g_t}(\phi(t,s))\|^2 \frac{1}{\overline{D}(t,s)}(\overline{D}(t,s) - \overline{\widehat{D}}(t,s)). \tag{3.5}$$

Set $E = \widehat{D} - D$ and $\overline{E} = \overline{\widehat{D}} - \overline{D}$. Then

$$\left.\frac{\partial \overline{E}}{\partial t}\right|_s = -k\frac{1}{\overline{D}(t,s)}\|\boldsymbol{g_t}(\phi(t,s))\|^2 \overline{E}(t,s) \tag{3.6}$$

Consequently, $\bar{E}$ is well defined for any $t > 0$ and C^1 versus s. Thus E and consequently $\widehat{D} = E + D$ are also well defined for all $t > 0$ and are C^1 versus $\boldsymbol{\eta}$.

Since for any s and $t_2 > t_1 \geq 0$ we have $|\overline{E}(t_2,s)| \leq |\overline{E}(t_1,s)|$, we have also

$$|\overline{E}(t_2,s)| \leq \max_\sigma |\overline{E}(t_1,\sigma)| = \|\widehat{D}(t_1,\boldsymbol{.}) - D(t_1,\boldsymbol{.})\|_{L^\infty}.$$

Thus, taking the max versus s, we get

$$\|\widehat{D}(t_2,\cdot) - D(t_2,\cdot)\|_{L^\infty} \leq \|\widehat{D}(t_1,\cdot) - D(t_1,\cdot)\|_{L^\infty}$$

This ends the proof of the first implication.

Since

$$\overline{E}(t,s) = \overline{E}(0,s)e^{-k\int_0^t \left(\frac{1}{\overline{D}(\tau,s)}\|g_\tau(\phi(\tau,s))\|^2\right)d\tau}$$

we have ($\phi(0,s) \equiv s$).

$$|\overline{E}(t,s)| \leq |E(0,s)|e^{-k\frac{1}{D_M}\int_0^t \|g_\tau(\phi(\tau,s))\|^2\ d\tau}.$$

Take $p > 0$. Then

$$\int_{\mathbb{S}^2} \left|E(t,\boldsymbol{\eta})\right|^p d\sigma_\eta = \int_{\mathbb{S}^2} \left|\overline{E}(t,s)\right|^p \det\left(\frac{\partial\phi}{\partial s}(t,s)\right) d\sigma_s.$$

By assumption $\frac{\partial\phi}{\partial s}$ is bounded. Thus exists $C > 0$ such that

$$\|E(t,\cdot)\|_{L^p} = \int_{\mathbb{S}^2} \left|E(t,\boldsymbol{\eta})\right|^p d\sigma_\eta \leq C \int_{\mathbb{S}^2} \left|\overline{E}(t,s)\right|^p d\sigma_s.$$

When $\int_0^{+\infty} \|g_\tau(\phi(\tau,s))\|^2 d\tau = +\infty$, for each s we have

$$\lim_{t\mapsto+\infty} \overline{E}(t,s) = 0.$$

Moreover $|\overline{E}(t,s)|$ is uniformly bounded by the L^∞ function $E(0,s)$. By Lebesgue dominate convergence theorem $\lim_{t\mapsto+\infty} \|\overline{E}(t,\cdot)\|_{L^p} = 0$. Previous inequality leads to

$$\lim_{t\mapsto+\infty} \|E(t,\cdot)\|_{L^p} = 0$$

which ends the proof of the second implication of Theorem 1.

When, for $t > T$, $\int_0^T \|g_\tau(\phi(\tau,s))\|^2 d\tau \geq \lambda t$, we have,

$$\forall s \in \mathbb{S}^2, |\overline{E}(t,s)| \leq |E(0,s)|e^{-k\frac{1}{D_M}\lambda t}.$$

Thus, for all $s \in \mathbb{S}^2$ we get $|\overline{E}(t,s)| \leq \|E(0,\cdot)\|_{L^\infty}$. Since $\boldsymbol{\eta} \mapsto \psi(t,\boldsymbol{\eta})$ is a diffeomorphism of $\mathbb{S}^2$, we get finally,

$$\forall \boldsymbol{\eta} \in \mathbb{S}^2, |E(t,\boldsymbol{\eta})| \leq \|E(0,\cdot)\|_{L^\infty} e^{-k\frac{1}{D_M}\lambda t}.$$

This proves

$$\|E(t,\cdot)\|_{L^\infty} \leq \|E(0,\cdot)\|_{L^\infty} e^{-k\frac{1}{D_M}\lambda t}$$

and concludes the proof of the theorem. □

3.2 Asymptotic observer based on depth estimation

Instead of relying the observer on optical flow estimation, we can base it on depth estimations as outputs of variational methods described in subsection 2.2. For example, Γ_{HS} is the output of the variational method with Sobolev prior. Then we propose the following observer for D ($k > 0$):

$$\frac{\partial \widehat{D}}{\partial t} = -\nabla \widehat{D} \cdot (\boldsymbol{f_t} + \Gamma_{\text{HS}} \boldsymbol{g_t}) - \boldsymbol{v} \cdot \boldsymbol{\eta} + k(1 - \widehat{D}\Gamma_{\text{HS}}) \tag{3.7}$$

that reads in pinhole coordinates

$$\frac{\partial \widehat{D}}{\partial t} = -\frac{\partial \widehat{D}}{\partial z_1}(f_1 + \Gamma_{\text{HS}} g_1) - \frac{\partial \widehat{D}}{\partial z_2}(f_2 + \Gamma_{\text{HS}} g_2) - (z_1 v_1 + z_2 v_2 + v_3) + k(1 - \widehat{D}\Gamma_{\text{HS}}). \tag{3.8}$$

For this observer we have the following convergence result.

Theorem 2. *Take assumptions of Theorem 1 concerning the scene surface Σ, D, $\boldsymbol{v}$ and $\boldsymbol{\omega}$. Consider the observer (3.7) where Γ_{HS} coincides with $\Gamma = \frac{1}{D}$ and where the initial condition is C^1 versus $\boldsymbol{\eta}$. Then $\forall t \geq 0$, the solution $\widehat{D}(t, \boldsymbol{\eta})$ of (3.7) exists, is unique, remains C^1 versus $\boldsymbol{\eta}$ and*

$$\|\widehat{D}(t,.) - D(t,.)\|_{L^\infty} \leq e^{-k\frac{1}{D}t} \|\widehat{D}(0,.) - D(0,.)\|_{L^\infty}$$

(exponential convergence in L^∞ topology).

The proof is similar to the one of Theorem 1.

Remark 2. *Observers (3.3) and (3.7) are contracting for the metric M defined as*

$$\delta\widehat{D}^T M \delta\widehat{D} := \frac{1}{2}\int_{\mathbb{S}^2} \delta\widehat{D}^2 d_{\sigma_\eta}. \tag{3.9}$$

Details on contraction analysis can be found in [Lohmiller and Slotine, 1996] or [Lohmiller and Slotine, 1998]. This is proven noticing that $\delta\widehat{D}$ satisfies the following system (for observer (3.3))

$$\frac{\partial \delta\widehat{D}}{\partial t} = -\nabla \delta\widehat{D} \cdot \varpi_t + k\boldsymbol{g_t} \cdot (-\delta\widehat{D}\boldsymbol{g_t})$$

Thus using the same notations than in the proof of Theorem 1

$$\left.\frac{\partial \overline{\delta\widehat{D}}}{\partial t}\right|_s = -k\|\boldsymbol{g_t}(\phi(t,s))\|^2 \frac{\overline{\delta\widehat{D}}(t,s)}{\overline{D}(t,s)} \tag{3.10}$$

which yields

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\int_{\mathbb{S}^2} \delta\widehat{D}^2 d_{\sigma_\eta}\right) = -k\int_{\mathbb{S}^2} \|\boldsymbol{g_t}(\phi(t,s))\|^2 \frac{\overline{\delta\widehat{D}}(t,s)^2}{\overline{D}(t,s)} \det\left(\frac{\partial \phi}{\partial s}(t,s)\right) d_{\sigma_s} \leq 0.$$

This proves that the time-derivative of the variation $\delta\widehat{D}$ is decreasing for the metric M.

Chapter 4

Implementations, simulations and experiments

Implémentations

Dans ce chapitre, l'implémentation des méthodes précédentes sont décrites, et des simulations sont présentées. Une expérience préliminaire traitant des données réelles montre que les hypothèses de continuité sur l'environnement ne sont peut-être pas nécessaires : les résultats sont cohérents avec la géométrie réelle de l'environnement filmé. Ces résultats ont été présentés dans [Zarrouati et al., 2012a, Zarrouati et al., 2012b, Zarrouati et al., 2012c].

In this chapter, implementations are described, and simulation results are presented. A preliminary experiment on real data shows that continuity assumptions on the environment may not be necessary, given that the results are consistent with reality. These results were presented in [Zarrouati et al., 2012a, Zarrouati et al., 2012b, Zarrouati et al., 2012c].

4.1 Simulations

4.1.1 Sequence of synthetic images and method of comparison

The non-linear asymptotic observers described in Chapter 3 are tested on a sequence of synthetic images characterized by the following:

- *virtual camera*: the size of each image is 640 by 480 pixels, the frame rate of the sequence is 60 Hz and the field of view is 50 deg by 40 deg;
- *motion of the virtual camera*: the trajectory of the camera is a combination of sinusoids; linear and angular velocities (plotted in Figures 4.1 and 4.2) are computed as the time-derivatives of position and attitude.
- *virtual scene*: the scene is a room of dimensions $3 \times 4 \times 4.5$ meters (width, length and height respectively); the walls are (virtually) painted with a gray pattern, whose intensity varies in horizontal and vertical directions as a sinusoid function;

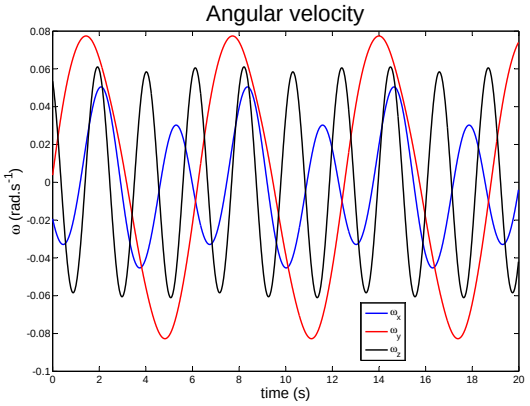


Figure 4.1: Angular velocity of the camera for the virtual sequence.

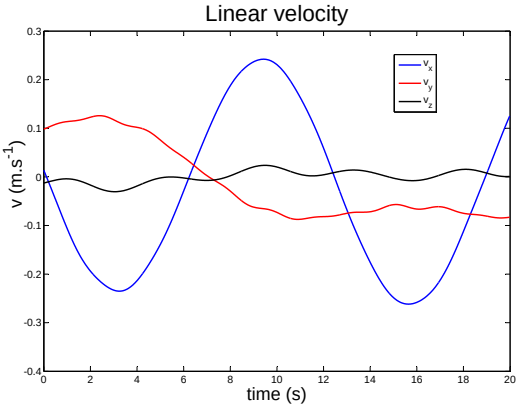


Figure 4.2: Linear velocity of the camera for the virtual sequence.

– *generation of the images*: each pixel of an image has an integer value varying from 1 to 256, directly depending on the intensity of the observed surface in the direction indexed by the pixel, to which a normally distributed noise varying with mean 0 and standard deviation σ is added.

To compare the performances of the various methods, we use the global relative error in the estimation of D, defined as

$$E = \frac{\int_{\mathcal{I}} \frac{|\widehat{D}(t,\boldsymbol{\eta}) - D(t,\boldsymbol{\eta})|}{D(t,\boldsymbol{\eta})} d\sigma_\eta}{\int_{\mathcal{I}} d\sigma_\eta}$$

where D is the true value of the depth field, $\widehat{D}$ is the estimation computed by any of the proposed methods and $\mathcal{I}$ is the image frame.

4.1.2 Implementation of the depth estimation based on optical flow measures

We test on the sequence described in 4.1.1 the depth estimation characterized by the PDE (3.3). The optical flow input V_{HS} or V_{TVL1} (V_1 and V_2 components) is computed by a classical Horn-Schunck method or the TV-L^1 method described in Subsection 2.1.1. Note that convergence Theorem 1 assumes that the domain of definition of the image is the entire unit sphere $\mathbb{S}^2$. Here the field of view of our virtual camera limits this domain to a portion of the sphere $\mathcal{I}$. However, the motion of our virtual camera ensures that most of the points of the scene appearing in the first image stay in the field of view of the camera during the whole sequence. The convergence of the method is only ensured for these points, and Neumann boundary conditions are chosen at the borders where optical flow points toward the inside of the image:

$$\frac{\partial \widehat{D}}{\partial n} = 0 \text{ if } n \cdot V_{\text{HS}} < 0 \tag{4.1}$$

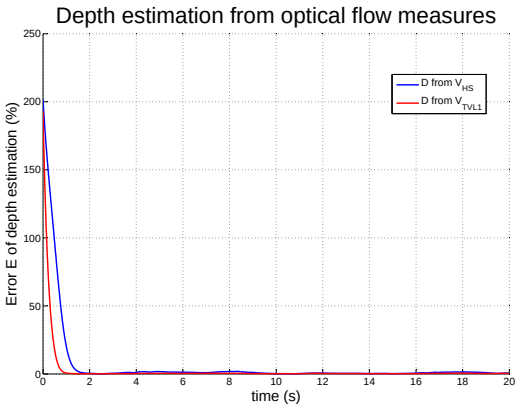


Figure 4.3: Overview of estimation errors of $\widehat{D}$ using the asymptotic observer (3.4) with optical flow input V_{HS} and V_{TVL1}. Images are perfect.

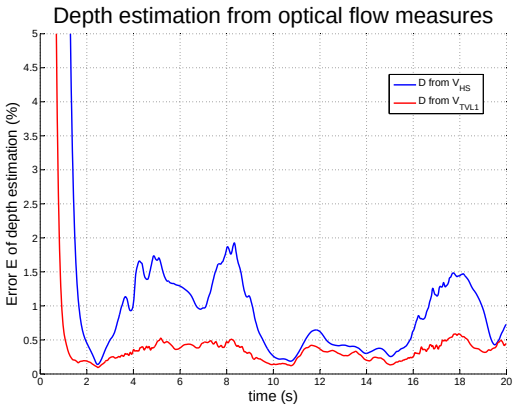


Figure 4.4: Zoom on estimation errors of $\widehat{D}$ using the asymptotic observer (3.4) with optical flow input V_{HS} and V_{TVL1}. Images are perfect.

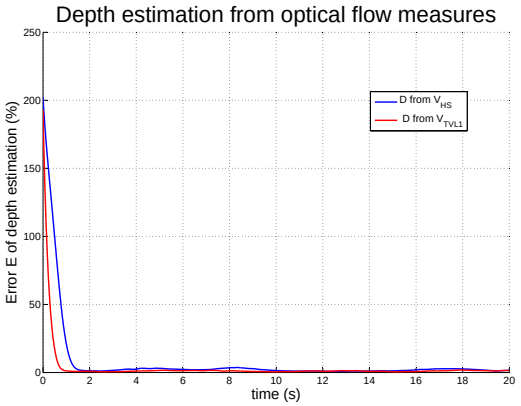


Figure 4.5: Overview of estimation errors of $\widehat{D}$ using the asymptotic observer (3.4) with optical flow input V_{HS} and V_{TVL1}. Images are corrupted by gaussian noise (σ = 1).

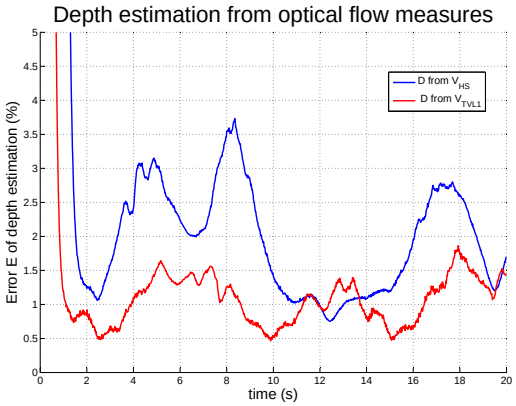


Figure 4.6: Zoom on estimation errors of $\widehat{D}$ using the asymptotic observer (3.4) with optical flow input V_{HS} and V_{TVL1}. Images are corrupted by Gaussian noise (σ = 1).

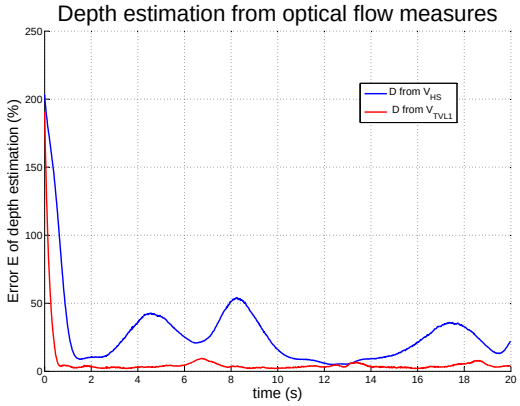


Figure 4.7: Overview of estimation errors of $\widehat{D}$ using the asymptotic observer (3.4) with optical flow input V_{HS} and V_{TVL1}. Images are corrupted by gaussian noise (σ = 10).

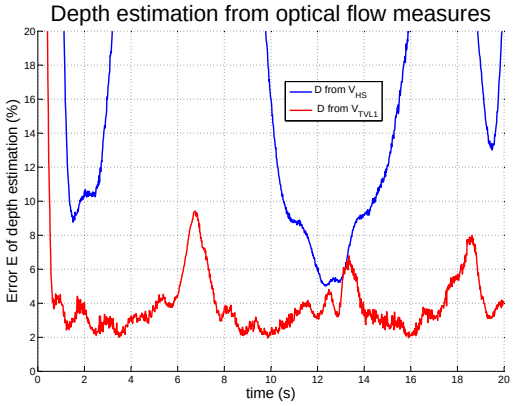


Figure 4.8: Zoom on estimation errors of $\widehat{D}$ using the asymptotic observer (3.4) with optical flow input V_{HS} and V_{TVL1}. Images are corrupted by Gaussian noise (σ = 10).

To test the accuracy and the robustness of this method, the observer is successively implemented on the same sequence of images, on which noise is added. This perturbation is a centered normally distributed noise, characterized by a standard deviation σ. Recall from Subsection 4.1.1 that the image pixel values vary from 1 to 256. Figures 4.3 and 4.4 show the result of this implementation for perfect data (σ = 0); in Figures 4.5 and 4.6, the noise corrupting the images is small (σ = 1); in Figures 4.7 and 4.8, the noise is magnified by ten (σ = 10). The same observer gain $k > 0$ is chosen for the various levels of noise, according to scaling considerations: k = 500 $s.m^{-1}$. These results show that the method is valid for perfect data, but is highly dependent on the quality of estimation of the optical flow: the TV-L^1 method of optical flow estimation, more robust, yields a fast convergence rate (less than 2 seconds), a reasonable accuracy (less than 10% of relative error for σ = 10) and a good robustness to noise. On the contrary, although the method using the Horn-Schunck optical flow estimation has a reasonable accuracy for perfect data, robustness to noise is not here. Note that for both methods, peaks of error occur when the magnitude of linear velocity is close to zero: like in triangulation methods where the baseline must be large enough to accurately reconstruct a 3D position, the quality of the results is degraded when linear displacement is too small.

4.1.3 Implementation of the asymptotic observer based on rough depth estimation

Subsequently, the observer described by (3.7) was applied to the same sequence. The input depth fields Γ_{HS} and Γ_{TVL1} are obtained as the outputs of (2.16) and the algorithm described in Subsection 2.2.2 respectively. The regularization parameter α is chosen accordingly to scaling considerations and taking into account the magnitude of noise: α = 300 $m.s^{-1}$ provides a slow

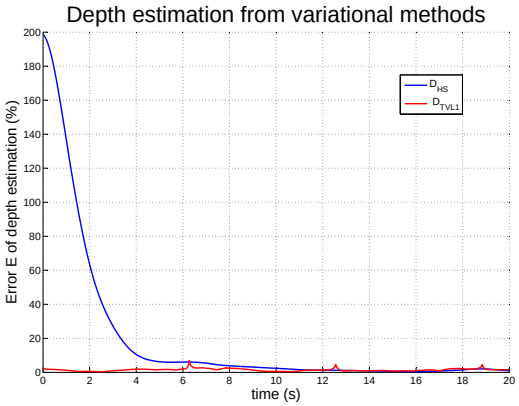


Figure 4.9: Overview of estimation errors of $\widehat{D}$ using the variational methods described in Subsections 2.2.1 and 2.2.2. Images are perfect.

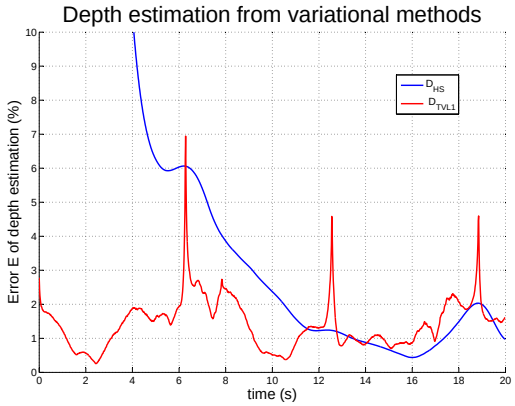


Figure 4.10: Zoom on estimation errors of $\widehat{D}$ using the variational methods described in Subsections 2.2.1 and 2.2.2. Images are perfect.

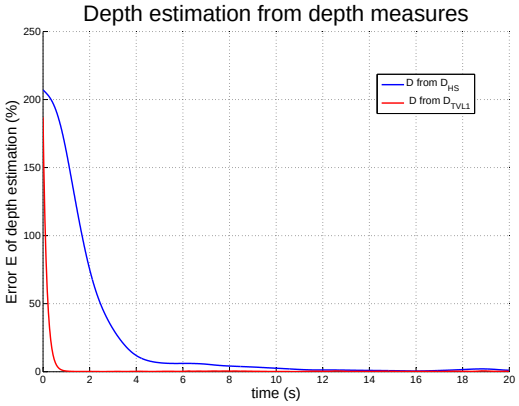


Figure 4.11: Overview of estimation errors of $\widehat{D}$ using the asymptotic observer (3.7) with depth inputs Γ_{HS} and Γ_{TVL1}. Images are perfect.

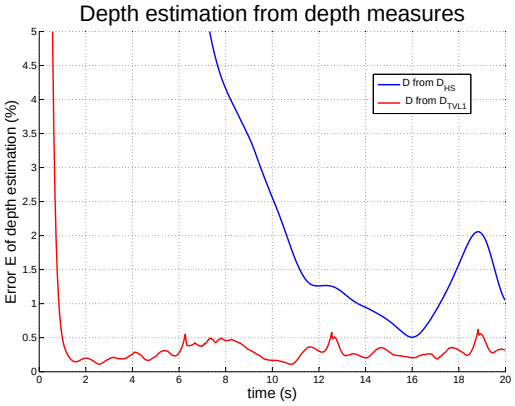


Figure 4.12: Zoom on estimation errors of $\widehat{D}$ using the asymptotic observer (3.7) with depth inputs Γ_{HS} and Γ_{TVL1}. Images are perfect.

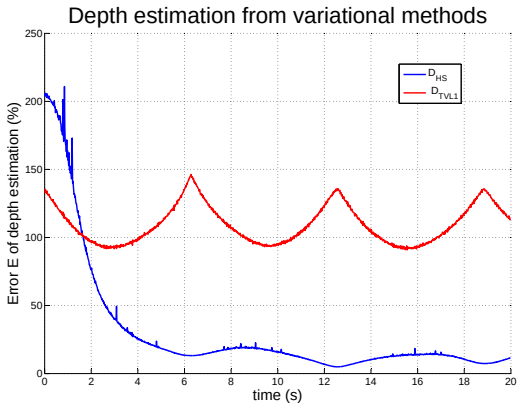


Figure 4.13: Overview of estimation errors of $\widehat{D}$ using the variational methods described in Subsections 2.2.1 and 2.2.2. Images are corrupted by Gaussian noise ($\sigma = 10$).

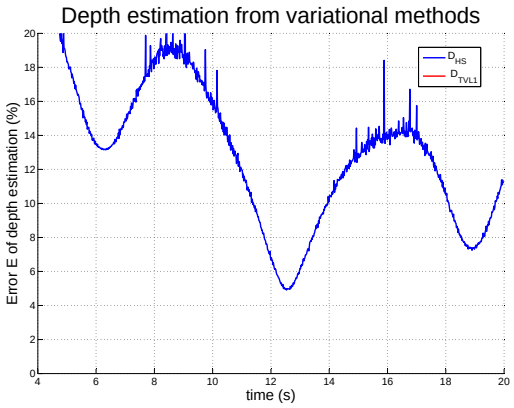


Figure 4.14: Zoom on estimation errors of $\widehat{D}$ using the variational methods described in Subsections 2.2.1 and 2.2.2. Images are corrupted by Gaussian noise ($\sigma = 10$).

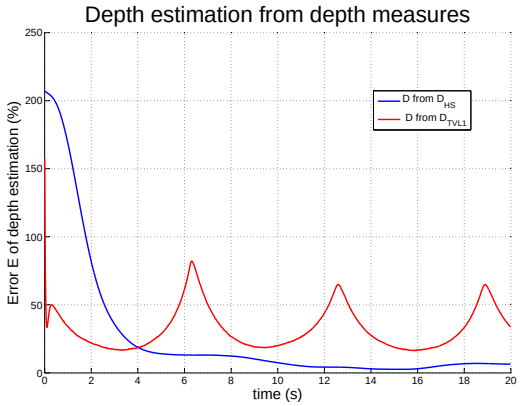


Figure 4.15: Overview of estimation errors of $\widehat{D}$ using the asymptotic observer (3.7) with depth inputs Γ_{HS} and Γ_{TVL1}. Images are corrupted by Gaussian noise ($\sigma = 10$).

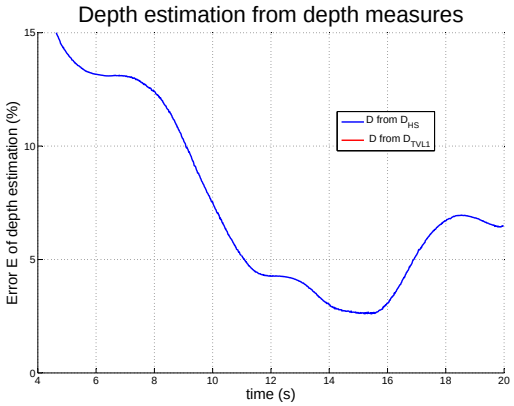


Figure 4.16: Zoom on estimation errors of $\widehat{D}$ using the asymptotic observer (3.7) with depth inputs Γ_{HS} and Γ_{TVL1}. Images are corrupted by Gaussian noise ($\sigma = 10$).

but clean convergence. The correction gain $k = 30\ m.s^{-1}$ enables a rate of convergence similar to the one of the previous observer (see Subsection 4.1.2).

As in the previous subsection , we first test the variational methods for different levels of noise corrupting the image data. For perfect and noisy data ($\sigma = 0$ and $\sigma = 10$ respectively), the error of estimation associated to the input depths Γ_{HS} and Γ_{TVL1} are plotted in Figures 4.9 and 4.10 and in Figures 4.13 and 4.14 (respectively). As expected, the variational method based on Sobolev prior is slow but shows a better robustness than the variational method based on TV-L^1 minimization.

Then, the observer (3.7) is tested, using the previous depth estimates as depth inputs. The results are plotted in Figures 4.11 and 4.12 and in Figures 4.15 and 4.16 respectively. For perfect data, the observer based on TV-L^1 estimations quickly provides almost perfect results (greater bound of 0.5% for the error) but is unusable for noisy data. The observer based on L^2 minimization shows more robustness but is still less accurate than the observer based on TV-L^1 optical flow estimation (the error is always greater than 5%).

4.2 Experiment on real data

Real data have been processed in preliminary experiments, to ensure that the regularity assumptions on the environment are not necessary to get consistent results. To realize the experiments, a camera was fixed on a motorized trolley traveling back and forth on a 2 meter-linear track in about 6 seconds. The resolution of the encoder of the motor enables to know the position and the speed of the trolley with a micrometric precision. This experimental benchmark was set up and provided by the CAS (The Systems and Control Centre of Mines ParisTech) and is shown in Figure 4.17.

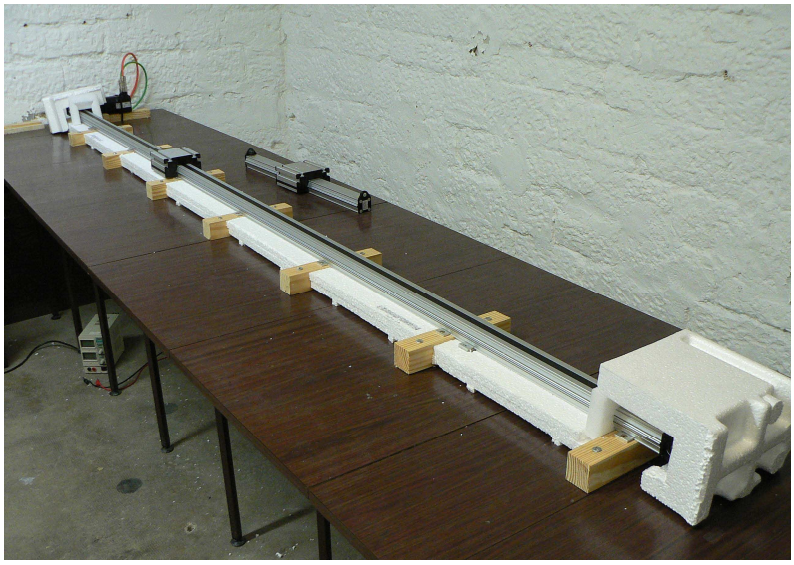


Figure 4.17: The linear track and the motorized trolley used to measure the linear velocity of the system. This experimental benchmark was set up and provided by the CAS (The Systems and Control Centre of Mines ParisTech).

The camera is a Flea2 - Point Grey Research VGA video cameras (640 by 480 pixels) acquiring data at 20.83 fps, with a Cinegon 1.8/4.8 C-mount lens, with an angular field of view of approximately 50 by 40 deg, and oriented orthogonally to the track. It was provided by Sysnav company.

Synchronization of linear velocity measurements and RGB images acquired by the camera has been achieved after the acquisition by comparison of images and specific motion times (e.g. starting point, ending point, turn-around point). This might result in some imprecision that can affect the overall depth estimation.

The scene is a static work environment, with desks, tables, chairs, lamps, lit up by electric light plugged on the mains, with frequency 50 Hz. The acquisition frame rate of the cameras produces an aliasing phenomenon on the video data at 4.17 Hz. In other words, the light intensity in the room is variable, at a frequency that can not be easily ignored, which does not comply with the initial hypothesis (1.12). However, the impact of this temporal dependence in the equations can be reduced by a normalization of the intensity of the images such as $y(x,y) = \frac{y(x,y)}{\bar{y}}$ where x and y are the horizontal and vertical indexes of the pixels in the image, $y(x,y)$ is the intensity of this pixel and $\bar{y}$ is the mean intensity on the entire image.

The depth field was estimated via the asymptotic observer (3.3) based on optical flow measures. The components of the optical flow were computed by the TV-L^1 variational method (see Subsection 2.1.4). The correction gain was tuned to $k = 100\ s.m^{-1}$. An example of image data is shown in Figure 4.18, and the depth estimate associated to that image at the same time $\bar{t}$ is shown in Figure 4.19. At that specific time $\bar{t} \approx 8$ s, the trolley already traveled once along

the track and is on its way back toward its starting point. Some specific estimates are extracted from the whole depth field (two tables, two chairs, a screen, two walls) and highlighted in black; they are compared to real measurements taken in the experimental room by a laser telemeter(in red): the estimate depth profile $\widehat{D}(\bar{t},\cdot)$ exhibits a strong correlation with these seven punctual reference values of $D(\bar{t},\cdot)$ since the relative error does not exceed 13%; the global appearance of the depth field looks very realistic.

Figure 4.18: The static scene as seen by the camera

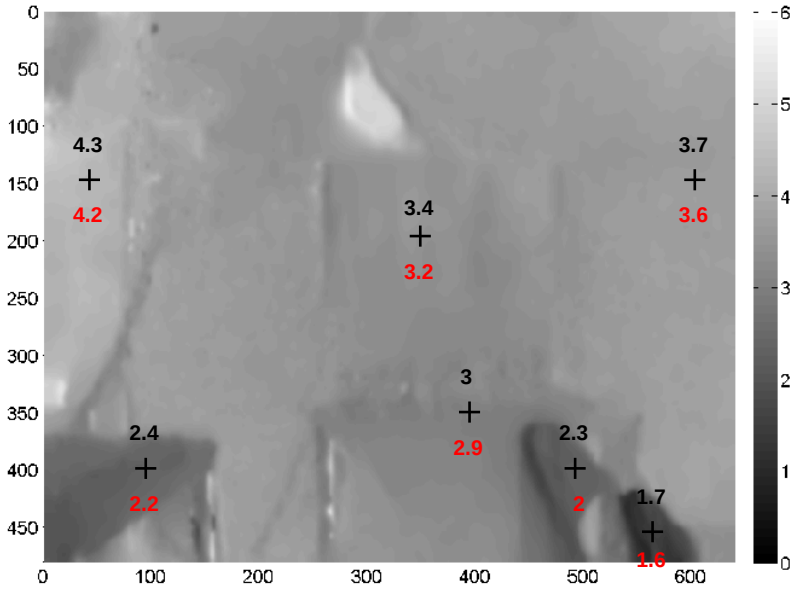


Figure 4.19: Estimation of the depth field associated with the image shown in Fig.4.18. Depth is associated to a gray level, whose scale in meters is on the right. The depth field was estimated via the asymptotic observer (3.3) based on optical flow measures. The components of the optical flow were computed by the TV-L^1 variational method. Some estimates are extracted from the entire field (in black) and compared to real measures (in red). The relative erreor does not exceed 13%.

Part II

From depth to motion

De la profondeur au mouvement

Introduction

The Kinect device has been a huge outbreak in the robotics and vision communities as it provides depth measurements registered at each pixel of a RGB image, at a relatively low cost: this output is named RGBD. Since its first release in November 2010, it has been extensively evaluated, in terms of accuracy [Gonzalez-Jorge et al., 2013, Khoshelham and Elberink, 2012], as well as calibration [Daniel Herrera et al., 2012] or possible applications [Smisek et al., 2011, Zhang, 2012]. The most obvious applications in the field of robotics are indoor modeling [Du et al., 2011, Henry et al., 2012], control [Stowers et al., 2011], obstacle avoidance [Correa et al., 2012]. But other applications have also raised interest: for example motion capture [Asteriadis et al., 2013, Calderita et al., 2013], surgery [Guo et al., 2013], training and learning [Guerrero and Uribe-Quevedo, 2013], gesture detection [Maisto et al., 2013], or human action recognition [Munaro et al., 2013]. Finally, the domain of 3D SLAM has literally exploded with new algorithms known as RGBD SLAM algorithms, such as [Endres et al., 2012, Audras et al., 2011, Huang et al., 2011] or Kinect Fusion [Izadi et al., 2011] and its Kintinuous extensions ([Whelan et al., 2012, Whelan et al., 2013a, Whelan et al., 2013b]).

The Kinect device made possible the work presented in the following part: our approach consist in the combined processing of dense intensity and depth fields (namely the RGBD data) for 6-DOF motion estimation.

Figure 4.20: The Kinect device, released by Microsoft in November 2010. It provides RGBD data, which enables motion estimation techniques based on combined processing of image and depth, as presented in this part.

Chapter 5

Estimation of biases on linear and angular velocities

Estimation de biais sur les vitesses linéaire et angulaire

Dans ce chapitre, nous étudions l'estimation de biais sur les vitesses linéaire et angulaire par l'exploitation combinée d'images RGBD et de données biaisées de vitesses linéaire et angulaire. Le système d'EDP lié au modèle de caméra sphérique décrit au Chapitre 1 est utilisé pour construire un observateur associé à une fonction de Lyapunov. On en déduit un système d'équations intégrales/aux dérivées partielles sur les champs et les biais estimés. La convergence de l'observateur est traitée dans le cadre d'hypothèses de régularité C^1 sur le mouvement de la caméra et la scène. Il est prouvé que les biais estimés convergent vers les biais réels si et seulement si la scène n'admet pas de symétrie de révolution. L'observateur peut être adapté à des capteurs plus réalistes pour lesquels les champs d'intensité lumineuse et de profondeur ne sont disponibles que sur une partie de la sphère $\mathbb{S}^2$. Ces résultats ont été présentés, partiellement dans [Zarrouati-Vissiere et al., 2012], et de manière plus complète dans [Zarrouati-Vissière et al., 2013].

In this chapter, the estimation of angular and linear velocity biases from the combined exploitation of RGBD images and biased linear and angular velocity inputs is studied. The PDE system associated to the spherical camera model described in Chapter 1 is used to design an observer based on a Lyapunov functional. It yields an integro/partial differential system for the estimated fields and biases. The observer convergence is investigated under C^1 regularity assumptions on the object motion and its scene. It relies on Ascoli-Arzela theorem and pre-compactness of the observer trajectories. It is proved that the estimated biases converge towards the true ones, if and only if, the scene admits no cylindrical symmetry. The observer design can be adapted to realistic sensors where brightness and depth data are only available on a subset of $\mathbb{S}^2$. These results were presented partially in [Zarrouati-Vissiere et al., 2012] and fully in [Zarrouati-Vissière et al., 2013].

5.1 Statement of the problem

We intend to study the situation, where linear and angular velocity are provided up to a slowly varying bias (the wind or an ocean current), which can be illustrated by [Bristeau et al., 2011]. Synchronized RGBD data are processed along with these measurements to estimate their biases, in order to improve the overall reconstructed trajectory. We recall from Chapter 1.3 that functions y and D (light intensity and depth respectively) obey to

$$\partial_t y = -\nabla y \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})),$$

$$\partial_t D = -\nabla D \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})) - \boldsymbol{v} \cdot \boldsymbol{\eta},$$

where $\partial_t y$ and $\partial_t D$ stand for partial derivatives of y and D with respect to t. The camera motion characterized by $(\boldsymbol{v}, \boldsymbol{\omega})$ is not known precisely. We assume here that sensor data provide $\boldsymbol{v_m}(t)$ and $\boldsymbol{\omega_m}(t)$ differing from true velocities by measurement biases, defined as constant errors:

$$\boldsymbol{v_m}(t) = \boldsymbol{v}(t) + \boldsymbol{v_b}, \quad \boldsymbol{\omega_m}(t) = \boldsymbol{\omega}(t) + \boldsymbol{\omega_b}, \tag{5.1}$$

where $\boldsymbol{v_b}, \boldsymbol{\omega_b} \in \mathbb{R}^3$ are constant:

$$\partial_t \boldsymbol{v_b} = 0, \quad \partial_t \boldsymbol{\omega_b} = 0. \tag{5.2}$$

Thus, the state equations of the system are

$$\begin{cases} \partial_t y = -\nabla y \cdot \boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \boldsymbol{\omega_b} + \frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{v_m} - \boldsymbol{v_b})), \\ \partial_t D = -\nabla D \cdot \boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \boldsymbol{\omega_b} + \frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{v_m} - \boldsymbol{v_b})) - \boldsymbol{\eta} \cdot (\boldsymbol{v_m} - \boldsymbol{v_b}), \\ \partial_t \boldsymbol{v_b} = 0, \\ \partial_t \boldsymbol{\omega_b} = 0. \end{cases} \tag{5.3}$$

Here $(\boldsymbol{v_m}, \boldsymbol{\omega_m})$ are considered as known inputs and (y, D) as the measured outputs. The state is $(y, D, \boldsymbol{v_b}, \boldsymbol{\omega_b})^T$. The goal is to estimate in real-time the parameters $(\boldsymbol{v_b}, \boldsymbol{\omega_b})$ from the known signals corresponding to sensors measuring y, D, $\boldsymbol{v_m}$ and $\boldsymbol{\omega_m}$. We assume

1. $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are in $C_b^1([0, +\infty), \mathcal{R}^3)$; $C(t)$ stays in a fixed compact subset K of Ω for all $t > 0$,
2. y_Σ is a C^1 function of $s \in \mathbb{S}^2$; the surface Σ is C^1

where $C_b^k([0, +\infty), X)$ is the set of functions whose time derivatives are bounded on X uniformly with respect to $t \in [0, +\infty)$ up to order k.

Thanks to the first assumption and (1.8), $C(t)$ and $q(t)$ belong to $C_b^2([0, +\infty))$. Thanks to the second assumption and (1.9), $\psi(t, \boldsymbol{\eta})$ and $\phi(t, s)$ belong to $C_b^1([0, +\infty) \times \mathbb{S}^2, \mathbb{S}^2)$. We deduce also from (1.11), (1.13) and the first assumption that

$$\begin{gathered} y, D \in C_b^1([0, +\infty) \times \mathbb{S}^2, \mathcal{R}) \text{ and} \\ D(t, \boldsymbol{\eta}) \geqslant D_* > 0\,, \forall (t, \boldsymbol{\eta}) \in [0, +\infty) \times \mathbb{S}^2\,. \end{gathered} \tag{5.4}$$

5.2 Observability of the reference system

5.2.1 Characterization of the observability

To prove the observability, we consider two state-trajectories

$$t \mapsto (y^1(t,\cdot), D^1(t,\cdot), \boldsymbol{v_b}^1(t), \boldsymbol{\omega_b}^1(t))^T \text{ and } t \mapsto (y^2(t,\cdot), D^2(t,\cdot), \boldsymbol{v_b}^2(t), \boldsymbol{\omega_b}^2(t))^T$$

and assume that they satisfy the same state equations (5.3) with the same inputs $t \mapsto (\boldsymbol{v_m}(t), \boldsymbol{\omega_m}(t))$ and outputs $t \mapsto (y(t,\cdot), D(t,\cdot))$. It immediately follows that $y^1 = y^2 = y$, $D^1 = D^2 = D$, $\boldsymbol{v_b}^1 = \boldsymbol{v_b}^2 + \boldsymbol{v_b}$ and $\boldsymbol{\omega_b}^1 = \boldsymbol{\omega_b}^2 + \boldsymbol{\omega_b}$ with $\partial_t \boldsymbol{v_b} = 0$ and $\partial_t \boldsymbol{\omega_b} = 0$. Then $t \mapsto (y(t,\cdot), D(t,\cdot), \boldsymbol{v_b}(t), \boldsymbol{\omega_b}(t))^T$ satisfy:

$$\begin{cases} \nabla y \cdot \boldsymbol{\eta} \times (\boldsymbol{\omega_b} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v_b}) = 0, \\ \nabla D \cdot \boldsymbol{\eta} \times (\boldsymbol{\omega_b} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v_b}) + \boldsymbol{\eta} \cdot \boldsymbol{v_b} = 0, \\ \partial_t \boldsymbol{v_b} = 0, \\ \partial_t \boldsymbol{\omega_b} = 0. \end{cases} \tag{5.5}$$

Observability means that (5.5) implies that $(\boldsymbol{\omega_b}, \boldsymbol{v_b}) = 0$.

Consider a solution $(y, D, \boldsymbol{\omega_b}, \boldsymbol{v_b})$ of (5.5). Take some fixed value $\bar{t}$ of t and set $\bar{y}(\cdot) = y(\bar{t}, \cdot)$ with $\bar{D}(\cdot) = D(\bar{t}, \cdot)$. Then $(\bar{y}, \bar{D})$ is a stationary solution of (1.16,1.17) when $(\boldsymbol{\omega}, \boldsymbol{v})$ is constant and equal to $(\boldsymbol{\omega_b}, \boldsymbol{v_b})$. Characterizing a scene, i.e., the surface Σ and its painting y_Σ, that admits a stationary solution of (1.16) and (1.17) with non zero constant values of $(\boldsymbol{\omega}, \boldsymbol{v})$, i.e. that is compatible with an effective motion producing stationary images y and D, is not obvious.

These considerations motivate the following terminology: for a given scene (Σ, y_Σ), a constant motion $(\boldsymbol{\omega}, \boldsymbol{v}) = (\boldsymbol{\omega_b}, \boldsymbol{v_b})$, such that a stationary solution (y, D) of (1.16,1.17) exists, i.e.,

$$\begin{cases} \nabla D \cdot \boldsymbol{\eta} \times (\boldsymbol{\omega_b} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v_b}) + \boldsymbol{\eta} \cdot \boldsymbol{v_b} = 0, \\ \nabla y \cdot \boldsymbol{\eta} \times (\boldsymbol{\omega_b} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v_b}) = 0, \end{cases} \tag{5.6}$$

is called a stationary motion inside (Σ, y_Σ). For any scene (Σ, y_Σ), the null motion $\boldsymbol{v_b} = \boldsymbol{\omega_b} = 0$ is always stationary. Apart from this trivial stationary motion, only very specific scenes (see Theorem 3) admit non trivial stationary motions $(\boldsymbol{v_b}, \boldsymbol{\omega_b})$. Thus observability of (5.3) with output (y, D) is equivalent to the following statement: the stationary motions $(\boldsymbol{v_b}, \boldsymbol{\omega_b})$ inside the scene (Σ, y_Σ) are all reduced to the trivial one $(0, 0)$.

5.2.2 Scenes with non trivial stationary motion

The following theorem characterizes the scenes admitting only trivial stationary motion, i.e., such that a stationary solution of (1.16,1.17) exists for some $(\boldsymbol{\omega}, \boldsymbol{v}) \neq 0$ constant.

Theorem 3. *Consider the assumptions given in Chapter 1.3 and Section 5.1. The scene (Σ, y_Σ) admits a non trivial stationary motion, if and only if, it admits a rotation axis.*

Proof of Theorem 3: Let us consider a non-zero motion $(\boldsymbol{v_b}, \boldsymbol{\omega_b})^T$. Under the assumption that the equations (5.6) have a solution $(y, D) \in C^1(\mathbb{S}^2, \mathbb{R}^2)$ we prove that (Σ, y_Σ) admits a rotation axis. Note that

$$\Sigma = \left\{ D(\boldsymbol{\eta})\boldsymbol{\eta}; \boldsymbol{\eta} \in \mathbb{S}^2 \right\},$$
$$y_\Sigma(s) = y(t, \boldsymbol{\eta}) \text{ with } \boldsymbol{\eta} = \phi(t, s) \text{(see Chapter 1.3)}.$$

Let us eliminate two trivial cases:

- $\boldsymbol{v_b} = 0, \boldsymbol{\omega_b} \neq 0$: the two first equations of (5.6) reduce to
$$\begin{cases} \nabla D \cdot (\boldsymbol{\eta} \times \boldsymbol{\omega_b}) = 0, \\ \nabla y \cdot (\boldsymbol{\eta} \times \boldsymbol{\omega_b}) = 0, \end{cases} \tag{5.7}$$
which characterizes a cylindrical symmetry for y and D, with same rotation axis defined by the direction of $\boldsymbol{\omega_b}$;
- $\boldsymbol{v_b} \neq 0, \boldsymbol{\omega_b} = 0$: let us replace $\boldsymbol{\eta}$ by $\boldsymbol{v_b}/\|\boldsymbol{v_b}\|$ in the first equation of (5.6): this yields $\|\boldsymbol{v_b}\| = 0$, which is absurd.

From now on, we consider only the case where $\boldsymbol{v_b}$ and $\boldsymbol{\omega_b}$ are both non-zero. Our strategy consists in finding a fixed vector $\mathbf{u}$ such that a translation of the origin of the camera frame in the direction $\mathbf{u}$ transforms the functions $D(\boldsymbol{\eta})$ and $y(\boldsymbol{\eta})$ in new functions $\widetilde{D}$ and $\widetilde{y}$ on the sphere $\mathbb{S}^2$ and those functions are invariant by rotation about the axis $\mathbb{R}\boldsymbol{\omega_b}$. This requires several technical steps.

Step 1: Let us prove that $\boldsymbol{v_b}$ and $\boldsymbol{\omega_b}$ are orthogonal. Starting from the first equation of (5.6), let us multiply it by $D^n(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b})$, where the choice of $n \in \mathbb{N}$ will follow, and integrate the result on $\mathbb{S}^2$:

$$I = \int_{\mathbb{S}^2} \Big(D^n \nabla D \cdot \boldsymbol{\eta} \times (\boldsymbol{\omega_b} + \frac{1}{D} \boldsymbol{\eta} \times \boldsymbol{v_b}) + D^n \boldsymbol{\eta} \cdot \boldsymbol{v_b} \Big) (\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) d\sigma_\eta = 0, \tag{5.8}$$

We have

$$I = \int_{\mathbb{S}^2} \Big[\frac{1}{n+1} \nabla D^{n+1} \cdot (\boldsymbol{\eta} \times \boldsymbol{\omega_b})(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) + \frac{1}{n} \nabla D^n \cdot (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{\omega_b}))(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) + D^n (\boldsymbol{\eta} \cdot \boldsymbol{v_b})(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) \Big] d\sigma_\eta.$$

The two first terms are integrated by parts:

$$I = \int_{\mathbb{S}^2} \Big[-\frac{1}{n+1} D^{n+1} \nabla \cdot ((\boldsymbol{\eta} \times \boldsymbol{\omega_b})(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b})) - \frac{1}{n} D^n \nabla \cdot ((\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v_b}))(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b})) + D^n (\boldsymbol{\eta} \cdot \boldsymbol{v_b})(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) \Big] d\sigma_\eta.$$

The derivatives are developed:

$$I = \int_{\mathbb{S}^2} \Big[-\frac{1}{n+1} D^{n+1} (\nabla \cdot (\boldsymbol{\eta} \times \boldsymbol{\omega_b})(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) + (\boldsymbol{\eta} \times \boldsymbol{\omega_b}) \cdot \nabla (\boldsymbol{\eta} \cdot \boldsymbol{\omega_b})) - \frac{1}{n} D^n \Big(\nabla \cdot (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v_b}))(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) + (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v_b})) \cdot \nabla (\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) \Big) + D^n (\boldsymbol{\eta} \cdot \boldsymbol{v_b})(\boldsymbol{\eta} \cdot \boldsymbol{\omega_b}) \Big] d\sigma_\eta.$$

Now, let us recall the following basic formulae of differential geometry on $\mathbb{S}^2$ (detailed and proven in Appendix A), where $\mathbf{P}$ is a constant vector:

$$\nabla(\boldsymbol{\eta}\cdot\mathbf{P}) = -\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\mathbf{P}), \tag{5.9}$$
$$\Delta(\boldsymbol{\eta}\cdot\mathbf{P}) = -2\boldsymbol{\eta}\cdot\mathbf{P}, \tag{5.10}$$
$$\nabla\cdot(\boldsymbol{\eta}\times\mathbf{P}) = 0. \tag{5.11}$$

Using the formulae (5.11) and (5.9) in the first term, and (5.10) and (5.9) in the second term yields:

$$\begin{aligned} I = \int_{\mathbb{S}^2}\Big[\frac{1}{n+1}D^{n+1}(\boldsymbol{\eta}\times\boldsymbol{\omega_b})\cdot(\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{\omega_b}))& \\ -\frac{1}{n}D^n((2\boldsymbol{\eta}\cdot\boldsymbol{v_b})(\boldsymbol{\eta}\cdot\boldsymbol{\omega_b})-(\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{v_b}))\cdot(\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{\omega_b})))& \\ +D^n(\boldsymbol{\eta}\cdot\boldsymbol{v_b})(\boldsymbol{\eta}\cdot\boldsymbol{\omega_b})\Big]d\sigma_\eta.& \end{aligned}$$

The first term is obviously zero. Then

$$(\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{v_b}))\cdot(\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{\omega_b})) = -(\boldsymbol{\eta}\cdot\boldsymbol{v_b})(\boldsymbol{\eta}\cdot\boldsymbol{\omega_b})+\boldsymbol{v_b}\cdot\boldsymbol{\omega_b}.$$

Thus

$$I = \int_{\mathbb{S}^2}\Big[-\frac{1}{n}D^n((3\boldsymbol{\eta}\cdot\boldsymbol{v_b})(\boldsymbol{\eta}\cdot\boldsymbol{\omega_b})-\boldsymbol{v_b}\cdot\boldsymbol{\omega_b})+D^n(\boldsymbol{\eta}\cdot\boldsymbol{v_b})(\boldsymbol{\eta}\cdot\boldsymbol{\omega_b})\Big]d\sigma_\eta.$$

Choosing $n=3$ yields

$$(\boldsymbol{v_b}\cdot\boldsymbol{\omega_b})\int_{\mathbb{S}^2}D^3d\sigma_\eta = 0. \tag{5.12}$$

Since $D>0$, we deduce that $\boldsymbol{v_b}\cdot\boldsymbol{\omega_b}=0$. This ends the proof of Step 1.

As $\boldsymbol{v_b}$ and $\boldsymbol{\omega_b}$ are orthogonal and both non-zeros, one can introduce a new vector, $l\mathbf{u}$, where $l>0$ is a length and $\mathbf{u}$ is a unit vector, by: $\boldsymbol{v_b}=\boldsymbol{\omega_b}\times l\mathbf{u}$. Let us also define $\mathbf{v}$ and $\mathbf{w}$ the unit vectors such that $\boldsymbol{v_b}=\|\boldsymbol{v_b}\|\mathbf{v}$ and $\boldsymbol{\omega_b}=\|\boldsymbol{\omega_b}\|\mathbf{w}$. Then, $(\mathbf{u},\mathbf{v},\mathbf{w})$ is an orthonormal and direct frame of $\mathbb{R}^3$. From now on, (5.6), multiplied by $lD/\|\boldsymbol{v_b}\|$ writes:

$$\begin{cases} \nabla D\cdot((D\boldsymbol{\eta}+l\mathbf{u})\times\mathbf{w})+lD\boldsymbol{\eta}\cdot\mathbf{v}=0,\\ \nabla y\cdot((D\boldsymbol{\eta}+l\mathbf{u})\times\mathbf{w})=0. \end{cases} \tag{5.13}$$

Step 2: Let us prove that $D(\boldsymbol{\eta})\boldsymbol{\eta}+l\mathbf{u}\neq 0, \forall\boldsymbol{\eta}\in\mathbb{S}^2$. It is sufficient to prove that $D(-\mathbf{u})\neq l$. Let $\boldsymbol{\eta}_\epsilon := \frac{-\mathbf{u}+\epsilon\mathbf{v}}{\sqrt{1+\epsilon^2}} = -\mathbf{u}+\epsilon\mathbf{v}+o(\epsilon)$. We have

$$\begin{aligned} D(\boldsymbol{\eta}_\epsilon)\boldsymbol{\eta}_\epsilon &= (D(-\mathbf{u})+\epsilon\,\nabla D|_{-\mathbf{u}}\cdot\mathbf{v}+o(\epsilon))\,(-\mathbf{u}+\epsilon\mathbf{v}+o(\epsilon))\\ &= -D(-\mathbf{u})\mathbf{u}+\epsilon\,(D(-\mathbf{u})\mathbf{v}-(\nabla D|_{-\mathbf{u}}\cdot\mathbf{v})\,\mathbf{u})+o(\epsilon) \end{aligned}$$

Thus, (5.13) evaluated at $\boldsymbol{\eta}_\epsilon$ yields

$$\nabla D|_{\boldsymbol{\eta}_\epsilon}\cdot[[(l-D(-\mathbf{u}))\mathbf{u}+\epsilon\,(D(-\mathbf{u})\mathbf{v}-(\nabla D|_{-\mathbf{u}}\cdot\mathbf{v})\,\mathbf{u})]\times\mathbf{w}]+\epsilon lD(-\mathbf{u})=o(\epsilon),$$

or

$$\nabla D|_{\boldsymbol{\eta}_\epsilon} \cdot [(D(-\mathbf{u}) - l)\mathbf{v} + \epsilon\,(D(-\mathbf{u})\mathbf{u} + (\nabla D|_{-\mathbf{u}} \cdot \mathbf{v})\,\mathbf{v})] + \epsilon l D(-\mathbf{u}) = o(\epsilon). \tag{5.14}$$

Equation (5.13) evaluated at $\boldsymbol{\eta} = -\mathbf{u}$ and substracted to (5.14) yields:

$$\begin{aligned}(D(-\mathbf{u}) - l)\left(\nabla D|_{\boldsymbol{\eta}_\epsilon} - \nabla D|_{-\mathbf{u}}\right) \cdot \mathbf{v} \\ + \epsilon\left[D(-\mathbf{u})\,\nabla D|_{\boldsymbol{\eta}_\epsilon} \cdot \mathbf{u} + (\nabla D|_{-\mathbf{u}} \cdot \mathbf{v})\left(\nabla D|_{\boldsymbol{\eta}_\epsilon} \cdot \mathbf{v}\right) + l D(-\mathbf{u})\right] = o(\epsilon).\end{aligned} \tag{5.15}$$

Let us define

$$\begin{array}{rccc} F: & \mathbb{R} & \to & \mathbb{R} \\ & \epsilon & \mapsto & D(-\mathbf{u})\,\nabla D|_{\boldsymbol{\eta}_\epsilon} \cdot \mathbf{u} + (\nabla D|_{-\mathbf{u}} \cdot \mathbf{v})\left(\nabla D|_{\boldsymbol{\eta}_\epsilon} \cdot \mathbf{v}\right) + l D(-\mathbf{u}). \end{array}$$

Assuming that $(D(-\mathbf{u}) - l) = 0$ implies that $F(\epsilon) = o(1)$ as $\epsilon \to 0$. Since F is a C^0 function of ϵ, it implies $F(0) = 0$, i.e.

$$(\nabla D|_{-\mathbf{u}} \cdot \mathbf{v})^2 + l D(-\mathbf{u}) = 0$$

which is absurd since $l > 0$. This concludes the Step 2.

Step 3: Let us prove that $f(\boldsymbol{\eta}) := \|D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}\|$ (respectively $y(\boldsymbol{\eta})$) satisfies $df(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta} = 0$ (respectively $dy(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta} = 0$) where $\delta\boldsymbol{\eta} := (D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}) \times \mathbf{w} + l(\boldsymbol{\eta} \cdot \mathbf{v})\boldsymbol{\eta}$, $\forall \boldsymbol{\eta} \in \mathbb{S}^2$. We have

$$f(\boldsymbol{\eta}) = g \circ k(\boldsymbol{\eta}) \quad \text{where} \quad g(\boldsymbol{x}) := \|\boldsymbol{x}\| \quad \text{and} \quad k(\boldsymbol{\eta}) := D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}.$$

Thanks to the Step 2, $f \in C^1(\mathbb{S}^2)$ and

$$df(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta} = dg[k(\boldsymbol{\eta})] \cdot \left(dk(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta}\right).$$

We have

$$\begin{aligned} dk(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta} &= (\nabla D(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta})\,\boldsymbol{\eta} + D(\boldsymbol{\eta})\delta\boldsymbol{\eta} \\ &= (\nabla D(\boldsymbol{\eta}) \cdot ([D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}] \times \mathbf{w} + l(\boldsymbol{\eta} \cdot \mathbf{v})\boldsymbol{\eta}))\,\boldsymbol{\eta} \\ &\quad + D(\boldsymbol{\eta})\,([D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}] \times \mathbf{w} + l(\boldsymbol{\eta} \cdot \mathbf{v})\boldsymbol{\eta}) \\ &= (\nabla D(\boldsymbol{\eta}) \cdot ([D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}] \times \mathbf{w}))\,\boldsymbol{\eta} + D(\boldsymbol{\eta})\,([D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}] \times \mathbf{w} + l(\boldsymbol{\eta} \cdot \mathbf{v})\boldsymbol{\eta}) \end{aligned}$$

because $\nabla D(\boldsymbol{\eta}) \perp \boldsymbol{\eta}$. Thanks to (5.13), we obtain

$$dk(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta} = D(\boldsymbol{\eta})[D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}] \times \mathbf{w}.$$

Thus

$$df(\boldsymbol{\eta}) \cdot \delta\boldsymbol{\eta} = \left\langle \frac{D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}}{\|D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}\|}, D(\boldsymbol{\eta})[D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}] \times \mathbf{w} \right\rangle = 0.$$

Similarly,

$$\begin{aligned} dy(\boldsymbol{\eta})\cdot\delta\boldsymbol{\eta} &= \nabla y\cdot\delta\boldsymbol{\eta}\\ &= \nabla y\cdot((D(\boldsymbol{\eta})\boldsymbol{\eta}+l\mathbf{u})\times\mathbf{w}+l(\boldsymbol{\eta}\cdot\mathbf{v})\boldsymbol{\eta})\\ &= \nabla y\cdot((D(\boldsymbol{\eta})\boldsymbol{\eta}+l\mathbf{u})\times\mathbf{w}) \text{ because } \nabla y(\boldsymbol{\eta})\perp\boldsymbol{\eta}\\ &= 0 \text{ according to (5.13).} \end{aligned}$$

This ends the proof of Step 3.

Step 4: Let us prove that, when $\boldsymbol{\eta}=\boldsymbol{\eta}(\tau)$ satisfies $\partial_\tau\boldsymbol{\eta}=[D(\boldsymbol{\eta})\boldsymbol{\eta}+l\mathbf{u}]\times\mathbf{w}+l(\boldsymbol{\eta}\cdot\mathbf{v})\boldsymbol{\eta}$, then $\widetilde{\boldsymbol{\eta}}:=[D(\boldsymbol{\eta})\boldsymbol{\eta}+l\mathbf{u}]/\|[D(\boldsymbol{\eta})\boldsymbol{\eta}+l\mathbf{u}]\|$ is a periodic trajectory on $\mathbb{S}^2$, describing a circle perpendicular to the axis $\mathbb{R}\mathbf{w}$. Let $\boldsymbol{\eta}_0\in\mathbb{S}^2$. We consider the characteristic (see [LeVeque, 1992, Serre, 1999] for a description of the method of characteristics)

$$\begin{cases} \frac{d\boldsymbol{\eta}}{d\tau}(\tau)=\Big(D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}\Big)\times\mathbf{w}+l\Big(\boldsymbol{\eta}(\tau)\cdot\mathbf{v}\Big)\boldsymbol{\eta}(\tau),\\ \boldsymbol{\eta}(0)=\boldsymbol{\eta}_0. \end{cases}$$

The right hand side is defined by a C^1 function of $\boldsymbol{\eta}$ thus the Cauchy-Lipschitz theorem guarantees the existence and uniqueness of a maximal solution. Moreover, no explosion is possible in finite time (the solution $\boldsymbol{\eta}(\tau)$ lives on $\mathbb{S}^2$) thus the maximal solution is defined for every $\tau\in\mathbb{R}$. From Step 3, we deduce that the quantity $\tau\mapsto\|D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}\|$ is constant along the characteristics:

$$\|D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}\|\equiv\widetilde{D}:=\|D[\boldsymbol{\eta}_0]\boldsymbol{\eta}_0+l\mathbf{u}\|,\quad\forall\tau\in\mathbb{R}.$$

Now let us consider

$$\widetilde{\boldsymbol{\eta}}(\tau):=\frac{D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}}{\|D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}\|}=\frac{D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}}{\widetilde{D}}.$$

Then, $\widetilde{\boldsymbol{\eta}}(\tau)$ is a C^1-function and

$$\begin{aligned} \widetilde{D}\frac{d\widetilde{\boldsymbol{\eta}}}{d\tau} &= \Big(\nabla D[\boldsymbol{\eta}(\tau)]\cdot\frac{d\boldsymbol{\eta}}{d\tau}\Big)\boldsymbol{\eta}(\tau)+D[\boldsymbol{\eta}(\tau)]\frac{d\boldsymbol{\eta}}{d\tau}\\ &= \Big(\nabla D[\boldsymbol{\eta}(\tau)]\cdot\Big(D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}\Big)\times\mathbf{w}\Big)\boldsymbol{\eta}(\tau)\\ &\quad +D[\boldsymbol{\eta}(\tau)]\Big(\Big(D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}\Big)\times\mathbf{w}+l\Big(\boldsymbol{\eta}(\tau)\cdot\mathbf{v}\Big)\boldsymbol{\eta}(\tau)\Big). \end{aligned}$$

Here, we have used the relation $\nabla D[\boldsymbol{\eta}(\tau)]\cdot\boldsymbol{\eta}(\tau)=0$ to get the last equality. Thanks to (5.13), we obtain

$$\widetilde{D}\frac{d\widetilde{\boldsymbol{\eta}}}{d\tau}=D[\boldsymbol{\eta}(\tau)]\Big(D[\boldsymbol{\eta}(\tau)]\boldsymbol{\eta}(\tau)+l\mathbf{u}\Big)\times\mathbf{w}=D[\boldsymbol{\eta}(\tau)]\widetilde{D}\widetilde{\boldsymbol{\eta}}(\tau)\times\mathbf{w}.$$

Thus, $\widetilde{\boldsymbol{\eta}}$ turns about the axis $\mathbb{R}\mathbf{w}$ as τ varies. Note that the speed $D[\boldsymbol{\eta}(\tau)]$ is bounded from below by a positive constant, thus the trajectory of $\widetilde{\boldsymbol{\eta}}$ is the (complete) circle on $\mathbb{S}^2$, perpendicular to the axis $\mathbb{R}\mathbf{w}$ and containing $\widetilde{\boldsymbol{\eta}}_0:=[D(\boldsymbol{\eta}_0)\boldsymbol{\eta}_0+l\mathbf{u}]/\widetilde{D}$.

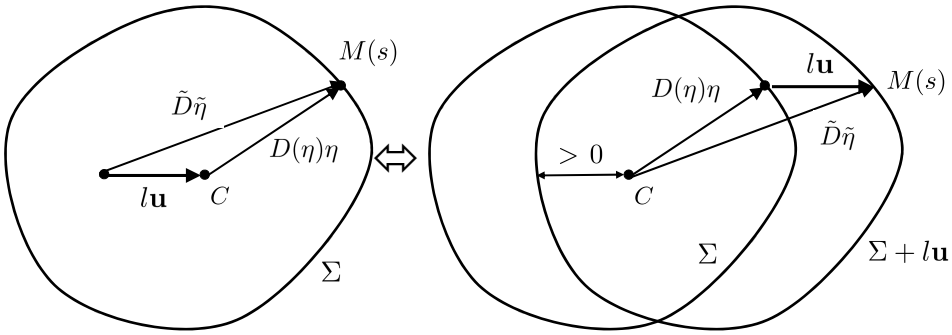


Figure 5.1: The vectors $D(\boldsymbol{\eta})\boldsymbol{\eta}$ taken from the camera center C and $\widetilde{D}\widetilde{\boldsymbol{\eta}}$ taken from the camera center translated by $-l\mathbf{u}$ arrive at the same point $M(s)$. It is equivalent to see this as a translation of the domain Σ by $l\mathbf{u}$.

Step 5: Let us prove that the map

$$\begin{array}{ccc} \mathbb{S}^2 & \to & \mathbb{S}^2 \\ \boldsymbol{\eta}_0 & \mapsto & \widetilde{\boldsymbol{\eta}}_0 := \frac{D(\boldsymbol{\eta}_0)\boldsymbol{\eta}_0 + l\mathbf{u}}{\|[D(\boldsymbol{\eta}_0)\boldsymbol{\eta}_0 + l\mathbf{u}]\|} \end{array}$$

is surjective. This step leans on the statement that when the convex set defined by the surface Σ is translated by $l\mathbf{u}$, the center of the camera frame stays inside the convex set $(\Sigma + l\mathbf{u})$. This statement is ensured by the fact that $D(-\mathbf{u}) > l$ (or equivalently, that $\widetilde{D}(-\mathbf{u}) > 0$, see Figure 5.2.2).

Thus, for any $\widetilde{\boldsymbol{\eta}}_0 \in \mathbb{S}^2$, $\exists M$ a point of the surface $\Sigma + l\mathbf{u}$ and $\exists \lambda > 0$ such that $\lambda\widetilde{\boldsymbol{\eta}}_0 = \overrightarrow{CM} = D(\boldsymbol{\eta}_0)\boldsymbol{\eta}_0 + l\mathbf{u}$ where $\boldsymbol{\eta}_0 \in \mathbb{S}^2$ (by definition of $\Sigma + l\mathbf{u}$). Then $\lambda = \|D(\boldsymbol{\eta}_0)\boldsymbol{\eta}_0 + l\mathbf{u}\|$ since $\widetilde{\boldsymbol{\eta}}_0$ is a unit vector, and we have shown that $\exists \boldsymbol{\eta}_0 \in \mathbb{S}^2$ such that $\widetilde{\boldsymbol{\eta}}_0 = \frac{D(\boldsymbol{\eta}_0)\boldsymbol{\eta}_0 + l\mathbf{u}}{\|D(\boldsymbol{\eta}_0)\boldsymbol{\eta}_0 + l\mathbf{u}\|}$. Thus the map $\boldsymbol{\eta}_0 \mapsto \widetilde{\boldsymbol{\eta}}_0$ is surjective.

Now, let us prove that $D(-\mathbf{u}) > l$. Let us assume that $D(-\mathbf{u}) \leqslant l$. Then, the function $\boldsymbol{\eta} \mapsto D(\boldsymbol{\eta}) + l\mathbf{u} \cdot \boldsymbol{\eta}$ is positive at $\boldsymbol{\eta} = \mathbf{u}$ and non positive at $\boldsymbol{\eta} = -\mathbf{u}$, thus (intermediate values theorem) there exists $\boldsymbol{\eta}^*$ such that $\boldsymbol{\eta}^* \cdot \mathbf{w} = 0$ and $D(\boldsymbol{\eta}^*) + l\mathbf{u} \cdot \boldsymbol{\eta}^* = 0$. Then, the vector $D(\boldsymbol{\eta}^*)\boldsymbol{\eta}^* + l\mathbf{u}$ is orthogonal to $\mathbf{w}$ (because $\mathbf{u}$ and $\boldsymbol{\eta}^*$ are) and orthogonal to $\boldsymbol{\eta}^*$, thus

$$D(\boldsymbol{\eta}^*)\boldsymbol{\eta}^* + l\mathbf{u} = \Big(\Big(D(\boldsymbol{\eta}^*)\boldsymbol{\eta}^* + l\mathbf{u}\Big) \cdot (\mathbf{w} \times \boldsymbol{\eta}^*)\Big)\, \mathbf{w} \times \boldsymbol{\eta}^*.$$

Note that $\mathbf{w} \times \boldsymbol{\eta}^*$ is unitary. Moreover,

$$\begin{aligned} \Big(D(\boldsymbol{\eta}^*)\boldsymbol{\eta}^* + l\mathbf{u}\Big) \cdot (\mathbf{w} \times \boldsymbol{\eta}^*) &= l\mathbf{u}.(\mathbf{w} \times \boldsymbol{\eta}^*) \text{ because } \boldsymbol{\eta}^* \perp \mathbf{w} \times \boldsymbol{\eta}^* \\ &= l\boldsymbol{\eta}^* \cdot (\mathbf{u} \times \mathbf{w}) \\ &= -l\boldsymbol{\eta}^* \cdot \mathbf{v} \end{aligned}$$

thus

$$D(\boldsymbol{\eta}^*)\boldsymbol{\eta}^* + l\mathbf{u} = l(\boldsymbol{\eta}^* \cdot \mathbf{v})\boldsymbol{\eta}^* \times \mathbf{w}.$$

Incorporating this relation in (5.13), we get

$$l(\boldsymbol{\eta}^* \cdot \mathbf{v})\nabla D(\boldsymbol{\eta}^*) \cdot \Big((\boldsymbol{\eta}^* \times \mathbf{w}) \times \mathbf{w}\Big) + D(\boldsymbol{\eta}^*)l\boldsymbol{\eta}^* \cdot \mathbf{v} = 0.$$

Moreover,

$$\Big(\boldsymbol{\eta}^* \times \mathbf{w}\Big) \times \mathbf{w} = \boldsymbol{\eta}^* - (\mathbf{w} \cdot \boldsymbol{\eta}^*)\mathbf{w} = \boldsymbol{\eta}^*$$

is orthogonal to $\nabla D(\boldsymbol{\eta}^*)$ thus

$$lD(\boldsymbol{\eta}^*)\boldsymbol{\eta}^* \cdot \mathbf{v} = 0.$$

Therefore, $\boldsymbol{\eta}^* \cdot \mathbf{w} = \boldsymbol{\eta}^* \cdot \mathbf{v} = 0$, i.e. $\boldsymbol{\eta}^* = \pm\mathbf{u}$, which is impossible because $D(\mathbf{u}) > 0$ and $D(-\mathbf{u}) \neq l$ (see Step 2). This proves that $D(-\mathbf{u}) > l$ and ends the Step 5.

Step 6: Conclusion. The origin of the camera frame is translated by the vector $-l\mathbf{u}$. We denote by $\widetilde{\boldsymbol{\eta}} \in \mathbb{S}^2$ the space variable in this new system of coordinates. Then, to any pair $(\boldsymbol{\eta}, D(\boldsymbol{\eta}))$ in the first system of coordinates corresponds a unique pair $(\widetilde{\boldsymbol{\eta}}, \widetilde{D})$ in the new system of coordinates:

$$\widetilde{\boldsymbol{\eta}} := \frac{D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}}{\|D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}\|}, \qquad \widetilde{D} := \|D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}\|,$$

and they describe the same point $M(s)$ on Σ:

$$M(s) = D(\boldsymbol{\eta})\boldsymbol{\eta} = -l\mathbf{u} + \widetilde{D}\widetilde{\boldsymbol{\eta}}.$$

In addition, $\widetilde{y}$ is defined so that for a given point $M(s)$ on Σ, the light intensity is the same in both systems of coordinates:

$$\begin{aligned} y_\Sigma(s) &= y(t, \boldsymbol{\eta}) \text{ where } \boldsymbol{\eta} = \phi(t, s) \text{ in the first system of coordinates} \\ &= \widetilde{y}(t, \widetilde{\boldsymbol{\eta}}) \text{ where } \widetilde{\boldsymbol{\eta}} = \frac{D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}}{\|D(\boldsymbol{\eta})\boldsymbol{\eta} + l\mathbf{u}\|} \text{ in the new system of coordinates.} \end{aligned}$$

Steps 3 and 4 of this proof show that the quantities $\widetilde{D}$ and $\widetilde{y}$ are constant when $\widetilde{\boldsymbol{\eta}}$ covers circles on $\mathbb{S}^2$ orthogonal to the axis $\mathbb{R}\mathbf{w}$. Moreover, Step 5 ensures that $\widetilde{\boldsymbol{\eta}}$ covers any such circle. Therefore, $\Sigma = \{\widetilde{D}(\widetilde{\boldsymbol{\eta}})\widetilde{\boldsymbol{\eta}}; \widetilde{\boldsymbol{\eta}} \in \mathbb{S}^2\}$ (in the new system of coordinates) and $y_\Sigma(s)$ have rotation axis, with direction $\mathbf{w}$. □

5.3 The asymptotic observer

5.3.1 A Lyapunov based observer

We propose the following observer for $\boldsymbol{\omega_b}$ and $\boldsymbol{v_b}$, inspired by the one proposed in [Bonnabel and Rouchon, 2009]:

$$\left\{\begin{array}{ll}
\partial_t \widehat{y} = & -\nabla\widehat{y}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m}-\widehat{\boldsymbol{\omega_b}}+\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m}-\widehat{\boldsymbol{v_b}})))+k_y(y-\widehat{y}),\\
\partial_t \widehat{D} = & -\nabla\widehat{D}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m}-\widehat{\boldsymbol{\omega_b}}+\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m}-\widehat{\boldsymbol{v_b}})))-(\boldsymbol{v_m}-\widehat{\boldsymbol{v_b}})\cdot\boldsymbol{\eta}+k_D(D-\widehat{D}),\\
\frac{d\widehat{\boldsymbol{\omega_b}}}{dt} = & -k_\omega\int_{\mathbb{S}^2}\Big(\lambda_y(\widehat{y}-y)(\nabla\widehat{y}\times\boldsymbol{\eta})+\lambda_D(\widehat{D}-D)(\nabla\widehat{D}\times\boldsymbol{\eta})\Big)d\sigma_\eta,\\
\frac{d\widehat{\boldsymbol{v_b}}}{dt} = & -k_v\int_{\mathbb{S}^2}\big(\lambda_D(\widehat{D}-D)\boldsymbol{\eta}+\lambda_y(\widehat{y}-y)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{y}))\\
& \qquad +\lambda_D(\widehat{D}-D)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{D}))\big)\,d\sigma_\eta,
\end{array}\right. \tag{5.16}$$

where k_y, k_D, k_ω, k_v, λ_y, $\lambda_D > 0$ are constant gains. The choice of this observer is motivated by the following Lyapunov function candidate

$$V := \frac{1}{2}\int_{\mathbb{S}^2}\left(\lambda_y\widetilde{y}^2+\lambda_D\widetilde{D}^2\right)d\sigma_\eta+\frac{\|\widetilde{\boldsymbol{\omega_b}}\|^2}{2k_\omega}+\frac{\|\widetilde{\boldsymbol{v_b}}\|^2}{2k_v} \tag{5.17}$$

with

$$\widetilde{y} := \widehat{y}-y,\quad \widetilde{D} := \widehat{D}-D,\quad \widetilde{\boldsymbol{\omega_b}} := \widehat{\boldsymbol{\omega_b}}-\boldsymbol{\omega_b} \text{ and } \widetilde{\boldsymbol{v_b}} := \widehat{\boldsymbol{v_b}}-\boldsymbol{v_b}. \tag{5.18}$$

Let us prove that V decreases along the trajectories of (5.16) for appropriate choices of k_y and k_D.. Let us consider any scalar field $h(t,\boldsymbol{\eta})$ defined on $\mathbb{S}^2$ and its integral $\mathcal{H}$ on the unit sphere:

$$\mathcal{H} = \int_{\mathbb{S}^2} h(t,\boldsymbol{\eta})^2 d\sigma_\eta = \int_{\mathbb{S}^2} h(t,q^*\boldsymbol{\eta}q)^2 d\sigma_\eta$$

since $\boldsymbol{\eta}\mapsto q^*\boldsymbol{\eta}q$ is an isometry on $\mathbb{S}^2$. For $\boldsymbol{\eta}$ a constant vector of the earth-fixed frame, $q^*\boldsymbol{\eta}q$ is the same vector, expressed in the camera frame. Thus, $h(t,q^*\boldsymbol{\eta}q)$ is the value of h corresponding to a specific object $M(s)$ of the scene. This yields

$$\begin{aligned}
\tfrac{d}{dt}\mathcal{H} &= \int_{\mathbb{S}^2}\tfrac{d}{dt}(h(t,q^*\boldsymbol{\eta}q)^2)d\sigma_\eta\\
&= 2\int_{\mathbb{S}^2} h(t,q^*\boldsymbol{\eta}q)\ \tfrac{d}{dt}h(t,q^*\boldsymbol{\eta}q)\big|_s\,d\sigma_\eta\\
&= 2\int_{\mathbb{S}^2} h(t,q^*\boldsymbol{\eta}q)\,(\partial_t h+\nabla h\cdot(\boldsymbol{\eta}\times\boldsymbol{\omega}))\,d\sigma_\eta.
\end{aligned}$$

One can apply this calculation rule to the scalar fields $\widetilde{y}$ and $\widetilde{D}$. Then, equations (1.16) and (5.16) yield

$$\partial_t\widetilde{y}+\nabla\widetilde{y}\cdot(\boldsymbol{\eta}\times\boldsymbol{\omega}) = \nabla\widehat{y}\cdot(\boldsymbol{\eta}\times(\widetilde{\boldsymbol{\omega_b}}+\frac{1}{D}\boldsymbol{\eta}\times\widetilde{\boldsymbol{v_b}}))-\nabla\widetilde{y}\cdot(\boldsymbol{\eta}\times(\frac{1}{D}\boldsymbol{\eta}\times\boldsymbol{v}))-k_y\widetilde{y}. \tag{5.19}$$

Equations (1.17) and (5.16) yield

$$\partial_t\widetilde{D}+\nabla\widetilde{D}\cdot(\boldsymbol{\eta}\times\boldsymbol{\omega}) = \nabla\widehat{D}\cdot(\boldsymbol{\eta}\times(\widetilde{\boldsymbol{\omega_b}}+\frac{1}{D}\boldsymbol{\eta}\times\widetilde{\boldsymbol{v_b}}))+\boldsymbol{\eta}\cdot\widetilde{\boldsymbol{v_b}}-\nabla\widetilde{D}\cdot(\boldsymbol{\eta}\times(\frac{1}{D}\boldsymbol{\eta}\times\boldsymbol{v}))-k_D\widetilde{D}. \tag{5.20}$$

Now, using $\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})=(\mathbf{a}\times\mathbf{b})\cdot\mathbf{c}$ and the expressions of $\frac{d\widetilde{\boldsymbol{\omega_b}}}{dt}$ and $\frac{d\widetilde{\boldsymbol{v_b}}}{dt}$, one gets

$$\dot{V}=-\int_{\mathbb{S}^2}\left(k_y\lambda_y\widetilde{y}^2+k_D\lambda_D\widetilde{D}^2\right)d\sigma_\eta-\int_{\mathbb{S}^2}\left(\lambda_y\frac{\nabla\widetilde{y}^2}{2}+\lambda_D\frac{\nabla\widetilde{D}^2}{2}\right)\cdot W(t,\boldsymbol{\eta})d\sigma_\eta. \tag{5.21}$$

where $(\mathbb{R},\mathbb{S}^2)\ni(t,\boldsymbol{\eta})\mapsto W(t,\boldsymbol{\eta})=\boldsymbol{\eta}\times(\frac{1}{D}\boldsymbol{\eta}\times v)\in\mathbb{R}^3$. Integration by parts of the second term of this expression yields

$$\dot{V}=-\int_{\mathbb{S}^2}\left(\lambda_y\left(k_y-\frac{\nabla\cdot W(t,\boldsymbol{\eta})}{2}\right)\widetilde{y}^2+\lambda_D\left(k_D-\frac{\nabla\cdot W(t,\boldsymbol{\eta})}{2}\right)\widetilde{D}^2\right)d\sigma_\eta\,. \tag{5.22}$$

Thus V is a Lyapunov function for trajectories of (5.16) when k_y,k_D are large enough (see Proposition 1 below).

The convergence analysis relies on an adaptation to this infinite dimensional setting of the Lasalle invariance principle. Precisely, we prove that any adherence value of $(\widetilde{y},\widetilde{D},\widetilde{\boldsymbol{\omega_b}},\widetilde{\boldsymbol{v_b}})$ as $[t\to+\infty]$ satisfies geometric equations equivalent to (5.6) (see Theorem 4). This proof requires the following successive steps.

1. The Lyapunov function V decreases along the trajectories of (5.16), for appropriate choices of k_y and k_D (see Proposition 1).
2. The non-linear partial differential equations (5.16) define a well posed Cauchy problem for every $t\geq 0$ (see Proposition 2).
3. The functions $\nabla\widehat{y}$ and $\nabla\widehat{D}$ are uniformly bounded with respect to $(t,\boldsymbol{\eta})\in[0,+\infty)\times\mathbb{S}^2$ (see Proposition 3).
4. The estimated state $(\widehat{y},\widehat{D},\widehat{\boldsymbol{\omega_b}},\widehat{\boldsymbol{v_b}})$ depends continuously on $(y,D,\boldsymbol{\omega},\boldsymbol{v})$ (see Proposition 4).

5.3.2 Decrease of the Lyapunov function

In the next statement are given sufficient conditions for the choice of correction gains k_y and k_D to ensure the decrease of the Lyapunov function V.

Proposition 1. *The Lyapunov function V decreases along the trajectories of (5.16) under the assumption that $k_y,k_D>\frac{L}{2}$ where L is defined as*

$$L:=\left\|\frac{1}{D(t,\boldsymbol{\eta})}\left(2\boldsymbol{\eta}\cdot\boldsymbol{v}(t)-\frac{\nabla D(t,\boldsymbol{\eta})}{D(t,\boldsymbol{\eta})}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{v}(t)))\right)\right\|_{L^\infty((0,+\infty)\times\mathbb{S}^2)}.$$

Proof of Proposition 1: Note that L is well defined thanks to (5.4) and assumption $\boldsymbol{v}\in C^1_b((0,+\infty),\mathbb{R}^3)$ (see Section 5.1). Moreover, $L=\|\nabla\cdot W(t,\boldsymbol{\eta})\|_{L^\infty((0,+\infty)\times\mathbb{S}^2)}$ (see formulae (5.9) and (5.10)). From (5.22), we deduce that

$$\dot{V}\leq-\int_{\mathbb{S}^2}\left(\lambda_y\left(k_y-\frac{L}{2}\right)\widetilde{y}^2+\lambda_D\left(k_D-\frac{L}{2}\right)\widetilde{D}^2\right)d\sigma_\eta:=-f(t) \tag{5.23}$$

which ends the proof of Proposition 1. □

From now on, we assume that k_y and k_D satisfy the assumption of Proposition 1.

5.3.3 Existence and uniqueness

Existence and uniqueness of solutions of (5.16) is given by the next statement. For the simplicity of notations, from now on, we assume that $k_v = k_\omega = 1$.

Proposition 2. *Let $R > 0$. There exists a constant*

$$k_* = k_*\Big(R, \|(\boldsymbol{v_m}, \boldsymbol{\omega_m})\|_{L^\infty(0,+\infty)}, D_*, \|(y, D)\|_{C^1([0,+\infty)\times\mathbb{S}^2)}\Big) > 0$$

such that, for every $\widehat{\boldsymbol{\omega_b}}^0, \widehat{\boldsymbol{v_b}}^0 \in \mathbb{R}^3$ with $\|(\widehat{\boldsymbol{\omega_b}}^0 - \boldsymbol{\omega_b}, \widehat{\boldsymbol{v_b}}^0 - \boldsymbol{v_b})\| \leqslant R$, $k_y, k_D > k_*$, there exists a unique solution $(\widehat{y}, \widehat{D}, \widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}}) \in C^1([0,+\infty)\times\mathbb{S}^2, \mathbb{R})^2 \times C^1([0,+\infty), \mathbb{R}^3)^2$ of the Cauchy-problem

$$\begin{cases}
\partial_t \widehat{y} & = -\nabla\widehat{y}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}))) + k_y(y - \widehat{y}),\\
\partial_t \widehat{D} & = -\nabla\widehat{D}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}))) - (\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}})\cdot\boldsymbol{\eta}\\
& \quad + k_D(D - \widehat{D}),\\
\frac{d\widehat{\boldsymbol{\omega_b}}}{dt} & = -\int_{\mathbb{S}^2}\big(\lambda_y(\widehat{y} - y)(\nabla\widehat{y}\times\boldsymbol{\eta}) + \lambda_D(\widehat{D} - D)(\nabla\widehat{D}\times\boldsymbol{\eta})\big)\,d\sigma_\eta,\\
\frac{d\widehat{\boldsymbol{v_b}}}{dt} & = -\int_{\mathbb{S}^2}\big(\lambda_D(\widehat{D} - D)\boldsymbol{\eta} + \lambda_y(\widehat{y} - y)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{y})\\
& \quad + \lambda_D(\widehat{D} - D)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{D}))\big)\,d\sigma_\eta,\\
\widehat{y}(0,\boldsymbol{\eta}) & = y(0,\boldsymbol{\eta}),\\
\widehat{D}(0,\boldsymbol{\eta}) & = D(0,\boldsymbol{\eta}),\\
\widehat{\boldsymbol{\omega_b}}(0) & = \widehat{\boldsymbol{\omega_b}}^0,\\
\widehat{\boldsymbol{v_b}}(0) & = \widehat{\boldsymbol{v_b}}^0.
\end{cases} \tag{5.24}$$

The proof of Proposition 2 is technical, thus we postpone it to Section 5.5.

5.3.4 Regularity and bounds

The following proposition states the uniform boundedness of the estimates and their partial derivatives.

Proposition 3. *Let R and k_* be as in Proposition 2. There exists*

$$\mathcal{C} = \mathcal{C}\Big(R, \|(\boldsymbol{v_m}, \boldsymbol{\omega_m})\|_{L^\infty(0,+\infty)}, D_*, \|(y, D)\|_{C^1([0,+\infty)\times\mathbb{S}^2)}\Big) > 0$$

such that for every $(\widehat{\boldsymbol{\omega_b}}^0, \widehat{\boldsymbol{v_b}}^0) \in \mathbb{R}^6$ with $\|(\widehat{\boldsymbol{\omega_b}}^0 - \boldsymbol{\omega_b}, \widehat{\boldsymbol{v_b}}^0 - \boldsymbol{v_b})\| \leqslant R$, $k_y, k_D > k_*$, the solution of the Cauchy problem (5.24) satisfies

$$\|\widehat{y}(t,\boldsymbol{\eta})\|, \|\widehat{D}(t,\boldsymbol{\eta})\|, \|\nabla\widehat{y}(t,\boldsymbol{\eta})\|, \|\nabla\widehat{D}(t,\boldsymbol{\eta})\| \leqslant \mathcal{C}, \quad \forall (t,\boldsymbol{\eta}) \in [0,+\infty)\times\mathbb{S}^2.$$

The proof of this result is a corollary of an intermediary result of the proof of Proposition 2 (see Proposition 6 in Section 5.5).

5.3.5 Continuity

The following proposition details the continuity of the estimated state with respect to $(y, D, \boldsymbol{\omega}, \boldsymbol{v})$.

Proposition 4. *Let $T > 0$. Let $(\boldsymbol{v}^n, \boldsymbol{\omega}^n)_{n\in\mathbb{N}}$, $(\boldsymbol{v}, \boldsymbol{\omega}) \in C_b^1([0,+\infty), \mathbb{R}^3)$, $(y^n, D^n)_{n\in\mathbb{N}}$, $(y, D) \in C_b^1([0,+\infty)\times\mathbb{S}^2, \mathbb{R})$ be associated solutions of (5.3) such that*

$$(\boldsymbol{v}^n, \boldsymbol{\omega}^n) \underset{n\to+\infty}{\longrightarrow} (\boldsymbol{v}, \boldsymbol{\omega}) \quad \text{in } C^0([0,T], \mathbb{R}^3)^2, \tag{5.25}$$

$$(y^n, D^n)_{n\in\mathbb{N}} \quad \text{is bounded in } C^1([0,T]\times\mathbb{S}^2, \mathbb{R})^2, \tag{5.26}$$

$$(y^n, D^n) \underset{n\to+\infty}{\longrightarrow} (y, D) \quad \text{in } C^0([0,T]\times\mathbb{S}^2, \mathbb{R})^2. \tag{5.27}$$

Let R and k_* be as in Proposition 2, $(\widehat{y}^n, \widehat{D}^n, \widehat{\boldsymbol{\omega}_b}^n, \widehat{\boldsymbol{v}_b}^n)_{n\in\mathbb{N}}$, $(\widehat{y}, \widehat{D}, \widehat{\boldsymbol{\omega}_b}, \widehat{\boldsymbol{v}_b})$ be associated solutions of the observer (5.24) with $k_y, k_D > k_*$ and initial conditions such that

$$(\widehat{\boldsymbol{\omega}_b}^n, \widehat{\boldsymbol{v}_b}^n)(0) \to (\widehat{\boldsymbol{\omega}_b}, \widehat{\boldsymbol{v}_b})(0) \quad \text{when } n \to +\infty.$$

Then, for every $T > 0$,

$$(\widehat{\boldsymbol{\omega}_b}^n, \widehat{\boldsymbol{v}_b}^n) \underset{n\to+\infty}{\longrightarrow} (\widehat{\boldsymbol{\omega}_b}, \widehat{\boldsymbol{v}_b}) \quad \text{in } C^0([0,T], \mathbb{R}^3)^2, \tag{5.28}$$

$$(\widehat{y}^n, \widehat{D}^n)(t) \underset{n\to+\infty}{\longrightarrow} (\widehat{y}, \widehat{D})(t) \quad \text{in } C^0([0,T]\times\mathbb{S}^2, \mathbb{R})^2, \quad \forall t \in [0,T]. \tag{5.29}$$

The proof of Proposition 4 relies on tools introduced in the proof of Proposition 2, thus, we postpone it to Section 5.6.

5.3.6 Convergence

The following theorem states that the proposed observer is optimal, in the sense that the conditions for convergence of the observer and observability of the system are strictly equivalent.

Theorem 4. *Let R and k_* be as in Proposition 2. For every $(\widehat{\boldsymbol{\omega}_b}^0, \widehat{\boldsymbol{v}_b}^0) \in \mathbb{R}^6$ with $\|(\widehat{\boldsymbol{\omega}_b}^0 - \boldsymbol{\omega}_b, \widehat{\boldsymbol{v}_b}^0 - \boldsymbol{v}_b)\| \leqslant R$, $k_y, k_D > k_*$, the solution of the Cauchy problem (5.24) satisfies*

1. $\|(\widehat{y} - y)(t)\|_{L^\infty(\mathbb{S}^2)} + \|(\widehat{D} - D)(t)\|_{L^\infty(\mathbb{S}^2)} \underset{t\to+\infty}{\longrightarrow} 0$,
2. *for every adherence value $(\overline{\boldsymbol{v}_b}, \overline{\boldsymbol{\omega}_b})$ of $(\widehat{\boldsymbol{v}_b}, \widehat{\boldsymbol{\omega}_b})$ as $[t \to +\infty]$, there exists $\bar{C} \in K \subset \Omega$ a camera position such that, $\forall \boldsymbol{\eta} \in \mathbb{S}^2$*

$$\nabla\bar{y}\cdot\boldsymbol{\eta}\times\left((\overline{\boldsymbol{\omega}_b} - \boldsymbol{\omega}_b) + \frac{1}{\bar{D}}\boldsymbol{\eta}\times(\overline{\boldsymbol{v}_b} - \boldsymbol{v}_b)\right) = 0 \tag{5.30}$$

$$\nabla\bar{D}\cdot\boldsymbol{\eta}\times\left((\overline{\boldsymbol{\omega}_b} - \boldsymbol{\omega}_b) + \frac{1}{\bar{D}}\boldsymbol{\eta}\times(\overline{\boldsymbol{v}_b} - \boldsymbol{v}_b)\right) + \boldsymbol{\eta}\cdot(\overline{\boldsymbol{v}_b} - \boldsymbol{v}_b) = 0 \tag{5.31}$$

where $\bar{y}(t,\boldsymbol{\eta})$ and $\bar{D}(t,\boldsymbol{\eta})$ are the brightness and depth fields associated to the position $\bar{C}$ of the camera. According to Theorem 3, when the system is observable (i.e. when (Σ, y_Σ) does not admit a rotation axis), then $(\widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}})(t) \underset{t\to+\infty}{\longrightarrow} (\boldsymbol{\omega_b}, \boldsymbol{v_b})$.

Proof of Theorem 4:

Step 1: Convergence of $f(t)$, defined by (5.23), to zero when $t \to +\infty$. The function $V(t)$ is nonincreasing and nonnegative, thus it converges when $t \to +\infty$. From (5.23), we deduce that the nonnegative fonction f satisfies

$$\int_0^t f(s)ds \leqslant V(0) - V(t)\,, \quad \forall t > 0\,.$$

Thus $f \in L^1(0,+\infty)$. In order to conclude, it is sufficient to prove that f is uniformly continuous on $[0,+\infty)$ (Barbalat Lemma). We have

$$\begin{aligned}\dot{f} = 2\int_{\mathbb{S}^2} \Big(&\lambda_y^* \widetilde{y}\Big(\nabla\widehat{y}\cdot\boldsymbol{\eta}\times(\widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta}\times\widehat{\boldsymbol{v_b}}) - \nabla\widehat{y}\cdot\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{v}) - k_y\widetilde{y}\Big)\\ &+\lambda_D^*\widetilde{D}\Big(\nabla\widehat{D}\cdot(\boldsymbol{\eta}\times(\widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta}\times\widehat{\boldsymbol{v_b}})) - \boldsymbol{\eta}\cdot\widehat{\boldsymbol{v_b}} - \nabla\widetilde{D}\cdot\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\boldsymbol{v}) - k_D\widetilde{D}\Big)\Big)\,d\sigma_\eta \quad (5.32)\end{aligned}$$

with $\lambda_y^* := \lambda_y\left(k_y - \frac{L}{2}\right)$ and $\lambda_D^* := \lambda_D\left(k_D - \frac{L}{2}\right)$. Thanks to Proposition 3, we get $\dot{f} \in L^\infty(0,+\infty)$, which ends the proof of Step 1.

Step 2: Proof of statement 1. The functions $(\widehat{y}-y, \widehat{D}-D)$ belong to $C_b^1([0,+\infty)\times\mathbb{S}^2,\mathbb{R})^2$ thanks to Proposition 3 and (5.4). Thus Ascoli theorem guarantees the existence of adherence values in $C^0(\mathbb{S}^2,\mathbb{R})^2$ of $(\widehat{y}-y, \widehat{D}-D)(t)$ when $t \to +\infty$. Thanks to Step 1, the only possible adherence value is $(0,0)$, thus the whole function converges to $(0,0)$ in $C^0(\mathbb{S}^2,\mathbb{R})^2$.

Step 3: Proof of statement 2. Let $(\overline{\boldsymbol{v_b}}^0, \overline{\boldsymbol{\omega_b}}^0)$ be an adherence value of $(\widehat{\boldsymbol{v_b}}, \widehat{\boldsymbol{\omega_b}})$ as $t \to +\infty$. Let $(t_n)_{n\in\mathbb{N}}$ be an increasing sequence of $[0,+\infty)$ such that $(\widehat{\boldsymbol{v_b}}, \widehat{\boldsymbol{\omega_b}})(t_n) \to (\overline{\boldsymbol{v_b}}^0, \overline{\boldsymbol{\omega_b}}^0)$. Let $T > 0$. The functions $\boldsymbol{v}, \boldsymbol{\omega}$ (resp. y, D) belong to $C_b^1([0,+\infty),\mathbb{R}^3)^2$ (resp. $C_b^1([0,+\infty)\times\mathbb{S}^2,\mathbb{R})^2$), thus Ascoli's theorem guarantees the existence of $\overline{\boldsymbol{v}}, \overline{\boldsymbol{\omega}} \in C^0([0,T],\mathbb{R}^3)^2$ (resp. $\overline{D}^0, \overline{y}^0 \in C^0(\mathbb{S}^2,\mathbb{R})^2$) such that, up to an extraction

$$\begin{aligned}(\boldsymbol{v},\boldsymbol{\omega})(t_n + .) \to (\overline{\boldsymbol{v}},\overline{\boldsymbol{\omega}})(.) \text{ in } C^0([0,T],\mathbb{R}^3)^2,\\ (y,D)(t_n) \to (\overline{y}^0, \overline{D}^0) \text{ in } C^0(\mathbb{S}^2,\mathbb{R})^2.\end{aligned}$$

The continuity of the flow for (5.3) justifies that

$$(y,D)(t_n + .) \to (\overline{y},\overline{D})(.) \text{ in } C^0([0,T]\times\mathbb{S}^2,\mathbb{R})^2$$

where

$$\begin{cases} \partial_t\overline{y} = -\nabla\overline{y}\cdot(\boldsymbol{\eta}\times(\overline{\boldsymbol{\omega}} + \frac{1}{\overline{D}}\boldsymbol{\eta}\times\overline{\boldsymbol{v}})),\\ \partial_t\overline{D} = -\nabla\overline{D}\cdot(\boldsymbol{\eta}\times(\overline{\boldsymbol{\omega}} + \overline{D}\boldsymbol{\eta}\times\overline{\boldsymbol{v}})) - \overline{\boldsymbol{v}}\cdot\boldsymbol{\eta},\\ \overline{y}(0) = \overline{y}^0,\\ \overline{D}(0) = \overline{D}^0. \end{cases}$$

Thanks to Proposition 4, we know that

$$(\widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}})(t_n + .) \to (\widehat{\overline{\boldsymbol{\omega_b}}}, \widehat{\overline{\boldsymbol{v_b}}}) \quad \text{in } C^0([0,T], \mathbb{R}^3)^2,$$
$$(\widehat{y}, \widehat{D})(t_n + .) \to (\widehat{\overline{y}}, \widehat{\overline{D}}) \quad \text{in } C^0([0,T] \times \mathbb{S}^2, \mathbb{R})^2,$$

where $(\widehat{\overline{y}}, \widehat{\overline{D}}, \widehat{\overline{\boldsymbol{\omega_b}}}, \widehat{\overline{\boldsymbol{v_b}}})$ is the solution of the observer associated to $(\overline{y}, \overline{D})$, $\overline{\boldsymbol{\omega}}_m(t) = \overline{\boldsymbol{\omega}}(t) + \boldsymbol{\omega_b}$, $\overline{\boldsymbol{v}}_m(t) = \overline{\boldsymbol{v}}(t) + \boldsymbol{v_b}$ and the initial conditions $(\widehat{\overline{\boldsymbol{\omega_b}}}, \widehat{\overline{\boldsymbol{v_b}}})(0) = (\overline{\boldsymbol{\omega_b}}^0, \overline{\boldsymbol{v_b}}^0)$.

We know that $V(t)$ converges to some limit V_∞ as $t \to +\infty$. Moreover, $V(t_n + t) \to \overline{V}(t)$ for every $t \in [0,T]$. Therefore, $\overline{V}(t) = V_\infty$ for every $t \in [0,T]$. In particular, $0 \leqslant \overline{f}(t) \leqslant |d\overline{V}/dt| \equiv 0$ i.e. $(\widehat{\overline{y}} - \overline{y})(t) = (\widehat{\overline{D}} - \overline{D})(t) = 0, \forall t \in [0,T]$. Substracting the equations on $\widehat{\overline{y}}$ and $\overline{y}$ (resp. on $\widehat{\overline{D}}$ and $\overline{D}$), we get the relation (5.30) (resp. (5.31)). □

5.4 Practical implementation, simulations and experimental results

5.4.1 Adaptation to a spherical cap

Concretely, a spherical camera is only a model, and the image perceived by real cameras only occupy a part of $\mathbb{S}^2$. Let us call K this portion: $y(t,\boldsymbol{\eta})$ and $D(t,\boldsymbol{\eta})$ are known only for $\boldsymbol{\eta} \in K$. The observer introduced in Section 5.3 can not be readily used since it brings into play the integral of y or D over the whole unit sphere. We will see that one can compensate this problem by considering virtual observations, equal to the real observations over the window defined by K. Let K_1 and K_2 be two compact sets s.t. $\overset{\circ}{K_1} \subset \overset{\circ}{K_2} \subset \overset{\circ}{K}$. Let ϕ be a $\mathcal{C}^\infty$ scalar field $\mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto \phi(\boldsymbol{\eta}) \in \mathbb{R}$, s.t. $\phi = 1$ on K_1, $\phi = 0$ and $\nabla\phi = 0$ on $\mathbb{S}^2 \setminus \overset{\circ}{K_2}$. Let us define $X = \phi y$ and $\Lambda = \phi D$. Then,

$$\partial_t X = -\nabla X \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})) + y\nabla\phi \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v}))$$

and

$$\partial_t \Lambda = -\nabla\Lambda \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})) + D\nabla\phi \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})) - \phi\boldsymbol{\eta} \cdot \boldsymbol{v}.$$

We propose the following adaptation of observer (5.16):

$$\left\{\begin{array}{ll}
\partial_t \widehat{X} = & -\nabla\widehat{X} \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}))) \\
& + y\nabla\phi \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}))) \\
& + k_X(X - \widehat{X}), \\
\partial_t \widehat{\Lambda} = & -\nabla\widehat{\Lambda} \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}))) \\
& + D\nabla\phi \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}))) \\
& - \phi(\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}) \cdot \boldsymbol{\eta} + k_\Lambda(\Lambda - \widehat{\Lambda}), \\
\partial_t \widehat{\boldsymbol{\omega_b}} = & -k_\omega \int_K \lambda_X(\widehat{X} - X)((\nabla\widehat{X} - y\nabla\phi) \times \boldsymbol{\eta}) \\
& + \lambda_\Lambda(\widehat{\Lambda} - \Lambda)((\nabla\widehat{\Lambda} - D\nabla\phi) \times \boldsymbol{\eta}) d\sigma_\eta, \\
\partial_t \widehat{\boldsymbol{v_b}} = & -k_v \int_K \lambda_\Lambda(\widehat{\Lambda} - \Lambda)\boldsymbol{\eta} \\
& + \lambda_X(\widehat{X} - X)(\frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{\eta} \times (\nabla\widehat{X} - y\nabla\phi)) \\
& + \lambda_\Lambda(\widehat{\Lambda} - \Lambda)(\frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{\eta} \times (\nabla\widehat{\Lambda} - D\nabla\phi)) d\sigma_\eta.
\end{array}\right. \tag{5.33}$$

Notice that the estimates $\widehat{X}$ and $\widehat{\Lambda}$ of the virtual observations are computed only on the domain K, and not on the entire sphere $\mathbb{S}^2$, and only their values on K are used in the dynamics of the estimated biases. Neumann boundary conditions are imposed on the parts of ∂K where the direction of propagation points toward the inside of the domain K:

$$\frac{\partial \widehat{X}}{\partial \mathbf{n}} = 0, \quad \frac{\partial \widehat{\Lambda}}{\partial \mathbf{n}} = 0$$

where $\mathbf{n}$ is the outwards-pointing normal of ∂K such that

$$\mathbf{n} \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}} + \frac{1}{D}\boldsymbol{\eta} \times (\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}}))) < 0.$$

Let us choose the candidate Lyapunov function

$$V = \frac{1}{2}\left(\int_K \left(\lambda_X \widetilde{X}^2 + \lambda_\Lambda \widetilde{\Lambda}^2\right) d\sigma_\eta + \frac{\widetilde{\boldsymbol{\omega_b}}^2}{k_\omega} + \frac{\widetilde{\boldsymbol{v_b}}^2}{k_v}\right).$$

One can prove that

$$\dot{V} = \int_K \left(\lambda_X \widetilde{X}\left(-\nabla \widetilde{X} \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}(\boldsymbol{\eta} \times \boldsymbol{v}))) - k_X \widetilde{X}\right)\right) d\sigma_\eta + \int_K \left(\lambda_\Lambda \widetilde{\Lambda}\left(-\nabla \widetilde{\Lambda} \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}(\boldsymbol{\eta} \times \boldsymbol{v}))) - k_\Lambda \widetilde{\Lambda}\right)\right) d\sigma_\eta$$

and integration by parts yields

$$\dot{V} = -\int_K \left(\lambda_X \left(k_X - \frac{\nabla \cdot \boldsymbol{W}(t,\boldsymbol{\eta})}{2}\right)\widetilde{X}^2 + \lambda_\Lambda \left(k_\Lambda - \frac{\nabla \cdot \boldsymbol{W}(t,\boldsymbol{\eta})}{2}\right)\widetilde{\Lambda}^2\right) d\sigma_\eta - \oint_{\partial K} \left(\lambda_X \widetilde{X}^2 + \lambda_\Lambda \widetilde{\Lambda}^2\right) \frac{\boldsymbol{W}(t,\boldsymbol{\eta})}{2} \cdot \vec{\mathbf{n}}\, dl_\eta$$

where $\boldsymbol{W}$ is defined as $(\mathbb{R}, \mathbb{S}^2) \ni (t, \boldsymbol{\eta}) \mapsto \boldsymbol{W}(t,\boldsymbol{\eta}) = \boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v}) \in \mathbb{R}^3$. We guess that for given environment and trajectory, sufficiently large correction gains k_X and k_Λ can ensure the decrease of the candidate Lyapunov function V. Then convergence analysis done when $K = \mathbb{S}^2$ should be extended to compact sub-domains K of $\mathbb{S}^2$. As the candidate Lyapunov function should decrease unless $\widetilde{X} = \widetilde{\Lambda} = 0$ on K, we guess that the necessary and sufficient condition for observability is the same as stated in Theorem 3, but restricted to the visible part of the environment. From a practical point of view, it is possible that specific views along the camera trajectory admit a cylindrical symmetry, although the environment does not: in that case, a residual error coupled between linear and angular velocity might subsist. However, as soon as a new view enables to raise the ambiguity (for example, a new wall appears in the field of view), this residual error vanishes. The observability condition in that situation does depend in a certain sense on the camera trajectory.

5.4.2 The observer in pinhole coordinates

The previous observer (5.33) can be finally adapted to a real model of camera: we choose here the widely spread pinhole camera model enabling a correspondence between the local coordinates on $\mathbb{S}^2$ with a rectangular grid of pixels. The pixel of coordinates (z_1, z_2) corresponds to the unit vector $\boldsymbol{\eta} \in \mathbb{S}^2$ of coordinates in $\mathbb{R}^3$: $\left(1 + z_1^2 + z_2^2\right)^{-1/2} (z_1, z_2, 1)^T$. The optical camera axis (pixel $(z_1, z_2) = (0, 0)$) corresponds here to the direction z_3. Directions 1 and 2 correspond respectively to the horizontal axis from left to right and to the vertical axis from top to bottom on the image frame.

The gradients ∇y and ∇D must be expressed with respect to z_1 and z_2 (see Subsection 1.3.2 for the detailed computations). Plugging these expressions in (5.33), we get a PDE system written in terms of (t, z_1, z_2) as independent variables. Due to space limitation, this system is not given here, but its derivation is straightforward and a little bit tedious.

5.4.3 Simulations

The non-linear asymptotic observer (5.33) is tested on a sequence of synthetic images characterized by the following:

- *virtual camera providing images restricted to the spherical cap* $K \subset \mathbb{S}^2$: the size of each image is 640 by 480 pixels, the frame rate of the sequence is 42 Hz and the field of view is 50 deg by 40 deg.
- *motion of the virtual camera* $(\boldsymbol{\omega}, \boldsymbol{v})$: it consists of the motion of a real hand-held camera (filtered data), combining translations and rotations in each direction; the real linear and angular velocities expressed in the camera frame are plotted in Fig ures 5.2 and Fig.5.3; zero-mean normally distributed noise (standard deviations σ_v and σ_ω) is added to these velocities to test the robustness;
- *virtual scene corresponding to* (Σ, y_Σ): it consists of the walls, ceiling and floor of a virtual room; the observed walls are virtually painted with a gray pattern, whose intensity varies in horizontal and vertical directions as a sinusoid function;
- *generation of the brightness images* y: each pixel of an image has an integer value varying from 1 to 256, directly depending on the intensity of the observed surface in the direction indexed by the pixel, to which a zero-mean normally distributed noise with standard deviation σ_y is added to test the robustness;
- *generation of the depth images* D: to each pixel of the rectangular grid of an image is attributed the depth of the corresponding element of the observed surface, computed with respect to position and orientation of the camera in the room, to which a zero-mean normally distributed noise with standard deviation σ_D is added to test the robustness.

The numerical resolution used to compute $\widehat{y}$, $\widehat{D}$, $\widehat{\boldsymbol{\omega_b}}$ and $\widehat{\boldsymbol{v_b}}$ according to (5.33) is based on a temporal Euler discretization scheme where $\nabla\widehat{y}$ and $\nabla\widehat{D}$ are computed via differentiation filters (Sobel filtering) directly from the image and depth previous estimates. The observer is then tested for reasonable biases: in rotation, a bias of 0.05 rad.s^{-1} (10,000 deg/h, for a low-cost gyroscope) around the horizontal axis; in translation, a bias of 2.5 m.s^{-1} in the horizontal direction (9 km/h, for the windspeed). In other words, $\boldsymbol{\omega_{b}}_x = 0.05$ rad.s^{-1} and $\boldsymbol{v_{b}}_x = 2.5$ m.s^{-1}. Biases in the other

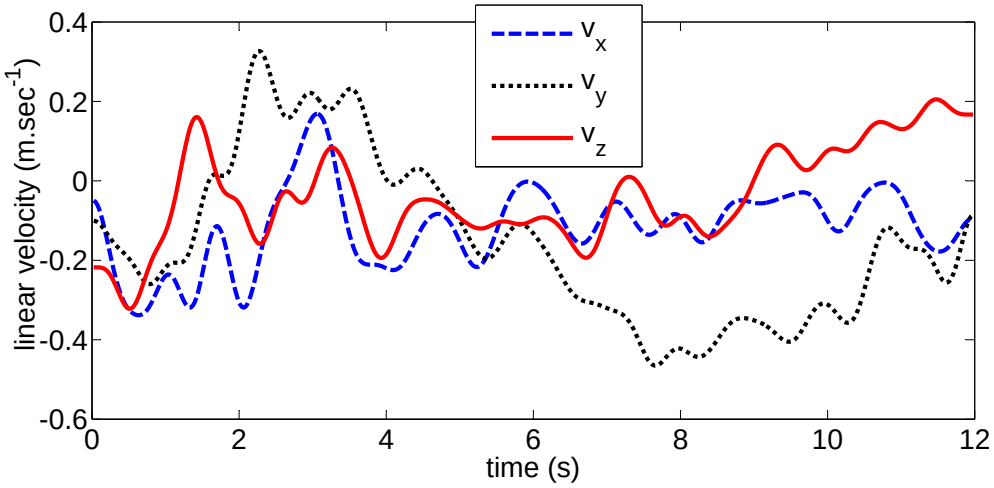


Figure 5.2: Components of the linear velocity v used to test the observer: translations in the horizontal, vertical and optical axis directions, respectively.

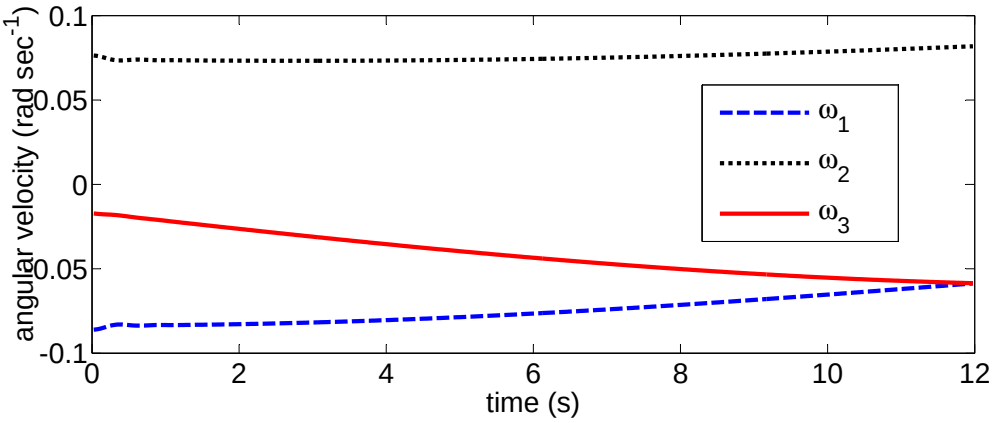


Figure 5.3: Components of the angular velocity ω used to test the observer: rotations around the horizontal, vertical and optical axis, respectively.

directions are set to 0. Initial conditions for $\widehat{y}$ and $\widehat{D}$ are $y(0, \boldsymbol{\eta})$ and $D(0, \boldsymbol{\eta})$. Initial conditions for the estimated biases are set to zero. The chosen correction gains are: $k_y = k_D = 2s^{-1}$, $k_v = 10^{-2}\text{m}^2.s^{-2}$ and $k_\omega = 10^{-5}\text{rad.m}.s^{-2}$. These correction gains are chosen in accordance with the expected values of biases and the scene averaged depth, to enable a reasonable convergence speed. The correction gains k_y and k_D are comparatively much larger than k_v, which is itself larger than k_ω, as large oscillations in the estimation of $\boldsymbol{\omega_b}$ can make the discretized observer to diverge. Finally, the ponderation coefficients are $\lambda_y = 1$ and $\lambda_D = 5000\text{m}^2$, chosen to compensate the difference of magnitudes of y and D. First, when image and depth data contain no noise ($\sigma_y = \sigma_D = 0$), the results are plotted in Figure 5.4 and Figure 5.5 as the instantaneous errors of estimations $\widetilde{\boldsymbol{v_b}}$ and $\widetilde{\boldsymbol{\omega_b}}$ expressed in the camera frame, respectively. In the first 6 s, errors slowly converge towards 0, and coupling between rotation and translation occurs: this reflects the fact that an horizontal translation can be interpreted as a rotation around the vertical axis to a certain extent. Oscillations decrease, and after convergence, errors stay bounded: for the bias in rotation, it does not exceed 0.002 rad.s^{-1} (4 % of the original bias); in translation, it is less than 0.01 m.s^{-1} (0.4 % of the original bias).

Then, to test the robustness of the method, noise is added to the image data ($\sigma_y = 30$, about 12% of the full scale), to the depth data ($\sigma_D = 25$ cm, which is three times as much that can be expected from a Kinect device), to linear velocity ($\sigma_v = 0.05\text{m}.s^{-1}$) and to angular velocity ($\sigma_\omega = 0.005$ rad.s^{-1}). Results are plotted in Figure 5.6 and Figure 5.7. Convergence time is shorter for biases in translation estimation: after 3 s, error does not exceed 0.2 m.s^{-1} (8 % of the original bias). For the rotation, convergence is slower (as k_ω is smaller), but in the last 3 sec of the simulation, biases are estimated up to 0.003 rad.s^{-1} (6 % of the original bias).

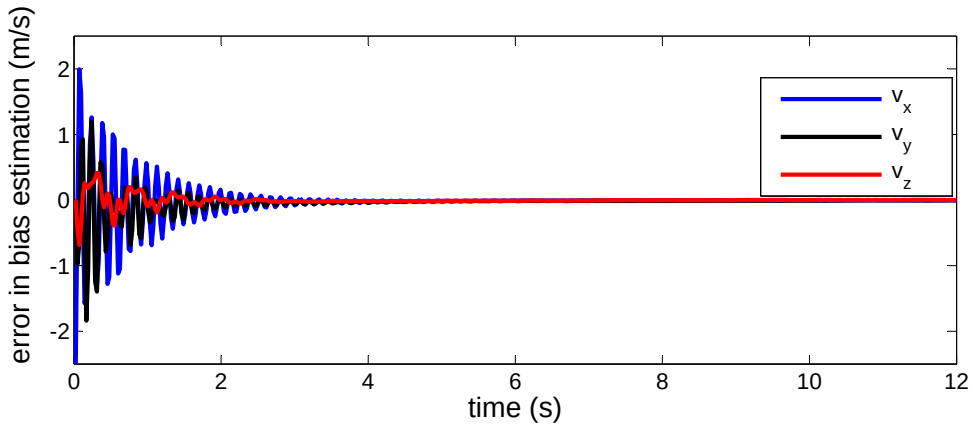


Figure 5.4: Error of estimation of the biases in translation for noiseless image and depth data. Real biases are $\boldsymbol{\omega}_{\boldsymbol{b}x} = 0.05$ rad.s^{-1} and $\boldsymbol{v}_{\boldsymbol{b}x} = 2.5$ m.s^{-1}

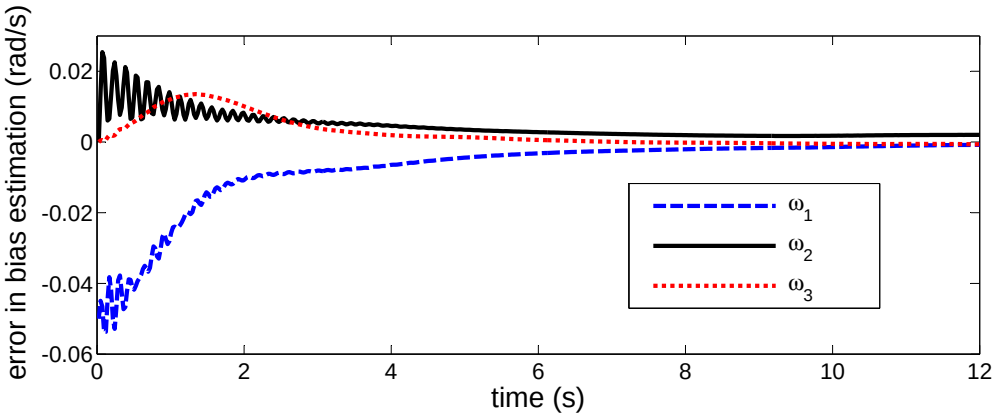


Figure 5.5: Error of estimation of the biases in rotation for noiseless image and depth data. Real biases are $\boldsymbol{\omega}_{\boldsymbol{b}x} = 0.05$ rad.s^{-1} and $\boldsymbol{v}_{\boldsymbol{b}x} = 2.5$ m.s^{-1}

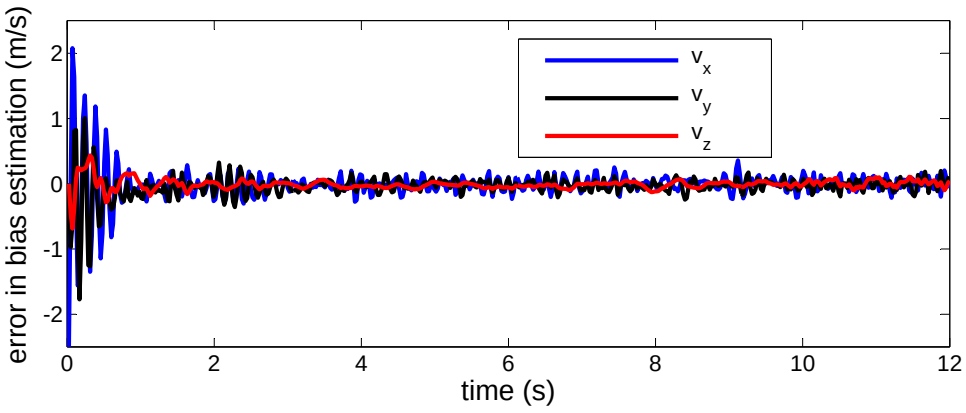


Figure 5.6: Error of estimation of the biases in translation for noisy image and depth data, and noisy velocities: σ_y = 30, σ_D = 25 cm, σ_v = 0.05m.s^{-1}, σ_ω = 0.005 rad.s^{-1}. Real biases are $\boldsymbol{\omega}_{\boldsymbol{b}x}$ = 0.05 rad.s^{-1} and $\boldsymbol{v}_{\boldsymbol{b}x}$ = 2.5 m.s^{-1}

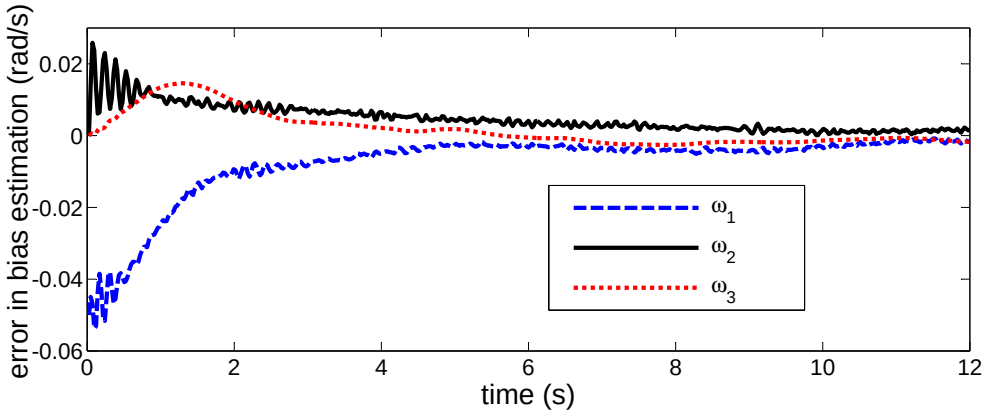


Figure 5.7: Error of estimation of the biases in rotation for noisy image and depth data, and noisy velocities: σ_y = 30, σ_D = 25 cm, σ_v = 0.05m.s^{-1}, σ_ω = 0.005 rad.s^{-1}. Real biases are $\boldsymbol{\omega}_{\boldsymbol{b}x}$ = 0.05 rad.s^{-1} and $\boldsymbol{v}_{\boldsymbol{b}x}$ = 2.5 m.s^{-1}

5.4.4 Experiments on real data

The observer described by the system of PDEs (5.33) is then tested on a sequence of real RGB and depth images. These images are acquired by a Kinect device, a pair of RBG and depth images is acquired at the same instant, at a frame rate of 25 Hz (see Section 9.1 for more details on the Kinect device). Gyrometers are embedded along with this device and provide angular velocity measurements at a frame rate of 125Hz. The system was provided by Sysnav company. Unfortunately, for this experiment, we did not have a biased linear velocity sensor synchronized with the system available. To overcome this issue, the system was only submitted to rotations, and only biases on angular velocity were estimated. More specifically, this is achieved by replacing (5.33) by the following system

$$\left\{\begin{array}{ll} \partial_t \widehat{X} = & -\nabla \widehat{X} \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}})) + y \nabla \phi \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}})) + k_X (X - \widehat{X}), \\ \partial_t \widehat{\Lambda} = & -\nabla \widehat{\Lambda} \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}})) + D \nabla \phi \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}})) + k_\Lambda (\Lambda - \widehat{\Lambda}), \\ \partial_t \widehat{\boldsymbol{\omega_b}} = & -k_\omega \int_K \lambda_X (\widehat{X} - X)((\nabla \widehat{X} - y \nabla \phi) \times \boldsymbol{\eta}) + \lambda_\Lambda (\widehat{\Lambda} - \Lambda)((\nabla \widehat{\Lambda} - D \nabla \phi) \times \boldsymbol{\eta}) d\sigma_\eta. \end{array}\right. \tag{5.34}$$

The measured angular velocity is plotted in Figure 5.8: the motion is relatively slow, with an order of magnitude of 0.15 rad.s^{-1} (8.5 deg.s^{-1}). In the first 40 sec, the system stands still, which

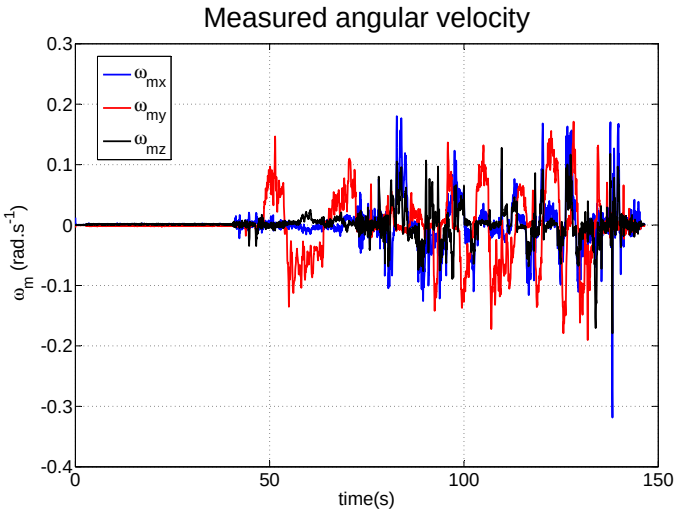


Figure 5.8: Angular velocity as measured by gyrometers embedded on the system.

reveals a bias on angular velocity: this bias is estimated by a rough mean calculation. This yields

$$\boldsymbol{\omega_b} = \begin{pmatrix} 1.0304 \\ -1.0384 \\ 1.2129 \end{pmatrix} 10^{-3} \text{rad}.s^{-1}.$$

The sequence of RGBD images is processed along with the measurements from gyrometers, which yields an estimation of angular velocity biases plotted in Figure 5.9. Biases in the x-axis and

y-axis directions converge quickly (about 50 sec) and are submitted to perturbations, but bias in the z-axis direction still evolves at the end of the sequence. The same sequence is thus processed a second time, with the biases initialized at their final values at the end of the first processing. This operation is repeated a few times, and the resulting estimated biases are plotted in Figure 5.10. After several iterations, the tendency of evolution is stationary: biases are submitted to perturbations but have a constant mean. The perturbations can be the consequence of several factors, the main being that the hypothesis of null linear velocity may be unsatisfied. The mean of estimated biases is

$$\widehat{\boldsymbol{\omega_b}}_{mean} = \begin{pmatrix} 1.050 \\ -0.975 \\ 1.170 \end{pmatrix} 10^{-3} \text{rad}.s^{-1}.$$

The magnitude of variations around this mean is about $10^{-4}\text{rad}.s^{-1}$. From this real data experiment, the method can be validated: a precision of $10^{-4}\text{rad}.s^{-1}$ ($5.7\text{mdeg}.s^{-1}$) can be expected on the angular velocity bias estimation.

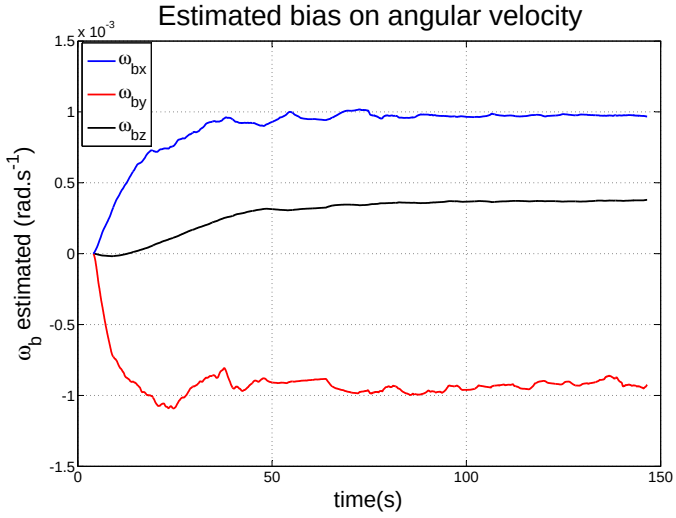


Figure 5.9: Estimated biases resulting from the processing of RGBD images and biased angular velocity. Bias in the z-axis direction (black) is still converging at the end of the sequence.

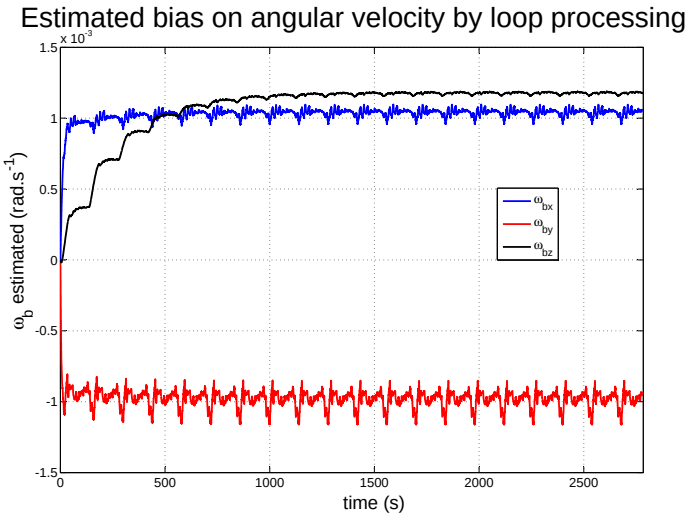


Figure 5.10: Estimated biased on angular velocity by loop processing of the same sequence of RGBD images and biased angular velocity measurements. This shows the convergence of biases in all directions towards a mean value, and perturbations around this mean value.

5.5 Proof of the well posedness of the observer

The goal of this section is the proof of Proposition 2 in 3 steps:
- the first step consists in proving the existence of local (in time) solutions,
- the second step consists in proving that solutions stays bounded in a appropriate way, so that
- we can iterate the local argument to get global (in time) solutions, in a third step.

5.5.1 Local solutions

The goal of this section is the proof of the following result.

Proposition 5. *Let R and $M > 0$. There exists*

$$T = T(M, R, \|(\boldsymbol{v_m}, \boldsymbol{\omega_m})\|_{L^\infty(0,+\infty)}, D_*, \|(y, D)\|_{C^1([0,+\infty)\times\mathbb{S}^2)}, k_y, k_D, \lambda_y, \lambda_D) > 0$$

such that, for every $\widehat{\boldsymbol{\omega_b}}^0$, $\widehat{\boldsymbol{v_b}}^0 \in \mathbb{R}^3$, $\widehat{y}^0$, $\widehat{D}^0 \in C^1(\mathbb{S}^2)$ with

$$\|(\widehat{\boldsymbol{\omega_b}}^0 - \boldsymbol{\omega_b}, \widehat{\boldsymbol{v_b}}^0 - \boldsymbol{v_b})\| \leqslant R\,, \tag{5.35}$$

$$\|\widehat{y}^0\|_{C^1(\mathbb{S}^2)}\,, \|\widehat{D}^0\|_{C^1(\mathbb{S}^2)} \leqslant M\,, \tag{5.36}$$

there exists a unique solution $(\widehat{y},\widehat{D},\widehat{\boldsymbol{\omega_b}},\widehat{\boldsymbol{v_b}}) \in C^1([0,T]\times\mathbb{S}^2,\mathbb{R})^2\times C^1([0,T],\mathbb{R}^3)^2$*, defined on the time interval* $[0,T]$*, of the Cauchy-problem*

$$\left\{\begin{array}{ll}
\partial_t\widehat{y} & = -\nabla\widehat{y}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m}-\widehat{\boldsymbol{\omega_b}}+\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m}-\widehat{\boldsymbol{v_b}})))+k_y(y-\widehat{y})\,,\\
\partial_t\widehat{D} & = -\nabla\widehat{D}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m}-\widehat{\boldsymbol{\omega_b}}+\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m}-\widehat{\boldsymbol{v_b}})))-(\boldsymbol{v_m}-\widehat{\boldsymbol{v_b}})\cdot\boldsymbol{\eta}\\
 & \quad +k_D(D-\widehat{D})\,,\\
\frac{d\widehat{\boldsymbol{\omega_b}}}{dt} & = -\int_{\mathbb{S}^2}\big(\lambda_y(\widehat{y}-y)(\nabla\widehat{y}\times\boldsymbol{\eta})+\lambda_D(\widehat{D}-D)(\nabla\widehat{D}\times\boldsymbol{\eta})\big)\,d\sigma_\eta,\\
\frac{d\widehat{\boldsymbol{v_b}}}{dt} & = -\int_{\mathbb{S}^2}\big(\lambda_D(\widehat{D}-D)\boldsymbol{\eta}+\lambda_y(\widehat{y}-y)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{y})\\
 & \quad +\lambda_D(\widehat{D}-D)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{D}))\big)\,d\sigma_\eta\,,\\
\widehat{y}(0,\boldsymbol{\eta}) & = \widehat{y}^0(\boldsymbol{\eta}),\\
\widehat{D}(0,\boldsymbol{\eta}) & = \widehat{D}^0(\boldsymbol{\eta}),\\
\widehat{\boldsymbol{\omega_b}}(0) & = \widehat{\boldsymbol{\omega_b}}^0,\\
\widehat{\boldsymbol{v_b}}(0) & = \widehat{\boldsymbol{v_b}}^0.
\end{array}\right. \tag{5.37}$$

Proof of Proposition 5: We prove the existence and uniqueness of the non-linear and non-local system of PDEs (5.37) by a standard fixed-point approach. Our strategy consists in explicitly solving the two first PDEs of (5.37) with fixed functions $t\mapsto(\widehat{\boldsymbol{W_b}},\widehat{\boldsymbol{V_b}})(t)$, instead of $t\mapsto(\widehat{\boldsymbol{\omega_b}},\widehat{\boldsymbol{v_b}})(t)$. Then, the explicit solutions $(\widehat{Y}(t),\widehat{d}(t))$ are plugged in the third and fourth ODEs of (5.37). We conclude by applying the Banach fixed point theorem (see [Brezis, 1983]) on the map $(\widehat{\boldsymbol{W_b}},\widehat{\boldsymbol{V_b}})\mapsto(\widehat{\boldsymbol{\omega_b}},\widehat{\boldsymbol{v_b}})$.

Let $R,M>0$, $\widehat{\boldsymbol{\omega_b}}^0$, $\widehat{\boldsymbol{v_b}}^0\in\mathbb{R}^3$, and $\widehat{y}^0$, $\widehat{D}^0\in C^1(\mathbb{S}^2)$ such that (5.35) and (5.36) hold. For any $T>0$, the functional space

$$E_T := \{(\widehat{\boldsymbol{W_b}},\widehat{\boldsymbol{V_b}})\in C^0([0,T],\mathbb{R}^6);\|(\widehat{\boldsymbol{W_b}},\widehat{\boldsymbol{V_b}})(t)-(\boldsymbol{\omega_b},\boldsymbol{v_b})\|\leqslant 2R,\forall t\in[0,T]\}$$

equipped with the $L^\infty((0,T),\mathbb{R}^6)$-norm is a Banach space. To any element $(\widehat{\boldsymbol{W_b}},\widehat{\boldsymbol{V_b}})\in E_T$, we associate the solutions $(\widehat{Y},\widehat{d},\widehat{\boldsymbol{\omega_b}},\widehat{\boldsymbol{v_b}})\in C^1([0,T]\times\mathbb{S}^2)^2\times C^1([0,T],\mathbb{R}^3)^2$ of the Cauchy problem

$$\left\{\begin{array}{ll}
\partial_t\widehat{Y} & = -\nabla\widehat{Y}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m}-\widehat{\boldsymbol{W_b}}+\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m}-\widehat{\boldsymbol{V_b}})))+k_y(y-\widehat{Y})\,,\\
\partial_t\widehat{d} & = -\nabla\widehat{d}\cdot(\boldsymbol{\eta}\times(\boldsymbol{\omega_m}-\widehat{\boldsymbol{W_b}}+\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{v_m}-\widehat{\boldsymbol{V_b}})))-(\boldsymbol{v_m}-\widehat{\boldsymbol{V_b}})\cdot\boldsymbol{\eta}+k_D(D-\widehat{d})\,,\\
\frac{d\widehat{\boldsymbol{\omega_b}}}{dt} & = -\int_{\mathbb{S}}\big(\lambda_y(\widehat{Y}-y)(\nabla\widehat{Y}\times\boldsymbol{\eta})+\lambda_D(\widehat{d}-D)(\nabla\widehat{d}\times\boldsymbol{\eta})\big)\,d\sigma_\eta\,,\\
\frac{d\widehat{\boldsymbol{v_b}}}{dt} & = -\int_{\mathbb{S}}\big(\lambda_D(\widehat{d}-D)\boldsymbol{\eta}+\lambda_y(\widehat{Y}-y)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{Y})\\
 & \quad +\lambda_D(\widehat{d}-D)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{d}))\big)\,d\sigma_\eta\,,\\
\widehat{Y}(0,\boldsymbol{\eta}) & = \widehat{y}^0(\boldsymbol{\eta})\,,\\
\widehat{d}(0,\boldsymbol{\eta}) & = \widehat{D}^0(\boldsymbol{\eta})\,,\\
\widehat{\boldsymbol{\omega_b}}(0) & = \widehat{\boldsymbol{\omega_b}}^0\,,\\
\widehat{\boldsymbol{v_b}}(0) & = \widehat{\boldsymbol{v_b}}^0\,.
\end{array}\right. \tag{5.38}$$

Let us emphasize that this Cauchy problem is well posed because $\widehat{Y}$ and $\widehat{d}$ have explicit expressions along the characteristics (see [LeVeque, 1992, Serre, 1999] for a description of the method of characteristics).

Indeed, let us introduce the flow Φ_t on the sphere $\mathbb{S}^2$ associated to the ODE

$$\left\{\begin{array}{l} \frac{d\boldsymbol{\eta}}{dt} = \boldsymbol{\eta}\times\Big(\boldsymbol{\omega_m}(t) - \widehat{\boldsymbol{W_b}}(t) + \frac{1}{D(t,\boldsymbol{\eta})}\boldsymbol{\eta}\times[\boldsymbol{v_m}(t) - \widehat{\boldsymbol{V_b}}(t)]\Big), \\ \boldsymbol{\eta}(0) = \boldsymbol{\eta}_0\,. \end{array}\right. \tag{5.39}$$

This means that the solution of (5.39) is $\boldsymbol{\eta}(t) = \Phi_t(\boldsymbol{\eta}_0)$, $\forall t, \boldsymbol{\eta}_0$. Note that the right hand side of this equation is continuous in t and C^1 in $\boldsymbol{\eta}$, thus the flow is uniquely defined. This flow is defined for every $t\in[0,T]$ because no explosion is possible in finite time: it lives on the sphere $\mathbb{S}^2$. Moreover, for every $t\in[0,T]$, $\boldsymbol{\eta}\mapsto\Phi_t(\boldsymbol{\eta})$ is a C^1-diffeomorphism of the sphere $\mathbb{S}^2$, thus, there exists

$$C = C(R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,T)}, D_*, \|D\|_{C^1([0,T]\times\mathbb{S}^2)}) > 0$$

such that

$$\left\|\frac{\partial\Phi_t}{\partial\boldsymbol{\eta}}(\boldsymbol{\eta})\right\|, \left\|\frac{\partial\Phi_t^{-1}}{\partial\boldsymbol{\eta}}(\boldsymbol{\eta})\right\| \leqslant C, \quad \forall(t,\boldsymbol{\eta})\in[0,T]\times\mathbb{S}^2. \tag{5.40}$$

With the characteristics method, we get the following explicit expressions

$$\widehat{Y}[t,\Phi_t(\boldsymbol{\eta})] = \widehat{y}^0(\boldsymbol{\eta})e^{-k_y t} + \int_0^t k_y y[\tau,\Phi_\tau(\boldsymbol{\eta})]e^{k_y(\tau-t)}d\tau,$$

$$\begin{array}{rl} \widehat{d}[t,\Phi_t(\boldsymbol{\eta})] = & \widehat{D}^0(\boldsymbol{\eta})e^{-k_D t} \\ & + \int_0^t \Big(k_D D(\tau,\Phi_\tau(\boldsymbol{\eta})) - (\boldsymbol{v_m} - \widehat{\boldsymbol{V_b}})(\tau).\Phi_\tau(\boldsymbol{\eta})\Big)e^{k_D(\tau-t)}d\tau, \end{array}$$

or, equivalently

$$\widehat{Y}[t,\boldsymbol{\eta}] = \widehat{y}^0[0,\Phi_t^{-1}(\boldsymbol{\eta})]e^{-k_y t} + \int_0^t k_y y[\tau,\Phi_\tau\circ\Phi_t^{-1}(\boldsymbol{\eta})]e^{k_y(\tau-t)}d\tau, \tag{5.41}$$

$$\begin{array}{rl} \widehat{d}[t,\boldsymbol{\eta}] = & \widehat{D}^0[\Phi_t^{-1}(\boldsymbol{\eta})]e^{-k_D t} \\ & + \int_0^t \Big(k_D D[\tau,\Phi_\tau\circ\Phi_t^{-1}(\boldsymbol{\eta})] - (\boldsymbol{v_m} - \widehat{\boldsymbol{V_b}})(\tau).[\Phi_\tau\circ\Phi_t^{-1}](\boldsymbol{\eta})\Big)e^{k_D(\tau-t)}d\tau. \end{array} \tag{5.42}$$

Thus, $\widehat{\boldsymbol{\omega_b}}$ and $\widehat{\boldsymbol{v_b}}$ are also explicit:

$$\widehat{\boldsymbol{\omega_b}}(t) = \widehat{\boldsymbol{\omega_b}}^0 - \int_0^t\int_{\mathbb{S}^2}\Big(\lambda_y(\widehat{Y}-y)(\nabla\widehat{Y}\times\boldsymbol{\eta}) + \lambda_D(\widehat{d}-D)(\nabla\widehat{d}\times\boldsymbol{\eta})\Big)d\sigma_\eta d\tau, \tag{5.43}$$

$$\begin{array}{rl} \widehat{\boldsymbol{v_b}}(t) = \widehat{\boldsymbol{v_b}}^0 & - \int_0^t\int_{\mathbb{S}^2}\Big(\lambda_D(\widehat{d}-D)\boldsymbol{\eta} + \lambda_y(\widehat{Y}-y)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{Y}) \\ & + \lambda_D(\widehat{d}-D)(\frac{1}{D}\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\nabla\widehat{d})\Big)d\sigma_\eta d\tau. \end{array} \tag{5.44}$$

Now, let us introduce the map

$$\left|\begin{array}{rccc} F_T: & E_T & \to & C^0([0,T],\mathbb{R}^6) \\ & (\widehat{\boldsymbol{W_b}},\widehat{\boldsymbol{V_b}}) & \mapsto & (\widehat{\boldsymbol{\omega_b}},\widehat{\boldsymbol{v_b}}). \end{array}\right.$$

Step 1: Let us prove the existence of

$$T^* = T^*(M,R,\|(\boldsymbol{\omega_m},\boldsymbol{v_m})\|_{L^\infty(0,1)}, D_*, \|(D,y)\|_{C^1([0,1]\times\mathbb{S}^2)}, k_y, k_D, \lambda_y, \lambda_D)\in(0,1)$$

such that F_T maps E_T into E_T for every $T < T^*$. From now on, we assume that $T \in (0,1)$. Let $(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}}) \in E_T$. Thanks to the explicit expressions (5.41), (5.42) and the bounds (5.40), (5.4) there exists a constant

$$C_1 = C_1(M, R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,1)}, D_*, \|(y, D)\|_{C^1([0,1]\times\mathbb{S}^2)}, k_y, k_D) > 0$$

such that

$$|\widehat{Y}(t,\boldsymbol{\eta})|, |\nabla\widehat{Y}(t,\boldsymbol{\eta})|, |\widehat{d}(t,\boldsymbol{\eta})|, |\nabla\widehat{d}(t,\boldsymbol{\eta})| \leqslant C_1, \quad \forall (t,\boldsymbol{\eta}) \in [0,T]\times\mathbb{S}^2. \tag{5.45}$$

Then, we deduce from (5.43), (5.44) and (5.35) that

$$|\widehat{\boldsymbol{\omega_b}}(t) - \boldsymbol{\omega_b}|, |\widehat{\boldsymbol{v_b}}(t) - \boldsymbol{v_b}| \leqslant R + C_2 T, \forall t \in [0,T],$$

where

$$C_2 = C_2(M, R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,1)}, D_*, \|(D, y)\|_{C^1([0,1]\times\mathbb{S}^2)}, k_y, k_D, \lambda_y, \lambda_D) > 0.$$

Thus if $T < R/C_2$, then $(\widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}})$ belongs to E_T. This ends Step 1 with $T^* := \min\{1, R/C_2\}$.

Step 2: Let us prove the existence of

$$T^{**} = T^{**}(M, R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,1)}, D_*, \|(y, D)\|_{C^1([0,1]\times\mathbb{S}^2)}, k_y, k_D, \lambda_y, \lambda_D) \in (0,1)$$

such that F_T is $(1/2)$-contractant on E_T for every $T < T^{**}$. Let $(\widehat{\boldsymbol{W_b}}^1, \widehat{\boldsymbol{V_b}}^1), (\widehat{\boldsymbol{W_b}}^2, \widehat{\boldsymbol{V_b}}^2) \in E_T$, Φ_t^1, Φ_t^2 be the associated flows on $\mathbb{S}^2$ and $(\widehat{Y}^1, \widehat{d}^1, \widehat{\boldsymbol{\omega_b}}^1, \widehat{\boldsymbol{v_b}}^1)$, $(\widehat{Y}^2, \widehat{d}^2, \widehat{\boldsymbol{\omega_b}}^2, \widehat{\boldsymbol{v_b}}^2)$ be the associated solutions of (5.38). The regularity of the flow of (5.39) with respect to the function $(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}})$ justifies the existence of a constant $C_3 = C_3(R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,T)}, D_*, \|D\|_{C^1([0,T]\times\mathbb{S}^2)}) > 0$ such that, for every $t \in [0,T]$,

$$\|\Phi_t^1 - \Phi_t^2\|_{C^1(\mathbb{S}^2)}, \|(\Phi_t^1)^{-1} - (\Phi_t^2)^{-1}\|_{C^1(\mathbb{S}^2)} \leqslant C_3 \|(\widehat{\boldsymbol{W_b}}^1 - \widehat{\boldsymbol{W_b}}^2, \widehat{\boldsymbol{V_b}}^1 - \widehat{\boldsymbol{V_b}}^2)\|_{L^\infty(0,T)}.$$

Using the regularity of y, $\overline{y}^0$, D, $\widehat{D}^0$ and bound (5.36), we deduce the existence of a constant $C_4 = C_4(M, R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,T)}, D_*, \|(y, D)\|_{C^1([0,T]\times\mathbb{S}^2)}, k_y, k_D) > 0$ such that

$$\|(\widehat{Y}^1 - \widehat{Y}^2, \widehat{d}^1 - \widehat{d}^2)(t)\|_{C^1(\mathbb{S}^2)} \leqslant C_4 \|(\widehat{\boldsymbol{W_b}}^1 - \widehat{\boldsymbol{W_b}}^2, \widehat{\boldsymbol{V_b}}^1 - \widehat{\boldsymbol{V_b}}^2)\|_{L^\infty(0,T)}, \forall t \in [0,T].$$

Thus, we deduce from (5.43) and (5.44) that

$$\|(\widehat{\boldsymbol{\omega_b}}^1 - \widehat{\boldsymbol{\omega_b}}^2, \widehat{\boldsymbol{v_b}}^1 - \widehat{\boldsymbol{v_b}}^2)\|_{L^\infty(0,T)} \leqslant C_5 T \|(\widehat{\boldsymbol{W_b}}^1 - \widehat{\boldsymbol{W_b}}^2, \widehat{\boldsymbol{V_b}}^1 - \widehat{\boldsymbol{V_b}}^2)\|_{L^\infty(0,T)}$$

for some constant C_5, that depends on the same quantitites as C_4. This gives the conclusion with $T^{**} = 1/(2C_5)$. □

5.5.2 Bounds on solutions

The goal of this section is the proof of the following result.

Proposition 6. *Let $R > 0$. There exists*

$$k_* = k_*(R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,1)}, D_*, \|(y, D)\|_{C^1([0,1]\times\mathbb{S}^2)}) > 0$$

such that for every $\widehat{\boldsymbol{\omega_b}}^0, \widehat{\boldsymbol{v_b}}^0 \in \mathbb{R}^3$ with (5.35), $k_y, k_D > k_*$, $T^* > 0$, if $(\widehat{y}, \widehat{D}, \widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}}) \in C^1([0,T^*]\times\mathbb{S}^2)^2 \times C^1([0,T^*],\mathbb{R}^3)^2$ is a solution of (5.24) on the time intervall $[0,T^*]$, then

$$\|(\widehat{y}, \widehat{D})(t)\|_{C^1(\mathbb{S}^2)} \leqslant \|(y, D)\|_{C^1(\mathbb{S}^2)} + 1\,, \quad \forall t \in [0,T^*].$$

The proof of this proposition relies on the following technical Lemma.

Lemma 5. *Let $M_0 > 0$. There exists $k_* = k_*(M_0) > 0$ such that*

– *for every $\mathbf{a} \in C^1([0,+\infty)\times\mathbb{S}^2,\mathbb{R}^3)$, $b \in C^1([0,+\infty)\times\mathbb{S}^2,\mathbb{R})$ such that*

$$\mathbf{a}(t,\boldsymbol{\eta})\cdot\boldsymbol{\eta} = 0\,, \quad \forall (t,\boldsymbol{\eta}) \in [0,+\infty)\times\mathbb{S}^2\,, \tag{5.46}$$

$$\|a(t,\boldsymbol{\eta})\|, \|b(t,\boldsymbol{\eta})\|, \left\|\frac{\partial \mathbf{a}}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right\|, \left\|\frac{\partial b}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right\| \leq M_0\,, \quad \forall (t,\boldsymbol{\eta}) \in [0,+\infty)\times\mathbb{S}^2\,, \tag{5.47}$$

– *for every $k > k_*$, $h^0 \in C^1(\mathbb{S}^2,\mathbb{R})$, $T^* > 0$ and $h \in C^1([0,T^*]\times\mathbb{S}^2,\mathbb{R})$ solution on $[0,T^*]$ of*

$$\begin{cases} \partial_t h(t,\boldsymbol{\eta}) = \frac{\partial h}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\cdot\mathbf{a}(t,\boldsymbol{\eta}) + b(t,\boldsymbol{\eta}) - kh(t,\boldsymbol{\eta})\,, \\ h(0,\boldsymbol{\eta}) = h^0(\boldsymbol{\eta})\,, \end{cases}$$

then,

$$\|h(t,.)\|_{C^1(\mathbb{S}^2)} \leqslant \|h^0\|_{C^1(\mathbb{S}^2)} + 1\,, \forall t \in [0,T^*].$$

Proof of Lemma 5: Let us consider the flow $\varphi(t,\boldsymbol{\eta})$ associated to the following equation

$$\begin{cases} \partial_t \varphi = -\mathbf{a}(t,\varphi) \\ \varphi(0,\boldsymbol{\eta}) = \boldsymbol{\eta}. \end{cases} \tag{5.48}$$

The ODE has a local (in time) solution for any $\boldsymbol{\eta} \in \mathbb{S}^2$ thanks to Cauchy-Lipschitz theorem because $\mathbf{a}$ is continuous in t and C^1 in $\boldsymbol{\eta}$. This solution lives on $\mathbb{S}^2$ thanks to assumption (5.46), thus no explosion is possible and $\varphi(t,\boldsymbol{\eta})$ is defined for every $(t,\boldsymbol{\eta}) \in [0,+\infty)\times\mathbb{S}^2$.

The new function:

$$z(t,\boldsymbol{\eta}) := h(t,\varphi(t,\boldsymbol{\eta})) \tag{5.49}$$

solves $\partial_t z(t,\boldsymbol{\eta}) = b(t,\varphi(t,\boldsymbol{\eta})) - kz$. Thus

$$z(t,\boldsymbol{\eta}) = e^{-kt}h_0(\boldsymbol{\eta}) + \int_0^t b(\tau,\varphi(\tau,\boldsymbol{\eta}))e^{k(\tau-t)}d\tau\,. \tag{5.50}$$

Step 1: Bound on h. For every $t \in [0,T^*]$, $\varphi(t,.)$ is a bijection of $\mathbb{S}^2$ thus

$$\begin{aligned}\|h(t,.)\|_{L^\infty(\mathbb{S}^2)} &= \|z(t,.)\|_{L^\infty(\mathbb{S}^2)}\\ &\leqslant \|h^0\|_{L^\infty(\mathbb{S}^2)}e^{-kt} + \int_0^t M_0 e^{-k(t-\tau)}d\tau \quad \text{by (5.50) and (5.47)}\\ &\leqslant \|h^0\|_{L^\infty(\mathbb{S}^2)} + \tfrac{M_0}{k}\\ &\leqslant \|h^0\|_{L^\infty(\mathbb{S}^2)} + 1 \quad \text{when } k > M_0\,.\end{aligned}$$

Step 2: Bound on $\partial h/\partial \boldsymbol{\eta}$. From (5.49) and (5.50), we deduce that

$$\begin{aligned}\tfrac{\partial h}{\partial \boldsymbol{\eta}}(t,\varphi(t,\boldsymbol{\eta})) &= \tfrac{\partial z}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\left(\tfrac{\partial \varphi}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right)^{-1}\\ &= e^{-kt}\tfrac{\partial h_0}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta})\left(\tfrac{\partial \varphi}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right)^{-1}\\ &\quad + \int_0^t e^{-k(t-\tau)}\tfrac{\partial b}{\partial \boldsymbol{\eta}}(\tau,\varphi(\tau,\boldsymbol{\eta}))\boldsymbol{\Sigma}(\tau,t,\boldsymbol{\eta})d\tau\end{aligned} \tag{5.51}$$

where

$$\boldsymbol{\Sigma}(\tau,t,\boldsymbol{\eta}) := \frac{\partial \varphi}{\partial \boldsymbol{\eta}}(\tau,\boldsymbol{\eta})\cdot\left(\frac{\partial \varphi}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right)^{-1}.$$

Note that

$$\begin{cases}\frac{\partial}{\partial t}\left(\frac{\partial \varphi}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right)^{-1} = \left(\frac{\partial \varphi}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right)^{-1}\frac{\partial \mathbf{a}}{\partial \boldsymbol{\eta}}(t,\varphi(t,\boldsymbol{\eta}))\,,\\ \left(\frac{\partial \varphi}{\partial \boldsymbol{\eta}}(0,\boldsymbol{\eta})\right)^{-1} = \mathbf{Id}\end{cases}$$

and

$$\begin{cases}\frac{\partial \boldsymbol{\Sigma}}{\partial \tau}(\tau,t,\boldsymbol{\eta}) = -\frac{\partial \mathbf{a}}{\partial \boldsymbol{\eta}}(\tau,\varphi(\tau,\boldsymbol{\eta}))\boldsymbol{\Sigma}(\tau,t,\boldsymbol{\eta})\,,\\ \boldsymbol{\Sigma}(t,t,\boldsymbol{\eta}) = \mathbf{Id}\,.\end{cases}$$

Thus, by Gronwall Lemma,

$$\begin{aligned}&\left\|\left(\frac{\partial \varphi}{\partial \boldsymbol{\eta}}(t,\boldsymbol{\eta})\right)^{-1}\right\| \leqslant e^{M_0 t}\,, \quad \forall (t,\boldsymbol{\eta}) \in [0,T^*]\times\mathbb{S}^2\,,\\ &\|\boldsymbol{\Sigma}(\tau,t,\boldsymbol{\eta})\| \leqslant e^{M_0|t-\tau|}\,, \quad \forall (\tau,t,\boldsymbol{\eta}) \in [0,T^*]^2\times\mathbb{S}^2\,.\end{aligned} \tag{5.52}$$

We deduce from (5.51) and (5.52) that

$$\begin{aligned}\|\tfrac{\partial h}{\partial \boldsymbol{\eta}}(t,\varphi(t,\boldsymbol{\eta}))\| &\leqslant e^{(M_0-k)t}\|\tfrac{\partial h_0}{\partial \boldsymbol{\eta}}\|_{L^\infty(\mathbb{S}^2)} + \int_0^t M_0 e^{(M_0-k)(t-\tau)}d\tau\\ &\leqslant \|\tfrac{\partial h_0}{\partial \boldsymbol{\eta}}\|_{L^\infty(\mathbb{S}^2)} + \tfrac{M_0}{k-M_0} \quad \text{when } k > M_0\\ &\leqslant \|\tfrac{\partial h_0}{\partial \boldsymbol{\eta}}\|_{L^\infty(\mathbb{S}^2)} + 1 \quad \text{when } k > 2M_0\,.\end{aligned}$$

This ends the proof of Lemma 5 with $k_* = 2M_0$. □

Proof of Proposition 6: We apply Lemma 5 to $\widetilde{y} = \widehat{y} - y$ and $\widetilde{D} = \widehat{D} - D$ with (see equations (5.19) and (5.20))

$$\mathbf{a}(t,\boldsymbol{\eta}) := -\boldsymbol{\eta}\times\left((\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}})(t) + \frac{1}{D(t,\boldsymbol{\eta})}\boldsymbol{\eta}\times(\boldsymbol{v_m} - \widehat{\boldsymbol{v_b}})(t)\right)$$

$$b_1(t,\boldsymbol{\eta}) := \nabla y(t,\boldsymbol{\eta}) \cdot \left(\boldsymbol{\eta} \times (\widehat{\boldsymbol{\omega_b}}(t) - \boldsymbol{\omega_b}) + \frac{1}{D(t,\boldsymbol{\eta})} \boldsymbol{\eta} \times (\widehat{\boldsymbol{v_b}}(t) - \boldsymbol{v_b}) \right)$$

$$b_2(t,\boldsymbol{\eta}) := \nabla D(t,\boldsymbol{\eta}) \cdot \left(\boldsymbol{\eta} \times (\widehat{\boldsymbol{\omega_b}}(t) - \boldsymbol{\omega_b}) + \frac{1}{D(t,\boldsymbol{\eta})} \boldsymbol{\eta} \times (\widehat{\boldsymbol{v_b}}(t) - \boldsymbol{v_b}) \right) + \widehat{\boldsymbol{v_b}}(t) \cdot \boldsymbol{\eta} - \boldsymbol{v_b} \cdot \boldsymbol{\eta} \,.$$

Note that $\|(\widehat{\boldsymbol{v_b}}, \widehat{\boldsymbol{\omega_b}})(t)\|$ is bounded uniformly with respect to $t \in [0, T^*]$ by a constant that depends only on R thanks to the decrease of the Lyapunov function V. Thus, the assumptions of Lemma 5 are satisfied with

$$M_0 = M_0(R, \|(\boldsymbol{\omega_m}, \boldsymbol{v_m})\|_{L^\infty(0,1)}, D_*, \|(y, D)\|_{C^1([0,1]\times\mathbb{S}^2)}) > 0 \,.$$

Therefore, there exists a constant $k_* > 0$ (that depends on the same quantities as M_0) such that, for every $k_y, k_D > k_*$ then

$$\|(\widehat{y} - y, \widehat{D} - D)(t)\|_{C^1(\mathbb{S}^2)} \leqslant 1 \,, \quad \forall t \in [0, T^*] \,,$$

which gives the conclusion. □

5.5.3 Global solutions

The goal of this section is the proof of Proposition 2 thanks to Propositions 5 and 6.
Let $\widehat{\boldsymbol{\omega_b}}^0, \widehat{\boldsymbol{v_b}}^0 \in \mathbb{R}^3$ and $R := \|(\widehat{\boldsymbol{\omega_b}}^0 - \boldsymbol{\omega_b}, \widehat{\boldsymbol{v_b}}^0 - \boldsymbol{v_b})\|$. We define

$$M := \|(y, D)\|_{C^1(\mathbb{S}^2)} + 1 \,.$$

By Proposition 5, there exists a time $T = T(M, R) > 0$ and a unique local solution of (5.24) defined on $[0, T]$. By Proposition 6, we have

$$\|(\widehat{y}, \widehat{D})(T)\|_{C^1(\mathbb{S}^2)} \leqslant M \,.$$

By decreasing of the Lyapunov function, we have

$$\|(\widehat{\boldsymbol{\omega_b}} - \boldsymbol{\omega_b}, \widehat{\boldsymbol{v_b}} - \boldsymbol{v_b})(T)\| \leqslant \sqrt{2V(T)} \leqslant \sqrt{2V(0)} = \|(\widehat{\boldsymbol{\omega_b}}^0 - \boldsymbol{\omega_b}, \widehat{\boldsymbol{v_b}}^0 - \boldsymbol{v_b})\| = R$$

Thus we can apply Proposition 5 with initial condition at $t = T$ and we get a solution defined on $[0, 2T]$. By iterating this argument, we get a solution defined for every $t \in [0, +\infty)$. □

5.6 Continuity of the flow

The goal of this section is to prove Proposition 4. We use the same notations as in the proof of Proposition 5.

Let $R \geqslant \|(\widehat{\boldsymbol{\omega_b}}^n(0) - \boldsymbol{\omega_b}, \widehat{\boldsymbol{v_b}}^n(0) - \boldsymbol{v_b})\|$ for every $n \in \mathbb{N}$. Thanks to the convergences in assumption, there exists $T_1 > 0$ such that the maps $(F^n_{T_1})_{n\in\mathbb{N}}$ and F_{T_1} associated to $(\boldsymbol{v}^n, \boldsymbol{\omega}^n, y^n, D^n)_{n\in\mathbb{N}}$ and $(\boldsymbol{v}, \boldsymbol{\omega}, y, D)$ are $(1/2)$-contractions of the same space E_{T_1}.

Step 1: We prove that, for every $(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}}) \in E_{T_1}$ then

$$\|(F^n_{T_1} - F_{T_1})(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}})\|_{L^\infty(0,T_1)} \to 0 \text{ when } n \to +\infty.$$

Let $(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}}) \in E_{T_1}$. The first component of $(F^n_{T_1} - F_{T_1})(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}})$ is

$$(F^n_{T_1} - F_{T_1})(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}})^{(1)}(t) = (\widehat{\boldsymbol{\omega_b}}^n - \widehat{\boldsymbol{\omega_b}})(0) + A_n(t) + B_n(t)$$

where, for every $t \in [0, T_1]$ (integrations by part)

$$A_n(t) = -\int_0^t \int_{\mathbb{S}^2} \lambda_y \Big((\widehat{Y}^n - y^n - \widehat{Y} + y)(\nabla \widehat{Y}^n \times \boldsymbol{\eta}) - (\widehat{Y}^n - \widehat{Y})\nabla(\widehat{Y} - y) \times \boldsymbol{\eta} \Big) d\sigma_\eta d\tau,$$

$$B_n(t) = -\int_0^t \int_{\mathbb{S}^2} \lambda_D \Big((\widehat{d}^n - D^n - \widehat{d} + D)(\nabla \widehat{d}^n \times \boldsymbol{\eta}) - (\widehat{d}^n - \widehat{d})\nabla(\widehat{d} - D) \times \boldsymbol{\eta} \Big) d\sigma_\eta d\tau.$$

For every $t \in [0, T_1]$, we have

$$\begin{aligned} |A_n(t)| \leqslant \lambda_y T_1 C \Big(& \|\widehat{Y}^n - y^n - \widehat{Y} + y\|_{L^\infty((0,T_1)\times\mathbb{S}^2)} \|\nabla \widehat{Y}^n\|_{L^\infty((0,T_1)\times\mathbb{S}^2)} \\ & + \|\nabla(\widehat{Y} - y)\|_{L^\infty((0,T_1)\times\mathbb{S}^2)} \|\widehat{Y}^n - \widehat{Y}\|_{L^\infty((0,T_1)\times\mathbb{S}^2)} \Big). \end{aligned} \tag{5.53}$$

Thanks to (5.25), we also have $\Phi^n_t \to \Phi_t$ in $C^0([0,T_1]\times\mathbb{S}^2, \mathbb{S}^2)$ (consequence of Gronwall lemma). From the explicit expressions of $\widehat{Y}^n, \widehat{Y}, \widehat{d}^n, \widehat{d}$ (see (5.41) and (5.42)) and the convergence (5.27), we deduce that

$$(\widehat{Y}^n, \widehat{d}^n) \to (\widehat{Y}, \widehat{d}) \quad \text{ in } C^0([0,T_1]\times\mathbb{S}^2, \mathbb{R})^2. \tag{5.54}$$

Moreover, thanks to (5.26), the quantities $\|\nabla \widehat{d}^n\|_{L^\infty((0,T_1)\times\mathbb{S}^2)}, \|\nabla \widehat{Y}^n\|_{L^\infty((0,T_1)\times\mathbb{S}^2)}$ are uniformly bounded. Thus, the right hand side of (5.53) converges to zero as $n \to +\infty$. We may work similarly to prove that $B_n(t)$ and $(F^n_{T_1} - F_{T_1})(\widehat{\boldsymbol{W_b}}, \widehat{\boldsymbol{V_b}})^{(2)}(t)$ converge to zero uniformly with respect to $t \in [0, T_1]$. This ends the proof of the first step.

Step 2: Conclusion. We have

$$\begin{aligned} & \|(\widehat{\boldsymbol{\omega_b}}^n - \widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}}^n - \widehat{\boldsymbol{v_b}})\|_{L^\infty(0,T_1)} \\ = \; & \|F^n_{T_1}(\widehat{\boldsymbol{\omega_b}}^n, \widehat{\boldsymbol{v_b}}^n) - F_{T_1}(\widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}})\|_{L^\infty(0,T_1)} \\ \leqslant \; & \|F^n_{T_1}(\widehat{\boldsymbol{\omega_b}}^n, \widehat{\boldsymbol{v_b}}^n) - F^n_{T_1}(\widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}})\|_{L^\infty(0,T_1)} + \|(F^n_{T_1} - F_{T_1})(\widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}})\|_{L^\infty(0,T_1)} \end{aligned} \tag{5.55}$$

and $F^n_{T_1}$ is a $(1/2)$-contraction of E_{T_1} thus

$$\|(\widehat{\boldsymbol{\omega_b}}^n - \widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}}^n - \widehat{\boldsymbol{v_b}})\|_{L^\infty(0,T_1)} \leqslant 2\|(F^n_{T_1} - F_{T_1})(\widehat{\boldsymbol{\omega_b}}, \widehat{\boldsymbol{v_b}})\|_{L^\infty(0,T_1)} \to 0$$

thanks to the first step. Iterating this argument on a finite number of intervals $[T_1, 2T_1]$, $[2T_1, 3T_1]$, etc, we conclude that (5.28) holds. Let φ_t be the flow associated to the equation

$$\frac{d\boldsymbol{\eta}}{dt} = \boldsymbol{\eta} \times \Big(\boldsymbol{\omega_m}(t) - \widehat{\boldsymbol{\omega_b}}(t) + \frac{1}{D(t,\boldsymbol{\eta})} \boldsymbol{\eta} \times [\boldsymbol{v_m}(t) - \widehat{\boldsymbol{v_b}}(t)] \Big)$$

and φ^n_t similarly associated to $(\widehat{\boldsymbol{\omega_b}}^n, \widehat{\boldsymbol{v_b}}^n)$. Then, by (5.28), $\varphi^n_t \to \varphi_t$ in $C^0([0,T]\times\mathbb{S}^2, \mathbb{S}^2)$. Thanks to the explicit expressions of $(\widehat{y}^n \widehat{D}^n)$ and $(\widehat{y}, \widehat{D})$ in terms of φ^n_t and φ_t (obtained by the characteristic method, as (5.41) and (5.42)), we deduce that (5.29) holds. This ends the proof of Proposition 4. □

Chapter 6

The geometrical flow method for velocity estimation

Estimation de vitesses linéaire et angulaire à partir d'images de profondeur

Dans ce chapitre est décrite une nouvelle méthode d'odométrie, basée sur le traitement d'images de profondeur seules, considérées comme un champ scalaire dense. L'adaptation de cette méthode à des images de profondeur réelles associées à un modèle de caméra pinhole est détaillée. Des résultats préliminaires d'implémentation de notre méthode sur des données réelles est prometteuse : un indice de confiance peut être associé à la vitesse calculée, ce qui est très utile en fusion de données lorsque la redondance de l'information est exploitée. Ces résultats ont été présentés dans [Zarrouati-Vissière et al., 2013a] et ont été l'objet d'une demande de brevet [Zarrouati-Vissière et al., 2013b].

In this chapter, a novel odometry method based on the sole processing of depth images as a dense scalar field is described. Adaptation of this method to real depth images associated to a pinhole camera model is detailed. Preliminary results of implementation of our method on real data seem promising: a confidence index can be assigned to the resulting velocity, which is of great help in data fusion where (partial) redundancy of information is involved. These results were presented in [Zarrouati-Vissière et al., 2013a] and are the subject of a filed patent application [Zarrouati-Vissière et al., 2013b].

6.1 The geometrical flow as a conservation law

Recall from chapter 1.3 the PDE satisfied by the depth D

$$\partial_t D = -\nabla D \cdot (\boldsymbol{\eta} \times (\boldsymbol{\omega} + \frac{1}{D}\boldsymbol{\eta} \times \boldsymbol{v})) - \boldsymbol{v} \cdot \boldsymbol{\eta}, \quad \text{see equation (1.17)}$$

The equation is called in this dissertation the geometrical flow equation by analogy with the optical flow equation (see equation (1.16)). We will work on this PDE, with the help of two

differential calculus relations, recalled in the following. These relations hold on the unit sphere $\mathbb{S}^2$, for $\boldsymbol{\eta} \in \mathbb{S}^2$ and $\mathbf{P} \in \mathbb{R}^3$ any vector independent of $\boldsymbol{\eta}$:

$$\nabla \cdot (\boldsymbol{\eta} \times \mathbf{P}) = 0, \quad \text{see equation (A.7)}$$

$$\nabla \cdot (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \mathbf{P})) = -2\boldsymbol{\eta} \cdot \mathbf{P}, \quad \text{see equation (A.4)}$$

where $\nabla\cdot$ denotes the divergence operator on $\mathbb{S}^2$ as a Riemaniann manifold.

From (1.17), the partial time-derivative of D^3 is computed:

$$\partial_t D^3 = -\nabla D^3 \cdot (\boldsymbol{\eta} \times \boldsymbol{\omega}) - \frac{3}{2}\nabla D^2 \cdot (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}))) - 3D^2 \boldsymbol{v} \cdot \boldsymbol{\eta} \tag{6.1}$$

Then, using (A.7):

$$\nabla D^3 \cdot (\boldsymbol{\eta} \times \boldsymbol{\omega}) = \nabla \cdot (D^3 \boldsymbol{\eta} \times \boldsymbol{\omega})$$

and using (A.4):

$$\begin{aligned} \nabla D^2 \cdot (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}))) &= \nabla \cdot (D^2 \boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}))) - D^2 \nabla \cdot (\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}))) \\ &= \nabla \cdot (D^2 \boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}))) + 2D^2 \boldsymbol{v} \cdot \boldsymbol{\eta} \end{aligned}$$

Thus, (6.1) can be reformulated as

$$\partial_t D^3 = -\nabla \cdot (D^3 \boldsymbol{\eta} \times \boldsymbol{\omega}) - \frac{3}{2}\nabla \cdot (D^2 \boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v})))$$

Which means that the partial time-derivative of D^3 can be completely expressed as the divergence of a vector field. Physically, the integral formulation of this point-wise equation is equivalent to the basic assumption that the volume of the environment in which the camera is moving is constant, which is a conservation law. This does not seem like a huge discovery. However, this point-wise formulation can be expressed as a local balanced equation, which is, to our knowledge, new.

Integrating this equation on a fixed portion Ω of $\mathbb{S}^2$ with boundary $\partial\Omega$, Stokes's theorem yields

$$\partial_t \int_\Omega D^3 d\sigma_\eta = -\oint_{\partial\Omega} (D^3 \boldsymbol{\eta} \times \boldsymbol{\omega}) \cdot \vec{\mathbf{n}}\, dl_\eta - \frac{3}{2}\oint_{\partial\Omega} (D^2 \boldsymbol{\eta} \times (\boldsymbol{\eta} \times \boldsymbol{v}))) \cdot \vec{\mathbf{n}}\, dl_\eta$$

where $d\sigma_\eta$ is an infinitesimal surface element of $\mathbb{S}^2$, dl_η is an infinitesimal length element, and $\vec{\mathbf{n}}$ is the outwards-pointing normal vector of $\partial\Omega$.

This equation is rearranged using the double product properties

$$\partial_t \int_\Omega D^3 d\sigma_\eta = \left(\oint_{\partial\Omega} (D^3 \boldsymbol{\eta} \times \vec{\mathbf{n}}) dl_\eta\right) \cdot \boldsymbol{\omega} + \left(\frac{3}{2}\oint_{\partial\Omega} (D^2 \boldsymbol{\eta} \times (\vec{\mathbf{n}} \times \boldsymbol{\eta})) dl_\eta\right) \cdot \boldsymbol{v} \tag{6.2}$$

Which is simplified, noticing that $\vec{\mathbf{n}}$, as the normal vector of boundary $\partial\Omega$ is an element of T_η (the tangent set of $\mathbb{S}^2$ at point $\boldsymbol{\eta}$), which implies $\boldsymbol{\eta} \times (\vec{\mathbf{n}} \times \boldsymbol{\eta}) = \vec{\mathbf{n}}$. Finally,

$$\partial_t \int_\Omega D^3 d\sigma_\eta = \left(\oint_{\partial\Omega} (D^3 \boldsymbol{\eta} \times \vec{\mathbf{n}}) dl_\eta\right) \cdot \boldsymbol{\omega} + \left(\frac{3}{2}\oint_{\partial\Omega} D^2\, \vec{\mathbf{n}}\, dl_\eta\right) \cdot \boldsymbol{v} \tag{6.3}$$

Formally, given two consecutive depth images and a fixed domain Ω of $\mathbb{S}^2$, the left term of (6.3) can be computed by discrete approximation of the time-derivative. Computation of $\left(\oint_{\partial\Omega}(D^3\boldsymbol{\eta}\times\vec{\mathbf{n}})dl_\eta\right)$ and $\left(\frac{3}{2}\oint_{\partial\Omega}D^2\,\vec{\mathbf{n}}\,dl_\eta\right)$ yields a scalar equation, linear with respect to $\boldsymbol{v}$ and $\boldsymbol{\omega}$. To each domain Ω corresponds such an equation. Six linearly independent equations yield the reconstruction of the linear and angular displacements $\boldsymbol{v}$ and $\boldsymbol{\omega}$ that occurred between the two depth images.

6.2 Implementation of the method for a pinhole camera model

To adapt this method to real images, it is necessary to express the integral terms of (6.3) with respect to the real coordinates of pixels in the image. As already mentioned in Chapter 1, this requires to know the calibration model and parameters of the camera. In this section, we consider a regular pinhole camera model, where distortion is neglected. The image is a rectangular matrix, whose size defines the resolution of the image: we will denote it (r_x, r_y). A pixel's position is denoted $(i,j) \in (1,r_x)\times(1,r_y)$. Using the notations of Chapter 1, the unitary vector $\boldsymbol{\eta}$ corresponding to the pixel's position (i,j) is given by

$$\boldsymbol{\eta} = \frac{1}{\sqrt{(\frac{i-x_0}{fr_x})^2+(\frac{j-y_0}{fr_y})^2+1}}\begin{pmatrix}\frac{i-x_0}{fr_x}\\ \frac{j-y_0}{fr_y}\\ 1\end{pmatrix}$$

For sake of simplicity, we define $z_1 := \frac{i-x_0}{fr_x}$ and $z_2 := \frac{j-y_0}{fr_y}$, so that

$$\boldsymbol{\eta} = \frac{1}{\sqrt{z_1^2+z_2^2+1}}(z_1,z_2,1)^T.$$

The implementation of the proposed method for a real image requires the choice of domains Ω on $\mathbb{S}^2$: a natural choice is to ensure that the domains, when reprojected on the plane image, are identified with rectangles (the smallest possible rectangle is a single pixel). In particular, the infinitesimal surface element $d\sigma_\eta$ is, reprojected on $\mathbb{S}^2$, the area corresponding to a pixel: it naturally depends on the position of this pixel in the image. The value of this area is established by the following consideration: it is simply the Euclidean norm of the vector $d\boldsymbol{\eta}_1 \times d\boldsymbol{\eta}_2$, where vectors $d\boldsymbol{\eta}_1$ and $d\boldsymbol{\eta}_2$ are infinitesimal displacements along each side of the pixel:

$$d\boldsymbol{\eta}_1 := \frac{\partial\boldsymbol{\eta}}{\partial z_1}dz_1, \quad \text{and } d\boldsymbol{\eta}_2 := \frac{\partial\boldsymbol{\eta}}{\partial z_2}dz_2$$

where dz_1 et dz_2 are the length and the width of a pixel.

Computation yields

$$\frac{\partial\boldsymbol{\eta}}{\partial z_1} = \frac{1}{(1+z_1^2+z_2^2)^{3/2}}\begin{pmatrix}1+z_2^2\\ -z_1z_2\\ -z_1\end{pmatrix}, \quad \frac{\partial\boldsymbol{\eta}}{\partial z_2} = \frac{1}{(1+z_1^2+z_2^2)^{3/2}}\begin{pmatrix}-z_1z_2\\ 1+z_1^2\\ -z_2\end{pmatrix}$$

hence

$$d\sigma_\eta = \frac{dz_1 dz_2}{(1+z_1^2+z_2^2)^{3/2}}.$$

Similarly, an infinitesimal length element dl_η corresponds to the norm of $d\boldsymbol{\eta}_1$ or $d\boldsymbol{\eta}_2$, depending on the direction of displacement along $\partial\Omega$: horizontally corresponds to dl_{η_1} and vertically to dl_{η_2}. This yields

$$dl_{\eta_1} = \frac{\sqrt{1+z_2^2}}{1+z_1^2+z_2^2}dz_1, \quad dl_{\eta_2} = \frac{\sqrt{1+z_1^2}}{1+z_1^2+z_2^2}dz_2.$$

The outwards-pointing normal $\vec{\mathbf{n}}$ along $\partial\Omega$ corresponds to the unit vector of T_η whose direction is given by $\pm d\boldsymbol{\eta}_i \times \boldsymbol{\eta}$ $(i = 1, 2)$: the sign depends on the direction in which the path is run, and $d\boldsymbol{\eta}_i$ is only the orientation. Thus

$$\vec{\mathbf{n}}_{+1} = \frac{1}{\sqrt{1+z_2^2}}\begin{pmatrix}0\\-1\\z_2\end{pmatrix}, \quad \vec{\mathbf{n}}_{-1} = \frac{1}{\sqrt{1+z_2^2}}\begin{pmatrix}0\\1\\-z_2\end{pmatrix},$$

$$\vec{\mathbf{n}}_{+2} = \frac{1}{\sqrt{1+z_1^2}}\begin{pmatrix}1\\0\\-z_1\end{pmatrix}, \quad \vec{\mathbf{n}}_{-2} = \frac{1}{\sqrt{1+z_1^2}}\begin{pmatrix}-1\\0\\z_1\end{pmatrix}$$

The previous computations are aggregated to finally obtain the expressions of $\boldsymbol{\eta}\times\vec{\mathbf{n}}\,dl_\eta$ and $\vec{\mathbf{n}}\,dl_\eta$ on any of the sides of a given rectangle, which is parameterized by the side coordinates $(\underline{z_1}, \overline{z_1}, \underline{z_2}, \overline{z_2})$. These expressions are gathered in the following table:

z_1	z_2	$(\boldsymbol{\eta}\times\vec{\mathbf{n}})dl_\eta$	$\vec{\mathbf{n}}\,dl_\eta$
$\in(\underline{z_1}, \overline{z_1})$	$\underline{z_2}$	$\frac{dz_1}{(1+z_1^2+\underline{z_2}^2)^{3/2}}\begin{pmatrix}1+\underline{z_2}^2\\-z_1\underline{z_2}\\-z_1\end{pmatrix}$	$\frac{dz_1}{1+z_1^2+\underline{z_2}^2}\begin{pmatrix}0\\-1\\\underline{z_2}\end{pmatrix}$
$\overline{z_1}$	$\in(\underline{z_2}, \overline{z_2})$	$\frac{dz_2}{(1+\overline{z_1}^2+z_2^2)^{3/2}}\begin{pmatrix}-\overline{z_1}z_2\\(1+\overline{z_1}^2)\\-z_2\end{pmatrix}$	$\frac{dz_2}{1+\overline{z_1}^2+z_2^2}\begin{pmatrix}1\\0\\-\overline{z_1}\end{pmatrix}$
$\in(\overline{z_1}, \underline{z_1})$	$\overline{z_2}$	$\frac{dz_1}{(1+z_1^2+\overline{z_2}^2)^{3/2}}\begin{pmatrix}-(1+\overline{z_2}^2)\\z_1\overline{z_2}\\z_1\end{pmatrix}$	$\frac{dz_1}{1+z_1^2+\overline{z_2}^2}\begin{pmatrix}0\\1\\-\overline{z_2}\end{pmatrix}$
$\underline{z_1}$	$\in(\overline{z_2}, \underline{z_2})$	$\frac{dz_2}{(1+\underline{z_1}^2+z_2^2)^{3/2}}\begin{pmatrix}\underline{z_1}z_2\\-(1+\underline{z_1}^2)\\z_2\end{pmatrix}$	$\frac{dz_2}{1+\underline{z_1}^2+z_2^2}\begin{pmatrix}-1\\0\\\underline{z_1}\end{pmatrix}$

6.3 Implementation of the method for a pinhole camera model including radial distortion

In this section, we use the notations of Chapter 1. Similarly to the previous implementation, the first step is to express the unitary vector $\boldsymbol{\eta}$ with respect to pixel coordinates (i, j). The fish-eye distortion model described by (1.6) yields

$$\boldsymbol{\eta} = \frac{1}{\sqrt{\frac{\tan(rw)^2}{(2\tan(w/2))^2} + 1}} \begin{pmatrix} z_1 \frac{\tan(rw)}{2r\tan(w/2)} \\ z_2 \frac{\tan(rw)}{2r\tan(w/2)} \\ 1 \end{pmatrix}$$

where $r := \sqrt{z_1^2 + z_2^2}$ (same definitions for z_1 and z_2 than in the previous section) and w is the distortion coefficient.

Then $d\sigma_\eta$ is computed using the same method than in the previous section:

$$d\sigma_\eta = 2w \frac{\tan(rw)}{\tan(w/2)^2} \frac{(1 + \tan(rw)^2)}{r(4 + \frac{\tan(rw)^2}{\tan(w/2)^2})^{3/2}} dz_1 dz_2.$$

Finally, the expressions of $\boldsymbol{\eta} \times \vec{\mathbf{n}}\, dl_\eta$ and $\vec{\mathbf{n}}\, dl_\eta$ on the four sides of a given rectangle are computed and gathered in the following table. For sake of concision of the relations, we use the following notations:

$$C_w = \cos(w/2), \quad S_w = \sin(w/2), \quad C_r = \cos(wr), \quad S_r = \sin(wr).$$

$z_1 \in (\underline{z_1}, \overline{z_1})$	$z_2 = \underline{z_2}$	$(\boldsymbol{\eta}\times \vec{\mathbf{n}})dl_\eta = \frac{dz_1 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)^{3/2}} \begin{pmatrix} (\underline{z_2}^2 S_r(4S_w^2C_r^2 + C_w^2S_r^2) + 4wrz_1^2S_w^2C_r) \\ -z_1\underline{z_2}(C_w^2S_r^3 + 4S_w^2C_r(C_rS_r - wr)) \\ -2z_1r^2wC_wS_rS_w \end{pmatrix}$
$z_1 \in (\underline{z_1}, \overline{z_1})$	$z_2 = \underline{z_2}$	$\vec{\mathbf{n}}\, dl_\eta = \frac{dz_1 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)} \begin{pmatrix} -2z_1\underline{z_2}S_w(S_rC_r - wr) \\ -2S_w(\underline{z_2}^2 S_rC_r + wz_1^2r) \\ \underline{z_2}rC_wS_r^2 \end{pmatrix}$
$z_1 = \overline{z_1}$	$z_2 \in (\underline{z_2}, \overline{z_2})$	$(\boldsymbol{\eta}\times \vec{\mathbf{n}})dl_\eta = \frac{dz_2 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)^{3/2}} \begin{pmatrix} -\overline{z_1}z_2(C_w^2S_r^3 + 4S_w^2C_r(C_rS_r - wr)) \\ (\overline{z_1}^2 S_r(C_w^2S_r^2 + 4S_w^2C_r^2) + 4wrz_2^2S_w^2C_r) \\ -2r^2wz_2C_wS_rS_w \end{pmatrix}$
$z_1 = \overline{z_1}$	$z_2 \in (\underline{z_2}, \overline{z_2})$	$\vec{\mathbf{n}}\, dl_\eta = \frac{dz_2 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)} \begin{pmatrix} 2S_w(\overline{z_1}^2 S_rC_r + wz_2^2r) \\ 2\overline{z_1}z_2S_w(S_rC_r - wr) \\ -\overline{z_1}rC_wS_r^2 \end{pmatrix}$
$z_1 \in (\overline{z_1}, \underline{z_1})$	$z_2 = \overline{z_2}$	$(\boldsymbol{\eta}\times \vec{\mathbf{n}})dl_\eta = \frac{dz_1 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)^{3/2}} \begin{pmatrix} -(\overline{z_2}^2 S_r(4S_w^2C_r^2 + C_w^2S_r^2) + 4wrz1^2S_w^2C_r) \\ z_1\overline{z_2}(C_w^2S_r^3 + 4S_w^2C_r(C_rS_r - wr)) \\ 2z_1r^2wC_wS_rS_w \end{pmatrix}$
$z_1 \in (\overline{z_1}, \underline{z_1})$	$z_2 = \overline{z_2}$	$\vec{\mathbf{n}}\, dl_\eta = \frac{dz_1 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)} \begin{pmatrix} 2z_1\overline{z_2}S_w(S_rC_r - wr) \\ 2S_w(\overline{z_2}^2 S_rC_r + wz_1^2r) \\ -\overline{z_2}rC_wS_r^2 \end{pmatrix}$
$z_1 = \underline{z_1}$	$z_2 \in (\overline{z_2}, \underline{z_2})$	$(\boldsymbol{\eta}\times \vec{\mathbf{n}})dl_\eta = \frac{dz_2 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)^{3/2}} \begin{pmatrix} z_1z_2(C_w^2S_r^3 + 4S_w^2C_r(C_rS_r - wr)) \\ -(\underline{z_1}^2 S_r(C_w^2S_r^2 + 4S_w^2C_r^2) + 4wrz_2^2S_w^2C_r) \\ 2r^2wz_2C_wS_rS_w \end{pmatrix}$
$z_1 = \underline{z_1}$	$z_2 \in (\overline{z_2}, \underline{z_2})$	$\vec{\mathbf{n}}\, dl_\eta = \frac{dz_2 C_w}{r^3(4S_w^2C_r^2+C_w^2S_r^2)} \begin{pmatrix} -2S_w(\underline{z_1}^2 S_rC_r + wz_2^2r) \\ -2\underline{z_1}z_2S_w(S_rC_r - wr) \\ \underline{z_1}rC_wS_r^2 \end{pmatrix}$

6.4 Experiments on real data

This method is tested on real depth images. The system of acquisition is the same than in Section 5.4.4. For the sequence of images tested, the system lies in the beginning on a table, in front of a laser telemeter, which measures the distance towards a plane surface of the system. The system is submitted to rotations and translations, in the following order:

- rotation about the vertical axis (y-axis in the camera reference frame $\mathcal{R}_{cam}$),
- rotation about the optical axis (z-axis in the camera reference frame $\mathcal{R}_{cam}$),
- rotation about the lateral axis (x-axis in the camera reference frame $\mathcal{R}_{cam}$),
- translation along the optical axis (the laser telemeter measurements provide a reference for this linear velocity)
- rapid rotation about the vertical axis (y-axis in the camera reference frame $\mathcal{R}_{cam}$).

Figure 6.1 illustrates the setup used for this experiment. The method presented in the previous

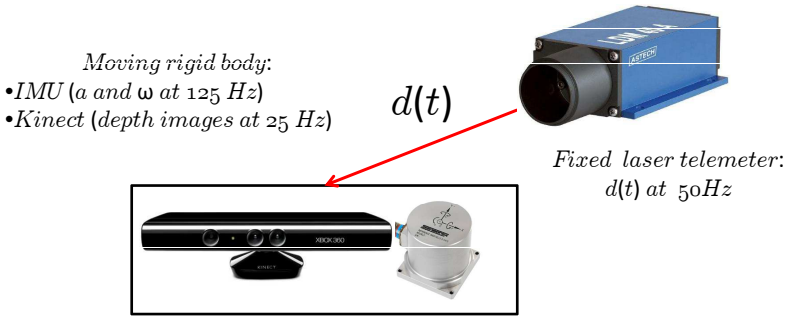


Figure 6.1: Setup of this experiment. The system lies on a table, in front of a laser telemeter, which measures the distance towards a plane surface of the system. The Kinect device acquires depth images, and angular velocities of the system are provided by embedded gyrometers.

sections is implemented, and the results are plotted in the following figures. The linear velocities expressed in the reference camera frame $\mathcal{R}_{cam}$ are represented in Figure 6.2. The linear velocity along the optical axis is plotted along with the velocity computed by time-differentiation and low-pass filtering of distance measurements from the laser telemeter in the third subplot. The magnitude of the difference does not exceed $0.1 m.s^{-1}$. This enables to validate the functioning of this method, although the results are noisy, especially in the x and y directions.

The angular velocities about the x, y and z axis are plotted along with gyrometers measurements in Figure 6.3. Again, both measurements are perfectly consistent, with the sole exception that the gyrometers measurements are biased, and the geometrical flow measurements are noisy. The magnitude of the difference does not exceed $10^{-3} rad.s^{-1}$.

From these preliminary results arise several questions: are any environments more favorable than others to this method? Can we describe them? Can we make sure that the choice of the domains Ω_i is the most efficient? Can we assign a confidence index to the measured velocities?

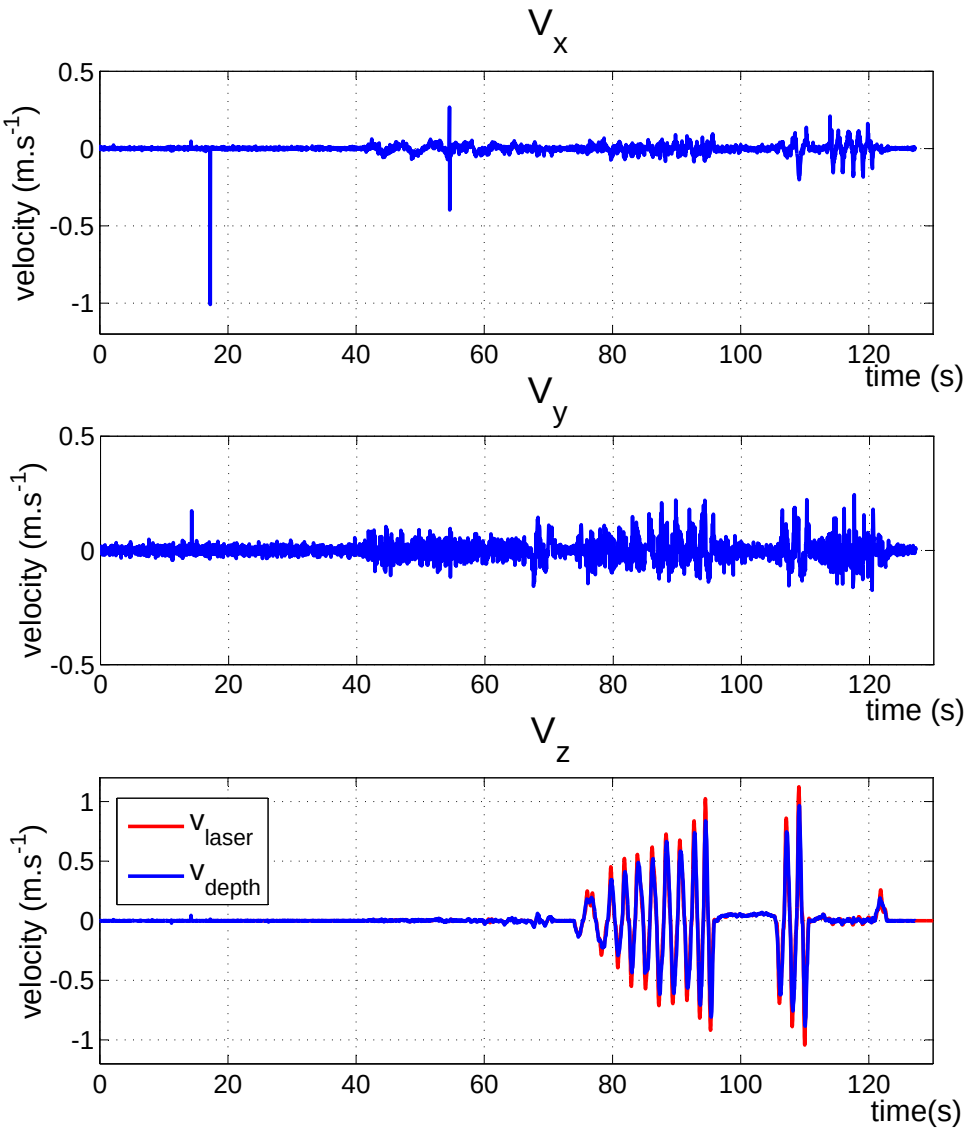


Figure 6.2: Linear velocities measured using the geometrical flow method. In the third subplot, linear velocity along the optical axis is plotted along with linear velocity computed from the laser telemeter measurements (red).

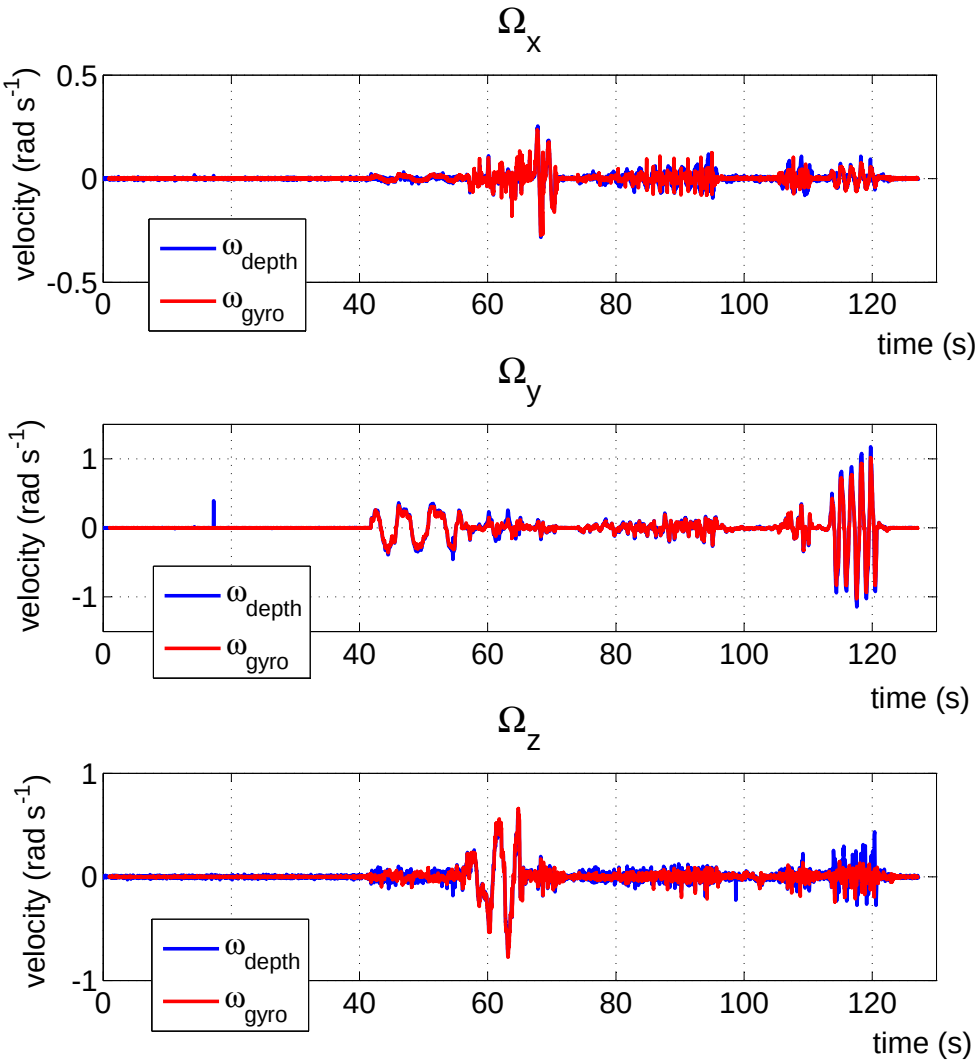


Figure 6.3: Angular velocities: measurements using the geometrical flow method (blue) and gyrometers measurements (red). Both measurements are perfectly consistent, with the sole exception that the gyrometers measurements are biased, and the geometrical flow measurements are noisy. The magnitude of the difference does not exceed $10^{-3} rad.s^{-1}$.

To the questions concerning the influence of the environment, one can immediately answer that the most favorable environment is the one that satisfies the hypothesis that enabled to formulate the geometrical flow equation in Section 1.3.1, especially hypothesis (H1). However, this is not sufficient to ensure that the 6-DOF motion can be fully reconstructed. In fact, the possibility to reconstruct the motion of the camera is ensured by the same condition that was pointed in Chapter 5 for observability of biases on velocity (see Theorem 3): the environment should not admit a rotation axis. This is proved working by contradiction and using usual linear algebra tools and Stokes formula.

This yields an issue for implementation of this method on real depth images: given that the actual field of view of the camera is much narrower than the entire field of view of the unit sphere in the ideal spherical model, a non-cylindrical symmetry should be visible even from a small visible part of the scene. For a given environment, ensuring that, for any point of view of the camera, the visible part of the scene won't admit a rotation axis is a difficult task, which requires a precise work on the geometry of the environment: in our opinion, this can not be achieved in real-time.

Similarly, the choice of the most efficient domains is a critical task: ideally, we would like to be able to choose only six domains that would enable to invert the linear system of six equations written in the form of equation (6.3). Unfortunately, even if a precise characterization of the environment ensures that 6 such domains exist, their choice require a thorough analysis of the geometry of the scene.

Fortunately, it is possible, once the domains are chosen and the (rectangular) linear system is (pseudo) inverted, to know the quality of the measurements: the covariance of the result is directly encrypted in the pseudo-inverse of the linear system weighted by the residuals of the linear system. This a precious information for data fusion such as Kalman filtering.

It is rather difficult to compare this novel method to other known techniques. The closest algorithm that can perform 6-DOF motion estimation from depth data is (to our knowledge) the Iterative Closest Point (ICP) algorithm, which is used in the seminal work known as Kinect fusion [Izadi et al., 2011]. ICP performs an estimation of displacement by iterative alignment of two point clouds, without any explicit point correspondence ([Rusinkiewicz and Levoy, 2001]). Variants have been studied to take into account surface normals or elastic deformations of point clouds (e.g for medical applications). It is mostly used for depth data merging. This algorithm has two main limitations: its domain of convergence is very local, and it requires an important processing time. By comparison, our proposed method runs at 140 Hz on a commercial laptop purchased in 2011 including an Intel Core i5-2520M CPU @ 2.50 GHz processor. To be specific, depending on the number of domains used to form the rectangular linear system to be inverted, the processing time can vary (see Table 6.1 for a comparison).

See for example [Gold et al., 1998], [Zhang, 1994], [Li and Hartley, 2007] for more details on ICP variants. Dense depth data can also be used as an aid for SLAM techniques: for example, it is well adapted to plane extraction [Georgiev et al., 2011, Dube and Zell, 2011], which can then be used in SLAM algorithms [Taguchi et al., 2012]. An interesting approach using RGBD images and presented in [Steinbrucker et al., 2011] is the minimization of optical flow constraint where 6-DOF motion and depth are made explicit. It does require RGB images and is highly

Number of domains considered	Processing time (ms)
40	6.5
100	8.8
200	15.3
500	38.0
1000	76.6

Table 6.1: Comparison of different processing times depending on the number of domains considered for the geometrical flow velocity estimation.

sensitive to noise due to the explicit use of image gradients. It is used conjointly with ICP in [Whelan et al., 2013a].

Chapter 7

Estimation of biases on angular velocity and acceleration

Estimation de biais pour le filtrage de la vitesse linéaire

Etant données des mesures de vitesses linéaire et angulaire par la méthode du flot géométrique telle que décrite en Chapitre 6, l'estimation de biais sur des données issues de gyromètres et d'accéléromètres est possible : on utilise pour cela des gyromètres et des accéléromètres biaisés et des magnétomètres. L'observabilité est garantie pour des mouvements non-planaires, et l'observateur est associé à une fonction de Lyapunov. On en déduit un système d'équations différentielles non-linéaires, dont la convergence asymptotique est étudiée dans le cadre d'hypothèses de bornitude sur les vitesses et de périodicité sur la vitesse angulaire. En outre, la périodicité des trajectoires garantit une convergence exponentielle locale, qui rend l'observateur robuste au bruit ou aux perturbations. L'implémentation sur des données réelles montre une estimation correcte des biais angulaires ; quant aux biais sur les accéléromètres, il n'est pas possible de conclure à la précision des résultats faute de référence, mais l'estimation réalisée permet une amélioration notable de la reconstruction de trajectoire. Ces résultats ont été présentés dans [Zarrouati-Vissière et al., 2013a].

Given the linear and angular velocity measurements provided by the geometrical flow method described in Chapter 6, estimation of biases on accelerometers and gyrometers is made possible, relying on biased accelerometers and gyrometers, and magnetometers measurements. Observability is guaranteed for non planar motion, and the observer design is based on a Lyapunov functional. It yields a system of non linear differential equations, whose asymptotic convergence towards equilibrium is studied in the framework of bounded velocities and periodic angular velocity. In addition, the periodicity of the trajectory guarantees local exponential convergence, which makes this observer robust to noise or perturbations. Implementation on real data shows that biases on angular velocity are correctly estimated; as for accelerometers biases, no reference enables to conclude on the accuracy of our results, but it is possible

to infer that trajectory estimation is indeed improved. These results were presented in [Zarrouati-Vissière et al., 2013a].

7.1 Problem formulation

The equations of motion of a 3D rigid body described in Appendix C, and more specifically equation C.5, can be rearranged in the following way:

$$\partial_t \boldsymbol{v} = \boldsymbol{v} \times \boldsymbol{\omega} + \boldsymbol{g} + \boldsymbol{a}$$

$$\partial_t \boldsymbol{g} = \boldsymbol{g} \times \boldsymbol{\omega}$$

where $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are the linear and angular velocity, $\boldsymbol{g}$ is the gravitational acceleration and $\boldsymbol{a}$ is the specific acceleration of the body, all of them expressed in the body frame. Embedded sensors make some measurements available: as real equipment, data contain errors and in particular constant errors, called biases. These biases should be estimated on line: this is the goal of this study, as well as trajectory reconstruction. To be specific, the measures are:

$$\boldsymbol{a_m} = \boldsymbol{a} + \boldsymbol{a_b}, \quad \boldsymbol{\omega_m} = \boldsymbol{\omega} + \boldsymbol{\omega_b}$$

where $\boldsymbol{a_b}$ and $\boldsymbol{\omega_b}$ are constant biases. Thus, the state equations of the system are

$$\begin{cases} \partial_t \boldsymbol{v} = \boldsymbol{v} \times (\boldsymbol{\omega_m} - \boldsymbol{\omega_b}) + \boldsymbol{g} + \boldsymbol{a_m} - \boldsymbol{a_b}, \\ \partial_t \boldsymbol{g} = \boldsymbol{g} \times (\boldsymbol{\omega_m} - \boldsymbol{\omega_b}), \\ \partial_t \boldsymbol{\omega_b} = 0, \\ \partial_t \boldsymbol{a_b} = 0. \end{cases} \tag{7.1}$$

In addition, measures of the linear and angular velocities are extracted from depth images processing (see Chapter 6): $\boldsymbol{v_m}$ and $\boldsymbol{\omega_m}'$. Comparison of $\boldsymbol{\omega_m}'$ and $\boldsymbol{\omega_m}$ gives a so-called measure of the angular bias: $\boldsymbol{\omega_{bm}}$.

Finally, $\boldsymbol{g}$, as the measure in a moving frame of the constant vector $\boldsymbol{A}$, has a constant norm. A last class of embedded sensors, the magnetometers, gives a measure $\boldsymbol{b}$ in the body frame of the magnetic field $\boldsymbol{B}$. $\boldsymbol{A}$ and $\boldsymbol{B}$ are known vectors. This yields the output equation

$$\boldsymbol{b} \cdot \boldsymbol{g} = \boldsymbol{A} \cdot \boldsymbol{B}.$$

The inputs of this system are $\boldsymbol{\omega_m}$ and $\boldsymbol{a_m}$, the outputs are $\boldsymbol{v_m}$, $\boldsymbol{\omega_{bm}}$ and $\boldsymbol{b}$.

7.2 Observability

To study the observability of the system, one looks for the assumptions on the trajectory under which two states $(\boldsymbol{v}^i, \boldsymbol{g}^i, \boldsymbol{\omega_b}^i, \boldsymbol{a_b}^i)$ $(i = 1, 2)$,verifying the same state equations with same inputs and outputs are necessarily equal. Since $\boldsymbol{v_m}$ is measured, $\boldsymbol{v}^1 = \boldsymbol{v}^2$. Similarly, $\boldsymbol{\omega_b}^1 = \boldsymbol{\omega_b}^2$.

Defining $\delta \boldsymbol{g} = \boldsymbol{g}^1 - \boldsymbol{g}^2$ and $\delta \boldsymbol{a_b} = \boldsymbol{a_b}^1 - \boldsymbol{a_b}^2$, the subtraction of the two systems of state equations yields

$$\begin{cases} 0 = \delta \boldsymbol{g} - \delta \boldsymbol{a_b}, \\ \partial_t \delta \boldsymbol{g} = \delta \boldsymbol{g} \times \boldsymbol{\omega}, \\ 0 = 0, \\ \partial_t \delta \boldsymbol{a_b} = 0, \end{cases}$$

and the output equation becomes

$$\delta \boldsymbol{g} \cdot \boldsymbol{b} = 0.$$

This is equivalent to

$$\begin{cases} \delta \boldsymbol{a_b} = \delta \boldsymbol{g}, \\ \partial_t \delta \boldsymbol{a_b} = 0, \\ \delta \boldsymbol{a_b} \times \boldsymbol{\omega} = 0, \\ \delta \boldsymbol{a_b} \cdot \boldsymbol{b} = 0. \end{cases} \tag{7.2}$$

which has to be verified for any time t. Thus, a necessary condition on the trajectory for this system to have a non trivial solution $\delta \boldsymbol{a_b}$ is that $\boldsymbol{\omega}(t)$ remains collinear to a fixed direction ($\delta \boldsymbol{a_b}$). In addition, this direction has to be orthogonal to $\boldsymbol{b}$.

Proposition 7. *The system is observable if and only if the angular velocity does not remain collinear to a fixed direction orthogonal to the magnetic field expressed in the body frame.*

Remark 3. *The following condition is sufficient for observability: $\exists t_1$ such that at least one of the following conditions is true*

$$\begin{cases} \boldsymbol{\omega}(t_1) \times \dot{\boldsymbol{\omega}}(t_1) \neq 0, \\ \boldsymbol{b}(t_1) \cdot \boldsymbol{\omega}(t_1) \neq 0. \end{cases} \tag{7.3}$$

7.3 Observer design

The observer is designed as follows:

$$\begin{cases} \partial_t \widehat{\boldsymbol{v}} = \widehat{\boldsymbol{v}} \times (\boldsymbol{\omega_m} - \widehat{\boldsymbol{\omega_b}}) + \widehat{\boldsymbol{g}} + \boldsymbol{a_m} - \widehat{\boldsymbol{a_b}} - k_v(\widehat{\boldsymbol{v}} - \boldsymbol{v_m}), \\ \partial_t \widehat{\boldsymbol{g}} = \widehat{\boldsymbol{g}} \times (\boldsymbol{\omega_m} - \boldsymbol{\omega_{bm}}) - k_{g1}(\widehat{\boldsymbol{v}} - \boldsymbol{v_m}) - k_{g2}(\widehat{\boldsymbol{g}} \cdot \boldsymbol{b} - \boldsymbol{A} \cdot \boldsymbol{B})\boldsymbol{b}, \\ \partial_t \widehat{\boldsymbol{\omega_b}} = -k_{w1}(\widehat{\boldsymbol{\omega_b}} - \boldsymbol{\omega_{bm}}) + k_{w2}(\widehat{\boldsymbol{v}} - \boldsymbol{v_m}) \times \widehat{\boldsymbol{v}}, \\ \partial_t \widehat{\boldsymbol{a_b}} = k_a(\widehat{\boldsymbol{v}} - \boldsymbol{v_m}). \end{cases} \tag{7.4}$$

From systems (7.1) and (7.4), and assuming that the measures $\boldsymbol{v_m}$ and $\boldsymbol{\omega_{bm}}$ are close enough to the real states $\boldsymbol{v}$ and $\boldsymbol{\omega_b}$ so that in particular, their dynamics are the same as the real dynamics, the error dynamics is

$$\begin{cases} \partial_t \widetilde{\boldsymbol{v}} = \widetilde{\boldsymbol{v}} \times (\boldsymbol{\omega_m} - \boldsymbol{\omega_b}) - \widehat{\boldsymbol{v}} \times \widetilde{\boldsymbol{\omega_b}} + \widetilde{\boldsymbol{g}} - \widetilde{\boldsymbol{a_b}} - k_v \widetilde{\boldsymbol{v}}, \\ \partial_t \widetilde{\boldsymbol{g}} = \widetilde{\boldsymbol{g}} \times (\boldsymbol{\omega_m} - \boldsymbol{\omega_b}) - k_{g1} \widetilde{\boldsymbol{v}} - k_{g2}(\widetilde{\boldsymbol{g}} \cdot \boldsymbol{b})\boldsymbol{b}, \\ \partial_t \widetilde{\boldsymbol{\omega_b}} = -k_{w1} \widetilde{\boldsymbol{\omega_b}} + k_{w2} \widetilde{\boldsymbol{v}} \times \widehat{\boldsymbol{v}}, \\ \partial_t \widetilde{\boldsymbol{a_b}} = k_a \widetilde{\boldsymbol{v}}. \end{cases} \tag{7.5}$$

where $\widetilde{\boldsymbol{v}} = \widehat{\boldsymbol{v}} - \boldsymbol{v_m}$, $\widetilde{\boldsymbol{g}} = \widehat{\boldsymbol{g}} - \boldsymbol{g}$, $\widetilde{\boldsymbol{\omega_b}} = \widehat{\boldsymbol{\omega_b}} - \boldsymbol{\omega_{bm}}$ and $\widetilde{\boldsymbol{a_b}} = \widehat{\boldsymbol{a_b}} - \boldsymbol{a_b}$. Linear velocity measurements ($\boldsymbol{v_m}$) are taken into account to correct the estimation of all states: although it seems quite natural for the linear velocity and the bias on acceleration estimates, that might be surprising for the specific acceleration and especially for the rotational bias estimates. Indeed, measurements of this bias being available, its estimation is straightforward. Actually, the correction terms are chosen so that the following Lyapunov functional is positive and decreases along the trajectories:

$$V = \lambda_v \|\widetilde{\boldsymbol{v}}\|^2 + \lambda_g \|\widetilde{\boldsymbol{g}}\|^2 + \lambda_w \|\widetilde{\boldsymbol{\omega_b}}\|^2 + \lambda_a \|\widetilde{\boldsymbol{a_b}}\|^2.$$

Indeed,

$$\begin{aligned}\dot{V} = &- \lambda_v k_v \|\widetilde{\boldsymbol{v}}\|^2 - \lambda_w k_{w1} \|\widetilde{\boldsymbol{\omega_b}}\|^2 - \lambda_g k_{g2} (\widetilde{\boldsymbol{g}} \cdot \boldsymbol{b})^2 \\ &+ (\lambda_v - \lambda_g k_{g1}) \widetilde{\boldsymbol{v}} \cdot \widetilde{\boldsymbol{g}} + (\lambda_a k_a - \lambda_v) \widetilde{\boldsymbol{v}} \cdot \widetilde{\boldsymbol{a_b}} + (\lambda_w k_{w2} - \lambda_v) \widetilde{\boldsymbol{v}} \cdot \widetilde{\boldsymbol{\omega_b}}\end{aligned}$$

Coupled terms are brought into play, that are canceled by the choice of specific correction gains. Choosing k_v, k_{w1}, $k_{g2} > 0$, $k_{g1} := \frac{\lambda_v}{\lambda_g}$, $k_a := \frac{\lambda_v}{\lambda_a}$ and $k_{w2} := \frac{\lambda_v}{\lambda_w}$ yields

$$\dot{V} = -\lambda_v k_v \|\widetilde{\boldsymbol{v}}\|^2 - \lambda_w k_{w1} \|\widetilde{\boldsymbol{\omega_b}}\|^2 - \lambda_g k_{g2} (\widetilde{\boldsymbol{g}} \cdot \boldsymbol{b})^2 \leq 0.$$

7.4 Asymptotic convergence

Proposition 8. *Assuming that linear and angular velocities are bounded and angular velocity is T-periodic, the error dynamics of the observer (defined in (7.5)) converge towards zero for any initial condition.*

Proof. **Step 1: Let us prove that $\widetilde{\boldsymbol{v}}$, $\widetilde{\boldsymbol{\omega_b}}$ and $\widetilde{\boldsymbol{g}} \cdot \boldsymbol{b}$ tend to 0 as t tends to ∞.**

The Lyapunov functional V decreases along the trajectories of the system and is positive, thus it tends to a limit $V_\infty \geq 0$. Using the previous expression of the time derivative of V,

$$V(t) = V(0) - \int_0^t \lambda_v k_v \|\widetilde{\boldsymbol{v}}(s)\|^2 + \lambda_w k_{w1} \|\widetilde{\boldsymbol{\omega_b}}(s)\|^2 + \lambda_g k_{g2} (\widetilde{\boldsymbol{g}}(s) \cdot \boldsymbol{b}(s))^2 ds$$

and we deduce that $t \mapsto \widetilde{\boldsymbol{v}}(t)$, $t \mapsto \widetilde{\boldsymbol{\omega_b}}(t)$ and $t \mapsto \widetilde{\boldsymbol{g}}(t) \cdot \boldsymbol{b}(t) \in L^2(0, \infty)$.

Then, $V(t) \in L^\infty(0, \infty)$ implies that $\widetilde{\boldsymbol{v}}(t)$, $\widetilde{\boldsymbol{\omega_b}}(t)$, $\widetilde{\boldsymbol{g}}(t)$ and $\widetilde{\boldsymbol{a_b}}(t) \in L^\infty(0, \infty)$ themselves. Assuming that $t \mapsto \boldsymbol{v}(t)$ and $t \mapsto \boldsymbol{\omega}(t) \in L^\infty(0, \infty)$, the time derivatives of $\widetilde{\boldsymbol{v}}$, $\widetilde{\boldsymbol{\omega_b}}$ and $\widetilde{\boldsymbol{g}} \cdot \boldsymbol{b}$ are bounded. Thus $\widetilde{\boldsymbol{v}}$, $\widetilde{\boldsymbol{\omega_b}}$ and $\widetilde{\boldsymbol{g}} \cdot \boldsymbol{b}$ are uniformly continuous. Using Barbalat's lemma, we conclude that

$$(\widetilde{\boldsymbol{v}}, \widetilde{\boldsymbol{\omega_b}}, \widetilde{\boldsymbol{g}} \cdot \boldsymbol{b})(t) \underset{t \to +\infty}{\longrightarrow} (0, 0, 0).$$

Step 2: The periodic system Since the angular velocity is T-periodic, let us define $(t_n)_{n \in \mathbb{N}}$ an increasing sequence such that $t_n \underset{n \to +\infty}{\longrightarrow} \infty$: up to an extraction, the bounded sequence $(t_n[T])_{n \in \mathbb{N}}$ has an adherence value, called α. Since the Lyapunov function V is bounded, the sequences $(\boldsymbol{b}, \widetilde{\boldsymbol{v}}, \widetilde{\boldsymbol{g}}, \widetilde{\boldsymbol{\omega_b}}, \widetilde{\boldsymbol{a_b}})(t_n)$ are bounded, thus up to an extraction, $\exists (\boldsymbol{b}^0_\infty, \widetilde{\boldsymbol{v}}^0_\infty, \widetilde{\boldsymbol{g}}^0_\infty, \widetilde{\boldsymbol{\omega_b}}^0_\infty, \widetilde{\boldsymbol{a_b}}^0_\infty) \in (\mathbb{R}^3)^5$ such that

$$(\boldsymbol{b}, \widetilde{\boldsymbol{v}}, \widetilde{\boldsymbol{g}}, \widetilde{\boldsymbol{\omega_b}}, \widetilde{\boldsymbol{a_b}})(t_n) \underset{n \to +\infty}{\longrightarrow} (\boldsymbol{b}^0_\infty, \widetilde{\boldsymbol{v}}^0_\infty, \widetilde{\boldsymbol{g}}^0_\infty, \widetilde{\boldsymbol{\omega_b}}^0_\infty, \widetilde{\boldsymbol{a_b}}^0_\infty).$$

Let us define the maps $(\alpha, +\infty) \ni t \mapsto (\boldsymbol{b}_\infty, \widetilde{\boldsymbol{v}}_\infty, \widetilde{\boldsymbol{g}}_\infty, \widetilde{\boldsymbol{\omega_b}}_\infty, \widetilde{\boldsymbol{a_b}}_\infty)(t) \in (\mathbb{R}^3)^5$ solutions of the following Cauchy problem:

$$\Sigma_\infty \begin{cases} \partial_t \boldsymbol{b}_\infty = \boldsymbol{b}_\infty \times \boldsymbol{\omega}, \\ \partial_t \widetilde{\boldsymbol{v}}_\infty = \widetilde{\boldsymbol{v}}_\infty \times \boldsymbol{\omega} + \widetilde{\boldsymbol{g}}_\infty - \widetilde{\boldsymbol{a_b}}_\infty - k_v \widetilde{\boldsymbol{v}}_\infty, \\ \partial_t \widetilde{\boldsymbol{g}}_\infty = \widetilde{\boldsymbol{g}}_\infty \times \boldsymbol{\omega} - k_{g1} \widetilde{\boldsymbol{v}}_\infty, \\ \partial_t \widetilde{\boldsymbol{\omega_b}}_\infty = -k_{w1} \widetilde{\boldsymbol{\omega_b}}_\infty, \\ \partial_t \widetilde{\boldsymbol{a_b}}_\infty = k_a \widetilde{\boldsymbol{v}}_\infty, \end{cases} \tag{7.6}$$

with initial conditions

$$\begin{cases} \boldsymbol{b}_\infty(\alpha) = \boldsymbol{b}_\infty^0, \\ \widetilde{\boldsymbol{v}}_\infty(\alpha) = \widetilde{\boldsymbol{v}}_\infty^0, \\ \widetilde{\boldsymbol{g}}_\infty(\alpha) = \widetilde{\boldsymbol{g}}_\infty^0, \\ \widetilde{\boldsymbol{\omega_b}}_\infty(\alpha) = \widetilde{\boldsymbol{\omega_b}}_\infty^0, \\ \widetilde{\boldsymbol{a_b}}_\infty(\alpha) = \widetilde{\boldsymbol{a_b}}_\infty^0. \end{cases} \tag{7.7}$$

Thanks to Step 1, we know that

$$\widetilde{\boldsymbol{v}}_\infty^0 = 0, \quad \widetilde{\boldsymbol{\omega_b}}_\infty^0 = 0, \quad \widetilde{\boldsymbol{g}}_\infty^0 \cdot \boldsymbol{b}_\infty^0 = 0.$$

Let us define $\boldsymbol{X} := (\boldsymbol{b}, \widetilde{\boldsymbol{v}}, \widetilde{\boldsymbol{g}}, \widetilde{\boldsymbol{\omega_b}}, \widetilde{\boldsymbol{a_b}})$ and $\boldsymbol{X}_\infty = (\boldsymbol{b}_\infty, \widetilde{\boldsymbol{v}}_\infty, \widetilde{\boldsymbol{g}}_\infty, \widetilde{\boldsymbol{\omega_b}}_\infty, \widetilde{\boldsymbol{a_b}}_\infty)$.

Step 3: Let us prove that $\forall \tau > 0, \boldsymbol{X}(t_n + \tau) \underset{n \to +\infty}{\longrightarrow} \boldsymbol{X}_\infty(\alpha + \tau)$. Let $\tau > 0$. On one hand, $\boldsymbol{X}_\infty$ is the solution of the periodic linear system of dynamical equations Σ_∞: it writes $\partial_t \boldsymbol{X}_\infty = A_\infty(t) \boldsymbol{X}_\infty$. On the other hand, $\boldsymbol{X}$ is the solution of almost the same system, differing by an additive term that is proven to tend to 0 as t tends to ∞: $\partial_t \boldsymbol{X} = A_\infty(t) \boldsymbol{X} + \boldsymbol{c}(t)$ with

$$\boldsymbol{c}(t) = \begin{pmatrix} 0 \\ -\widehat{\boldsymbol{v}} \times \widetilde{\boldsymbol{\omega_b}} \\ -k_{g2} \boldsymbol{b} (\widetilde{\boldsymbol{g}} \cdot \boldsymbol{b}) \\ k_{w2} \widetilde{\boldsymbol{v}} \times \widehat{\boldsymbol{v}} \\ 0 \end{pmatrix} \tag{7.8}$$

Let us introduce $\mathcal{R}_\infty(t, s)$ the resolvant asssociated to $A_\infty(t)$:

$$\boldsymbol{X}_\infty(t) = \mathcal{R}_\infty(t, s) \cdot \boldsymbol{X}_\infty(s) \quad \forall s, t > 0.$$

In particular,

$$\boldsymbol{X}_\infty(\alpha + \tau) = \mathcal{R}_\infty(\alpha + \tau, \alpha) \cdot \boldsymbol{X}_\infty^0 \quad \forall \tau > 0. \tag{7.9}$$

Since the system Σ_∞ is T-periodic, $\mathcal{R}_\infty(t + T, s + T) = \mathcal{R}_\infty(t, s)$. Let us define $(k_n)_{n \in \mathbb{N}}$ such that $t_n = k_n T + \alpha_n$. From Step 2, $\alpha_n \underset{n \to +\infty}{\longrightarrow} \alpha$. Then,

$$\forall n \in \mathbb{N}, \mathcal{R}_\infty(t_n + \tau, t_n) = \mathcal{R}_\infty(\alpha_n + \tau, \alpha_n)$$

and

$$\mathcal{R}_\infty(\alpha_n + \tau, \alpha_n) \underset{n \to +\infty}{\longrightarrow} \mathcal{R}_\infty(\alpha + \tau, \alpha) \tag{7.10}$$

by continuity of the resolvant. Now, from the variation of constants method,

$$\boldsymbol{X}(t_n+\tau) = \mathcal{R}_\infty(t_n+\tau,t_n)\cdot\boldsymbol{X}(t_n) + \int_{t_n}^{t_n+\tau} \mathcal{R}_\infty(t_n+\tau,s)\cdot\boldsymbol{c}(s)ds. \tag{7.11}$$

Substracting (7.9) and(7.11) yields

$$\boldsymbol{X}(t_n+\tau)-\boldsymbol{X}_\infty(\alpha+\tau) = \mathcal{R}_\infty(t_n+\tau,t_n)\cdot\boldsymbol{X}(t_n)-\mathcal{R}_\infty(\alpha+\tau,\alpha)\cdot\boldsymbol{X}^0_\infty + \int_{t_n}^{t_n+\tau} \mathcal{R}_\infty(t_n+\tau,s)\cdot\boldsymbol{c}(s)ds.$$

Thus,

$$\|\boldsymbol{X}(t_n+\tau)-\boldsymbol{X}_\infty(\alpha+\tau)\| \le A_n(\tau)+B_n(\tau)$$

with

$$A_n(\tau) = \|\mathcal{R}_\infty(t_n+\tau,t_n)\cdot\boldsymbol{X}(t_n)-\mathcal{R}_\infty(\alpha+\tau,\alpha)\cdot\boldsymbol{X}^0_\infty\|$$

and

$$B_n(\tau) = \|\int_{t_n}^{t_n+\tau} \mathcal{R}_\infty(t_n+\tau,s)\cdot\boldsymbol{c}(s)ds.\|$$

First, let us prove that $A_n(\tau) \underset{n\to+\infty}{\longrightarrow} 0 \quad \forall\tau>0$:

$$\begin{aligned}
A_n &= \|(\mathcal{R}_\infty(t_n+\tau,t_n)-\mathcal{R}_\infty(\alpha+\tau,\alpha))\cdot\boldsymbol{X}(t_n)+\mathcal{R}_\infty(\alpha+\tau,\alpha)\cdot(\boldsymbol{X}(t_n)-\boldsymbol{X}^0_\infty)\| \\
&= \|(\mathcal{R}_\infty(\alpha_n+\tau,\alpha_n)-\mathcal{R}_\infty(\alpha+\tau,\alpha))\cdot\boldsymbol{X}(t_n)+\mathcal{R}_\infty(\alpha+\tau,\alpha)\cdot(\boldsymbol{X}(t_n)-\boldsymbol{X}^0_\infty)\| \\
&\le \|(\mathcal{R}_\infty(\alpha_n+\tau,\alpha_n)-\mathcal{R}_\infty(\alpha+\tau,\alpha))\|\sup_{n\in\mathbb{N}}\|\boldsymbol{X}(t_n)\| + \|\mathcal{R}_\infty(\alpha+\tau,\alpha)\|\sup_{n\in\mathbb{N}}\|\boldsymbol{X}(t_n)-\boldsymbol{X}^0_\infty\|
\end{aligned}$$

From (7.10),

$$\|(\mathcal{R}_\infty(\alpha_n+\tau,\alpha)-\mathcal{R}_\infty(\alpha+\tau,\alpha))\| \underset{n\to+\infty}{\longrightarrow} 0.$$

From Step 1, $\sup_{n\in\mathbb{N}}\|\boldsymbol{X}(t_n)\| < +\infty$ By definition of $\boldsymbol{X}^0_\infty$, $\|\boldsymbol{X}(t_n)-\boldsymbol{X}^0_\infty\| \underset{n\to+\infty}{\longrightarrow} 0$. This proves that $A_n \underset{n\to+\infty}{\longrightarrow} 0$.

Let us prove that $B_n \underset{n\to+\infty}{\longrightarrow} 0$. Let us define the maps $f_n : s\in(0,\tau) \mapsto c(t_n+s)$. By definition of $c(t)$, $\forall s\in(0,\tau), f_n(s) \underset{n\to+\infty}{\longrightarrow} 0$. As $f_n(s)$ is bounded on $(0,\tau)$, according to the dominated convergence theorem, $\|f_n\|_{L_1(0,\tau)} \underset{n\to+\infty}{\longrightarrow} 0$. Thus $\|c\|_{L_1(t_n,t_n+\tau)} \underset{n\to+\infty}{\longrightarrow} 0$. Since $B_n \le \tau\|\mathcal{R}_\infty\|\|c\|_{L_1(t_n,t_n+\tau)}$, we proved that $B_n \underset{n\to+\infty}{\longrightarrow} 0$. Thus, $\boldsymbol{X}(t_n+\tau) \underset{n\to+\infty}{\longrightarrow} \boldsymbol{X}_\infty(\alpha+\tau)$.

Step 4: Let us prove that $\forall\tau>0, \boldsymbol{X}_\infty(\alpha+\tau)=0$ From Step 1 and Step 3, we know that

$$\forall\tau>0, \begin{cases} \widetilde{\boldsymbol{v}}_\infty(\alpha+\tau)=0 \\ \widetilde{\boldsymbol{\omega_b}}_\infty(\alpha+\tau)=0 \\ \widetilde{\boldsymbol{g}}_\infty(\alpha+\tau)\cdot\boldsymbol{b}_\infty(\alpha+\tau)=0 \end{cases} \tag{7.12}$$

Since $\boldsymbol{X}_\infty$ is the solution of Σ_∞ and $\boldsymbol{\omega}\in\mathcal{C}^1(\alpha,+\infty)$, we deduce that $\boldsymbol{X}_\infty\in\mathcal{C}^1(\alpha,+\infty)$ and thus, from the two first equalities of (7.12) that

$$\partial_t\widetilde{\boldsymbol{v}}_\infty(\alpha+\tau)=0, \quad \partial_t\widetilde{\boldsymbol{\omega_b}}_\infty(\alpha+\tau)=0 \quad \forall\tau>0.$$

Injecting all of this in Σ_∞ yields

$$\forall \tau > 0, \begin{cases} \partial_t \boldsymbol{b}_\infty(\alpha+\tau) = \boldsymbol{b}_\infty(\alpha+\tau) \times \boldsymbol{\omega}(\alpha+\tau), \\ 0 = \widetilde{\boldsymbol{g}}_\infty(\alpha+\tau) - \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau), \\ \partial_t \widetilde{\boldsymbol{g}}_\infty(\alpha+\tau) = \widetilde{\boldsymbol{g}}_\infty(\alpha+\tau) \times \boldsymbol{\omega}(\alpha+\tau), \\ \partial_t \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau) = 0, \end{cases} \tag{7.13}$$

which, in addition to the third equality of (7.12), is equivalent to

$$\forall \tau > 0, \begin{cases} \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau) = \widetilde{\boldsymbol{g}}_\infty(\alpha+\tau), \\ \partial_t \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau) = 0, \\ \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau) \times \boldsymbol{\omega}(\alpha+\tau) = 0, \\ \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau) \cdot \boldsymbol{b}_\infty(\alpha+\tau) = 0. \end{cases}$$

Finally, using the Proposition 7 and the following Remark, we deduce that if $\exists t_1$ such that $\boldsymbol{\omega}(t_1) \times \dot{\boldsymbol{\omega}}(t_1) \neq 0$ or $\boldsymbol{b}(t_1) \cdot \boldsymbol{\omega}(t_1) \neq 0$, then $\widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau) = \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty(\alpha+\tau) = 0 \forall \tau > 0$, which concludes the proof of Step 4. □

7.5 Exponential convergence

Proposition 9. *If in addition to the previous assumptions, we assume that the trajectory is T-periodic, the observer exponentially locally converges towards the real trajectory.*

Proof. Since the trajectory is T-periodic, the time functions $\boldsymbol{b}(t)$, $\boldsymbol{v}(t)$ and $\boldsymbol{\omega}(t)$ are T-periodic. To prove the exponential convergence, let us underline how the steps of the proof of asymptotic convergence are modified. The step 1 remains unchanged. In step 2, the Σ_∞ system is defined as in the following:

$$\Sigma_\infty \begin{cases} \partial_t \boldsymbol{b}_\infty = \boldsymbol{b}_\infty \times \boldsymbol{\omega}, \\ \partial_t \widetilde{\boldsymbol{v}}_\infty = \widetilde{\boldsymbol{v}}_\infty \times \boldsymbol{\omega} - \boldsymbol{v} \times \widetilde{\boldsymbol{\omega}_{\boldsymbol{b}}}_\infty + \widetilde{\boldsymbol{g}}_\infty - \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty - k_v \widetilde{\boldsymbol{v}}_\infty, \\ \partial_t \widetilde{\boldsymbol{g}}_\infty = \widetilde{\boldsymbol{g}}_\infty \times \boldsymbol{\omega} - k_{g1} \widetilde{\boldsymbol{v}}_\infty - k_{g2} (\widetilde{\boldsymbol{g}}_\infty \cdot \boldsymbol{b}_\infty) \boldsymbol{b}_\infty, \\ \partial_t \widetilde{\boldsymbol{\omega}_{\boldsymbol{b}}}_\infty = -k_{w1} \widetilde{\boldsymbol{\omega}_{\boldsymbol{b}}}_\infty + k_{w2} \widetilde{\boldsymbol{v}}_\infty \times \boldsymbol{v}, \\ \partial_t \widetilde{\boldsymbol{a}_{\boldsymbol{b}}}_\infty = k_a \widetilde{\boldsymbol{v}}_\infty, \end{cases} \tag{7.14}$$

with same initial conditions. In the step 3, the term $\boldsymbol{c}(t)$ is now defined as

$$\boldsymbol{c}(t) = \begin{pmatrix} 0 \\ -\widetilde{\boldsymbol{v}} \times \widetilde{\boldsymbol{\omega}_{\boldsymbol{b}}} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{7.15}$$

Same conditions remain true on the time function $\boldsymbol{c}$ and on the resolvant of Σ_∞, thus the conclusion holds:

$$\forall \tau > 0, \boldsymbol{X}(t_n + \tau) \underset{n \to +\infty}{\longrightarrow} \boldsymbol{X}_\infty(\alpha+\tau).$$

In the Step 4, we deduce as previously from the observability assumption on the system that $\forall \tau > 0, \boldsymbol{X}_\infty(\alpha+\tau) = 0$. Then, since Σ_∞ is periodic, a usual result on periodic systems states that $\exists \bar{U}, \lambda > 0$ such that $\|\mathcal{R}_\infty(t,s)\| \leq \bar{U}\exp^{-\lambda(t-s)}$. Then

$$\boldsymbol{X}(t) = \mathcal{R}_\infty(t)\cdot\boldsymbol{X}(0) + \int_0^t \mathcal{R}_\infty(t,s)\cdot\boldsymbol{c}(s)ds$$

where $\boldsymbol{c}$ is quadratic in $\boldsymbol{X}$: $\exists \mu > 0$ such that $\|\boldsymbol{c}(s)\| \leq \mu\|\boldsymbol{X}(s)\|^2$. Thus

$$\|\boldsymbol{X}(t)\| \leq \bar{U}\exp^{-\lambda t}\|\boldsymbol{X}(0)\| + \int_0^t \bar{U}\exp^{-\lambda(t-s)}\,\mu\|\boldsymbol{X}(s)\|^2 ds$$

or

$$\|\boldsymbol{X}(t)\|\exp^{\lambda t} \leq \bar{U}\|\boldsymbol{X}(0)\| + \int_0^t \bar{U}\exp^{\lambda s}\,\mu\|\boldsymbol{X}(s)\|^2 ds$$

Let us define $\delta_0 := \|\boldsymbol{X}(0)\|$, $\delta_1 > \max(\delta_0, \bar{U}\delta_0)$, $t_1 := \sup\{t > 0; \|\boldsymbol{X}(s)\| \leq \delta_1, \forall s \in (0,t)\}$. Working by contradiction, we assume that $t_1 < +\infty$. We apply Gronwall's lemma to the inequality

$$\|\boldsymbol{X}(t)\|\exp^{\lambda t} \leq \bar{U}\delta_0 + \int_0^t \bar{U}\mu\delta_1\exp^{\lambda s}\|\boldsymbol{X}(s)\|ds, \quad t \leq t_1.$$

It yields

$$\|\boldsymbol{X}(t)\|\exp^{\lambda t} \leq \bar{U}\delta_0\exp^{\bar{U}\mu\delta_1 t}$$

$$\|\boldsymbol{X}(t)\| \leq \delta_1\exp^{(\bar{U}\mu\delta_1-\lambda)t}$$

Assuming $\delta_1 \leq \lambda/(2\mu\bar{U})$ yields $\|\boldsymbol{X}(t)\| \leq \delta_1\exp^{-\lambda/2t}$: in particular, this implies that $\|\boldsymbol{X}(t_1)\| < \delta_1$, which contradicts the definition of t_1. To conclude, under the assumption that $\|\boldsymbol{X}(0)\| \leq \min(\lambda/(2\mu\bar{U}), \lambda/(2\mu\bar{U}^2))$, $\|\boldsymbol{X}(t)\|$ has an exponential decay:

$$\|\boldsymbol{X}(t)\| \leq \lambda/(2\mu\bar{U})\exp^{-\lambda/2t}.$$

The asymptotic convergence of the periodic system yields the exponential convergence of the complete system for initial conditions close enough to the real trajectory. □

7.6 Robustness to noise

As for any real system equipped with sensors, measurements are not ideal: the inputs and outputs of the system are noisy representations of the real variables that they represent. Let us define $(\boldsymbol{\xi}_a, \boldsymbol{\xi}_w, \boldsymbol{\xi}_{wb}, \boldsymbol{\xi}_v, \boldsymbol{\xi}_b)$ such that

$$\begin{cases} \boldsymbol{a_m} &= \boldsymbol{a} + \boldsymbol{a_b} + \boldsymbol{\xi}_a, \\ \boldsymbol{\omega_m} &= \boldsymbol{\omega} + \boldsymbol{\omega_b} + \boldsymbol{\xi}_w, \\ \boldsymbol{\omega_{bm}} &= \boldsymbol{\omega_b} + \boldsymbol{\xi}_{wb}, \\ \boldsymbol{v_m} &= \boldsymbol{v} + \boldsymbol{\xi}_v, \\ \boldsymbol{b}_m &= \boldsymbol{b} + \boldsymbol{\xi}_b. \end{cases} \tag{7.16}$$

Using the observer defined by (7.4), the error dynamics is now given by

$$\left\{\begin{array}{ll} \partial_t\widetilde{\boldsymbol{v}} = & \widetilde{\boldsymbol{v}}\times\boldsymbol{\omega} - \widehat{\boldsymbol{v}}\times\widetilde{\boldsymbol{\omega_b}} + \widetilde{\boldsymbol{g}} - \widetilde{\boldsymbol{a_b}} - k_v\widetilde{\boldsymbol{v}} + \boldsymbol{v}\times\boldsymbol{\xi}_w + \boldsymbol{\xi}_a + k_v\boldsymbol{\xi}_v + \widetilde{\boldsymbol{v}}\times\boldsymbol{\xi}_w, \\ \partial_t\widetilde{\boldsymbol{g}} = & \widetilde{\boldsymbol{g}}\times\boldsymbol{\omega} - k_{g1}\widetilde{\boldsymbol{v}} - k_{g2}(\widetilde{\boldsymbol{g}}\cdot\boldsymbol{b})\boldsymbol{b} + \boldsymbol{g}\times(\boldsymbol{\xi}_w - \boldsymbol{\xi}_{wb}) + k_{g1}\boldsymbol{\xi}_v - k_{g2}(\widetilde{\boldsymbol{g}}\cdot\boldsymbol{b})\boldsymbol{\xi}_b - k_{g2}(\widehat{\boldsymbol{g}}\cdot\boldsymbol{\xi}_b)(\boldsymbol{b}+\boldsymbol{\xi}_b) \\ & +\widetilde{\boldsymbol{g}}\times(\boldsymbol{\xi}_w - \boldsymbol{\xi}_{wb}), \\ \partial_t\widetilde{\boldsymbol{\omega_b}} = & -k_{w1}\widetilde{\boldsymbol{\omega_b}} + k_{w2}\widetilde{\boldsymbol{v}}\times\boldsymbol{v} + k_{w1}\boldsymbol{\xi}_{wb} - k_{w2}\boldsymbol{\xi}_v\times\boldsymbol{v} - k_{w2}\boldsymbol{\xi}_v\times\widetilde{\boldsymbol{v}}, \\ \partial_t\widetilde{\boldsymbol{a_b}} = & k_a\widetilde{\boldsymbol{v}} - k_a\boldsymbol{\xi}_v. \end{array}\right. \tag{7.17}$$

In compact form, and using the notations of Section 7.5 studying exponential convergence, this writes:

$$\partial_t\boldsymbol{Y} = A_\infty(t)\boldsymbol{Y} + \boldsymbol{c}(t) + \boldsymbol{f_1}(t) + f_2(t)\boldsymbol{Y}$$

where

$$\boldsymbol{f_1}(t) := \begin{pmatrix} \boldsymbol{v}\times\boldsymbol{\xi}_w + \boldsymbol{\xi}_a + k_v\boldsymbol{\xi}_v \\ \boldsymbol{g}\times(\boldsymbol{\xi}_w - \boldsymbol{\xi}_{wb}) + k_{g1}\boldsymbol{\xi}_v - k_{g2}(\widetilde{\boldsymbol{g}}\cdot\boldsymbol{b})\boldsymbol{\xi}_b - k_{g2}(\widehat{\boldsymbol{g}}\cdot\boldsymbol{\xi}_b)(\boldsymbol{b}+\boldsymbol{\xi}_b) \\ k_{w1}\boldsymbol{\xi}_{wb} - k_{w2}\boldsymbol{\xi}_v\times\boldsymbol{v} \\ -k_a\boldsymbol{\xi}_v \end{pmatrix}, \tag{7.18}$$

and

$$f_2(t)\boldsymbol{Y} := \begin{pmatrix} \widetilde{\boldsymbol{v}}\times\boldsymbol{\xi}_w \\ \widetilde{\boldsymbol{g}}\times(\boldsymbol{\xi}_w - \boldsymbol{\xi}_{wb}) \\ -k_{w2}\boldsymbol{\xi}_v\times\widetilde{\boldsymbol{v}} \\ 0 \end{pmatrix}. \tag{7.19}$$

Recall also that

- the resolvent $\mathcal{R}_\infty$ associated to the linear T-periodic matrix A_∞ satisfies $\|\mathcal{R}_\infty(t,s)\| \leq \bar{U}\exp^{-\lambda(t-s)}$;
- $\boldsymbol{c}(t)$ is actually quadratic in $\boldsymbol{Y}(t)$: $\boldsymbol{c}(t) = Q(\boldsymbol{Y}(t))$.

Let us recall the result of Section 7.5 on the following Cauchy problem:

$$\left\{\begin{array}{l} \partial_t\boldsymbol{X} = A_\infty(t)\boldsymbol{X} + \boldsymbol{c}(t) \\ \boldsymbol{X}(0) = \boldsymbol{X}_0. \end{array}\right. \tag{7.20}$$

$$\exists\delta, K, \lambda > 0 \text{ such that } \|\boldsymbol{X}_0\| < \delta \text{ implies } \|\boldsymbol{X}(t)\| \leq K\|\boldsymbol{X}_0\|\exp^{-\lambda t}\ \forall t > 0. \tag{7.21}$$

The following proposition states that our observer is locally robust to deterministic noise on measurements:

Proposition 10. *In addition to the previous assumptions, let us assume that the perturbations are characterized by some constants F_1 and F_2 such that $|\boldsymbol{f_1}(t)| \leq F_1$ and $\|f_2(t)\boldsymbol{Y}\| \leq F_2\|\boldsymbol{Y}\|$. $\boldsymbol{X}(t)$ is the solution of (7.20) with $\|\boldsymbol{X}_0\| < \delta$ so that (7.21) is satisfied. Then the solution $\boldsymbol{Y}$ to the Cauchy problem*

$$\left\{\begin{array}{l} \partial_t\boldsymbol{Y} = A_\infty(t)\boldsymbol{Y} + Q(\boldsymbol{Y}(t)) + \boldsymbol{f_1}(t) + f_2(t)\boldsymbol{Y} \\ \boldsymbol{Y}(0) = \boldsymbol{X}_0 \end{array}\right. \tag{7.22}$$

satisfies

$$\|\boldsymbol{Y}(t) - \boldsymbol{X}(t)\| \leq \frac{K_L(F_1 + F_2\|\boldsymbol{X}_0\|)}{c_L} + o(F_1 + F_2)$$

for some positive constants K_L and c_L depending on the initial condition $\|\boldsymbol{X}_0\|$ and the level of perturbation F_2.

Proof. **Step 1: Linearization of the perturbed system** Let us define $\boldsymbol{Y}_1$ the solution of the following Cauchy problem:

$$\begin{cases} \partial_t \boldsymbol{Y}_1 = (A_\infty(t) + \partial_X Q(\boldsymbol{X}(t)) + f_2(t))\,\boldsymbol{Y}_1 + f_1(t) + f_2(t)\boldsymbol{X}(t) \\ \boldsymbol{Y}_1(0) = 0 \end{cases} \tag{7.23}$$

Since Q is quadratic in $\boldsymbol{X}$, $\partial_X Q(\boldsymbol{X}(t))$ is linear in $\boldsymbol{X}$: in particular, (7.21) implies that $\|\partial_X Q(\boldsymbol{X}(t))\| \leq Q_1$ for some positive constant $Q_1(\|\boldsymbol{X}_0\|)$. Similarly, (7.21) implies that $\|f_2(t)\boldsymbol{X}(t)\| \leq F_2\|\boldsymbol{X}_0\|$.

Step 2: Let us prove that the resolvent associated to the linearized system has an exponential decay. Let us consider the following Cauchy problem

$$\begin{cases} \partial_t \boldsymbol{Y}_2 = (A_\infty(t) + \partial_X Q(\boldsymbol{X}(t)) + f_2(t))\,\boldsymbol{Y}_2 \\ \boldsymbol{Y}_2(0) = \boldsymbol{Y}_0 \end{cases} \tag{7.24}$$

The explicit solution of this problem is given by the method of variation of constants, where $\mathcal{R}_\infty$ is the resolvent associated to A_∞:

$$\boldsymbol{Y}_2(t) = \mathcal{R}_\infty(t,0)\boldsymbol{Y}_0 + \int_0^t \mathcal{R}_\infty(t,s)(\partial_X Q(\boldsymbol{X}(s)) + f_2(s))\boldsymbol{Y}_2(s)ds.$$

This yields

$$\|\boldsymbol{Y}_2(t)\| \leq \|\mathcal{R}_\infty(t,0)\|\|\boldsymbol{Y}_0\| + \int_0^t \|\mathcal{R}_\infty(t,s)\|\big(Q_1(\|\boldsymbol{X}_0\| + F_2\big)\|\boldsymbol{Y}_2(s)\|ds,$$

$$\|\boldsymbol{Y}_2(t)\| \leq \bar{U}\|\boldsymbol{Y}_0\|\exp^{-\lambda t} + \int_0^t \bar{U}\exp^{-\lambda(t-s)}\big(Q_1(\|\boldsymbol{X}_0\|) + F_2\big)\|\boldsymbol{Y}_2(s)\|ds.$$

Then, by a similar reasoning than used in Section 7.5, $\exists \delta_y, K_L(\|\boldsymbol{X}_0\|, \sigma_2)$ and $c_L(\|\boldsymbol{X}_0\|, F_2)$ such that

$$\|\boldsymbol{Y}_0\| < \delta_y \text{ implies } \|\boldsymbol{Y}_2(t)\| \leq K_L\|\boldsymbol{Y}_0\|\exp^{-c_L t}\,\forall t > 0. \tag{7.25}$$

On the other hand, the solution of the Cauchy problem $\boldsymbol{Y}_2$ writes $\boldsymbol{Y}_2(t) = \mathcal{R}_L(t,0)\boldsymbol{Y}_0$, where $\mathcal{R}_L$ is the resolvent associated to $A_\infty(t) + \partial_X Q(\boldsymbol{X}(t)) + f_2(t)$. This proves that $\|\mathcal{R}_L(t,s)\| \leq K_L\exp^{-c_L(t-s)}$.

Step 3: Let us prove that $\|\boldsymbol{Y}_1(t)\| \leq \frac{K_L(F_1+F_2\|\boldsymbol{X}_0\|)}{c_L}$. Using the definition of $\mathcal{R}_L$ and the variation of constants to the linearized Cauchy problem (7.23) yields

$$\boldsymbol{Y}_1(t) = \int_0^t \mathcal{R}_L(t,s)(\boldsymbol{f_1}(s) + f_2(s)\boldsymbol{X}(s))ds,$$

thus

$$\|\boldsymbol{Y}_1(t)\| \leq \int_0^t K_L \exp^{-c_L(t-s)}(F_1 + F_2\|\boldsymbol{X}_0\|)ds,$$

and finally

$$\|\boldsymbol{Y}_1(t)\| \leq \frac{K_L(F_1 + F_2\|\boldsymbol{X}_0\|)}{c_L}.$$

Step 4: Conclusion We proved that $\|\boldsymbol{Y}_1(t)\| = O(F_1 + F_2)$. Let us introduce the function $\boldsymbol{Y}_3 = \boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{Y}_1$. We write $\boldsymbol{Y}_3 = (F_1 + F_2)\boldsymbol{y}$. We will prove that $\boldsymbol{y} = O(F_1 + F_2)$. $\boldsymbol{Y}_3$ is the solution of the following Cauchy problem:

$$\begin{cases} \partial_t \boldsymbol{Y}_3 = (A_\infty(t))\,\boldsymbol{Y}_3 + Q(\boldsymbol{Y}_3 + \boldsymbol{X} + \boldsymbol{Y}_1) - Q(\boldsymbol{X}) - \partial_X Q(\boldsymbol{X}(t)) \cdot \boldsymbol{Y}_1 + f_2(t)(\boldsymbol{X}(t) + \boldsymbol{Y}_3(t)) \\ \boldsymbol{Y}_3(0) = 0 \end{cases} \tag{7.26}$$

Since Q is quadratic, $Q(\boldsymbol{Y}_3 + \boldsymbol{X} + \boldsymbol{Y}_1)$ can be written in the form

$$Q(\boldsymbol{Y}_3) + Q(\boldsymbol{X} + \boldsymbol{Y}_1) + L\boldsymbol{Y}_3 \cdot (\boldsymbol{X} + \boldsymbol{Y}_1)$$

where L is a constant. Since $\boldsymbol{Y}_1 = O(F_1 + F_2)$, $Q(\boldsymbol{X} + \boldsymbol{Y}_1)$ can be written in the form

$$Q(\boldsymbol{X}) + \partial_X Q(\boldsymbol{X}(t)) \cdot \boldsymbol{Y}_1 + (F_1 + F_2)o(1).$$

Thus, (7.26) writes

$$\begin{cases} \partial_t \boldsymbol{Y}_3 = (A_\infty(t) + L(\boldsymbol{X} + \boldsymbol{Y}_1) + f_2(t))\,\boldsymbol{Y}_3 + Q(\boldsymbol{Y}_3) + (F_1 + F_2)o(1) + f_2(t)\boldsymbol{X}(t) \\ \boldsymbol{Y}_3(0) = 0 \end{cases} \tag{7.27}$$

From the linear part of this system, we conclude as in Step 2 that the resolvent has an exponential decay since the terms $L(\boldsymbol{X} + \boldsymbol{Y}_1)$ and f_2 are bounded and small. We write this resolvent $\mathcal{R}_3$:

$$\|\mathcal{R}_3(t,s)\| \leq K_3 \exp^{-c_3(t-s)}.$$

Then, by variation of constants,

$$\boldsymbol{Y}_3(t) = \int_0^t \mathcal{R}_3(t,s)(Q(\boldsymbol{Y}_3)(s) + (F_1 + F_2)o(1) + f_2(s)\boldsymbol{X}(s))ds,$$

thus

$$(F_1 + F_2)\|\boldsymbol{y}(t)\| \leq \int_0^t K_3 \exp^{-c_3(t-s)}((F_1 + F_2)^2 Q(\boldsymbol{y})(s) + (F_1 + F_2)o(1) + F_2 K\|\boldsymbol{X}_0\| \exp^{-\lambda s})ds.$$

Finally,

$$\|\boldsymbol{y}(t)\| \leq \frac{K\|\boldsymbol{X}_0\|}{c_3} \exp^{-\lambda t} + \frac{K_3}{c_3} o(1) + (F_1 + F_2) \int_0^t K_3 \exp^{-c_3(t-s)}(Q(\boldsymbol{y})(s))ds.$$

This proves that $\|\boldsymbol{y}(t)\| = O(F_1 + F_2)$ and then, $\|\boldsymbol{Y}_3\| = o(F_1 + F_2)$, which concludes the proof of Proposition 10. □

7.7 Implementation on real data

To test this method on real data, the same sequence of depth images that what used to illustrate the geometrical flow velocity estimation method in Section 6.4 is processed. Embedded gyrometers , accelerometers and magnetometers provide angular velocity, specific acceleration and magnetic field expressed in the rigid-body reference frame $\mathcal{R}_{cam}$. The asymptotic observer (7.4) is implemented on this data, and provides estimates for $\boldsymbol{\omega_b}$, $\boldsymbol{a_b}$ and $\boldsymbol{v}$ plotted in Figures 7.1, 7.2 and 7.3 respectively. The noisy velocity measurements obtained by the geometrical flow estimation method are clearly filtered by this observer. As for the biases, it seems that the sequence is not long enough to enable asymptotic convergence towards constant values.

Loop processing of the sequence is thus performed as in Subsection 5.4.4: the results of the estimated biases values are plotted in Figures 7.4 and 7.5. The estimated biases on angular velocity are submitted to perturbations (the magnitude of these perturbations does not exceed $5.10^{-5}\text{rad}.s^{-1}$) but converge towards a mean value:

$$\widehat{\boldsymbol{\omega_b}}_{mean} = \begin{pmatrix} 1.025 \\ 0.075 \\ -0.50 \end{pmatrix} 10^{-3}\text{rad}.s^{-1}.$$

This value is compared with a rough mean computation of the angular velocity measured by the gyrometers during the 40 first seconds, when the system is known to stand still. This value is

$$\boldsymbol{\omega_b} = \begin{pmatrix} 1.04 \\ -0.21 \\ -0.43 \end{pmatrix} 10^{-3}\text{rad}.s^{-1}.$$

We conclude that the proposed observer can estimate biases up to a precision of about $3.10^{-4}\text{rad}.s^{-1}$ (about $60\text{deg}.h^{-1}$). The estimated biases on acceleration converge slowly towards a constant value (the magnitude of the perturbations does not exceed $5.10^{-4}\text{m}.s^{-2}$)

$$\boldsymbol{a_b} = \begin{pmatrix} 0.072 \\ -0.023 \\ -0.12 \end{pmatrix} \text{m}.s^{-2}$$

which is consistent with the usual bias values observed on sensors. The only possibility to validate the accuracy of this result consists of showing that it actually improves the linear velocity estimation: Figure 7.6 is a comparison of the initial filtered velocity (obtained by the regular processing of the sequence of data),and the filtered velocity obtained after a complete bias estimation(obtained by multiple loop processing of the same sequence). This figure shows that the velocity estimation is clearly improved, especially for the component along the y-direction. This is explained by the fact that velocity measurements along this direction are particularly noisy: the observer relies mostly on the accelerometers to predict the state, which is rarely updated by valid velocity measurements. Thus, a bias on the specific acceleration is more visible than for the components where valid velocity fixes are available.

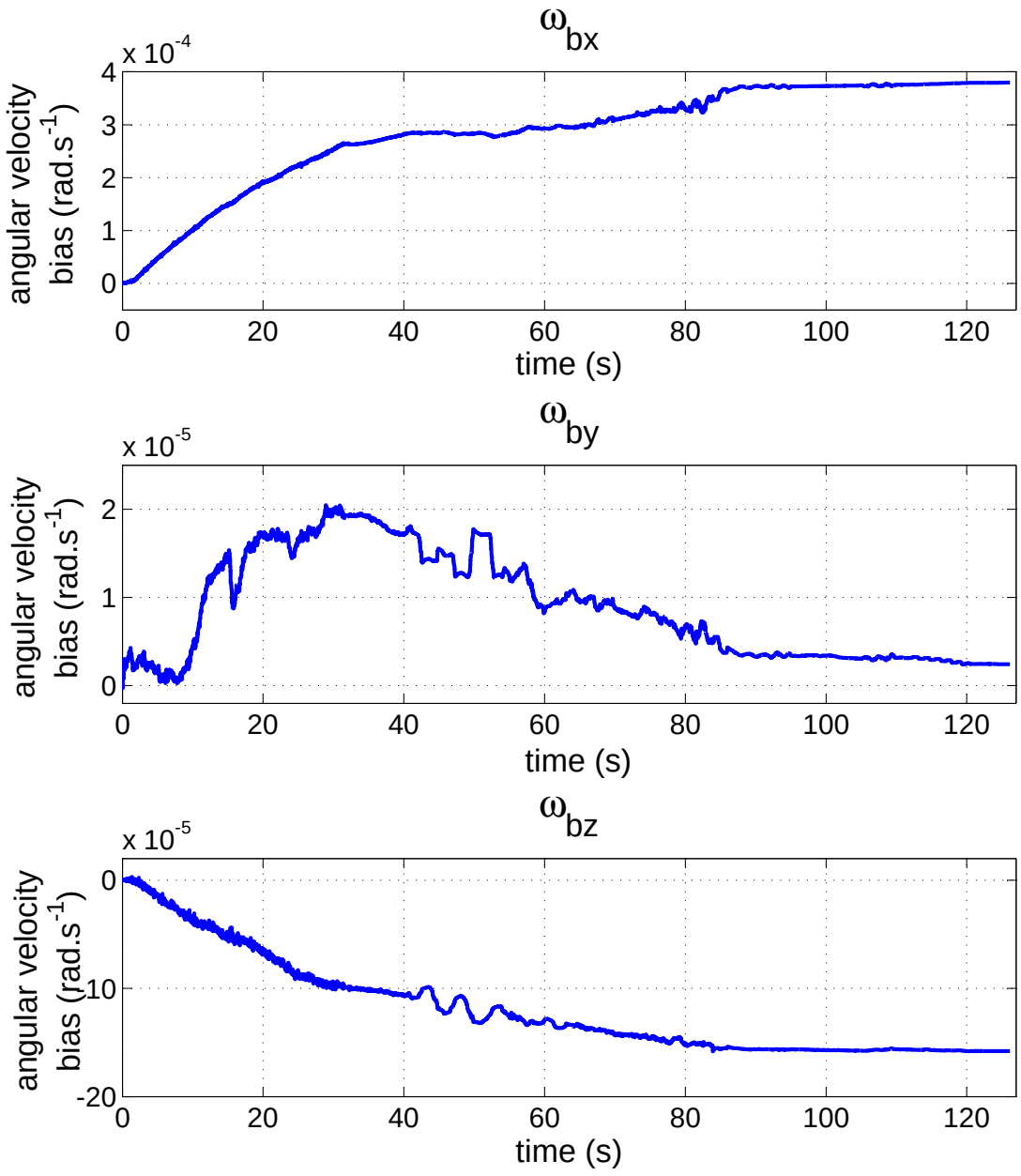


Figure 7.1: Bias estimation on angular velocity by the observer 7.4 processing RGBD data along with accelerometers, magnetometers, gyrometers and velocity measurements.

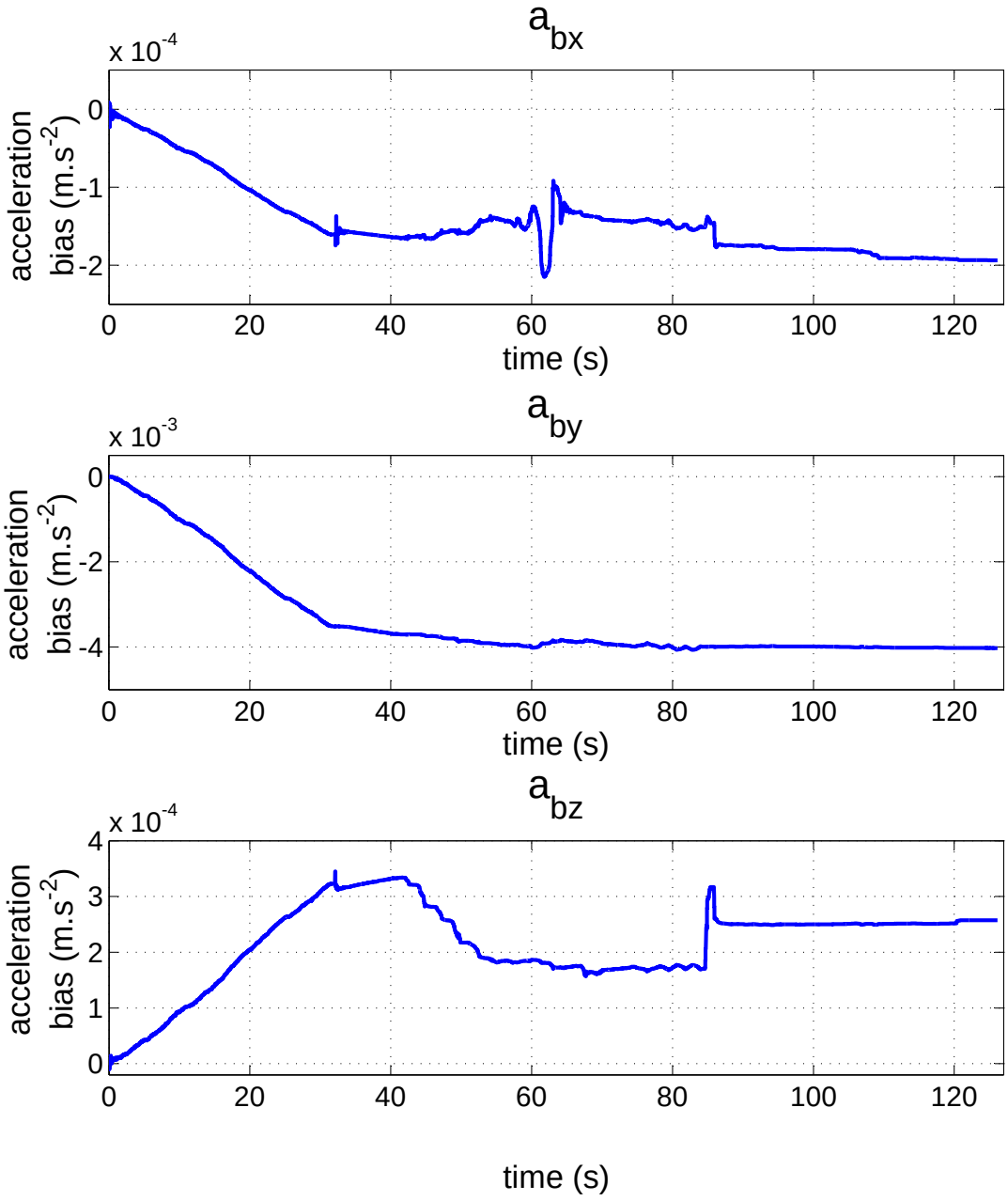


Figure 7.2: Bias estimation on acceleration by the observer 7.4 processing RGBD data along with accelerometers, magnetometers, gyrometers and velocity measurements.

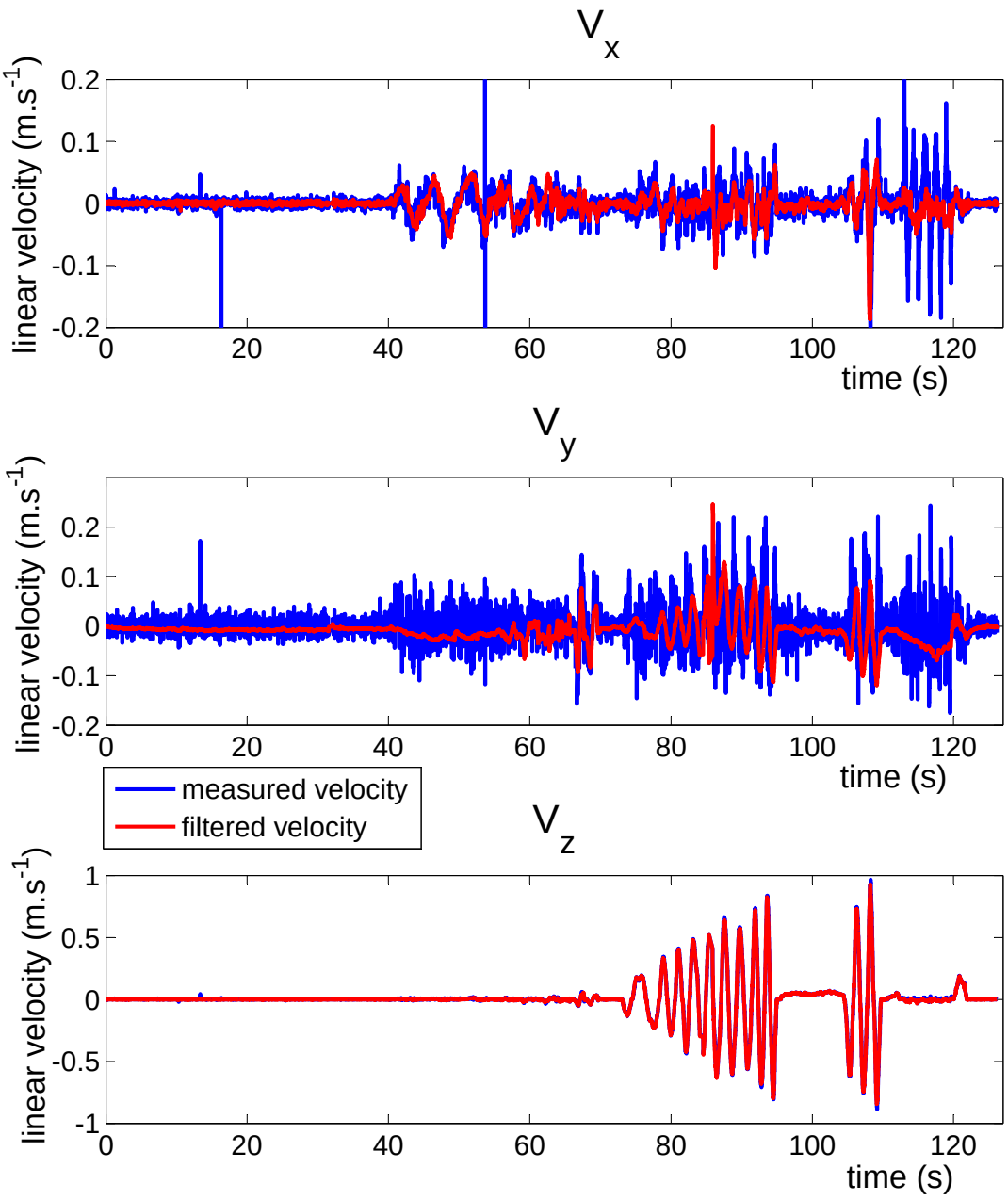


Figure 7.3: Filtering of the linear velocity measurements by the observer 7.4 processing RGBD data along with accelerometers, magnetometers, gyrometers and velocity measurements.

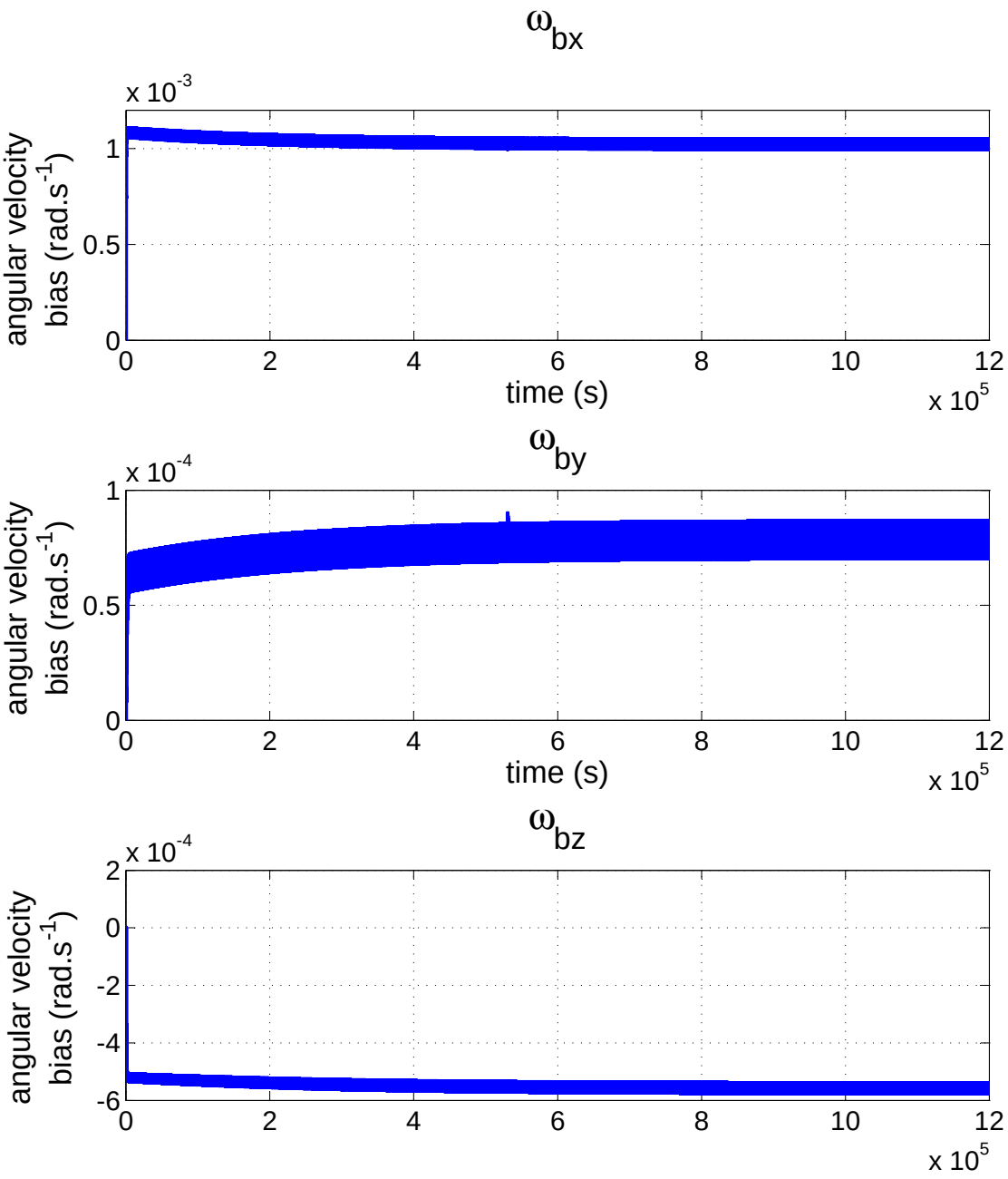


Figure 7.4: Bias estimation on angular velocity by the observer 7.4 applied to loop-processing of RGBD data along with accelerometers, magnetometers, gyrometers and velocity measurements.

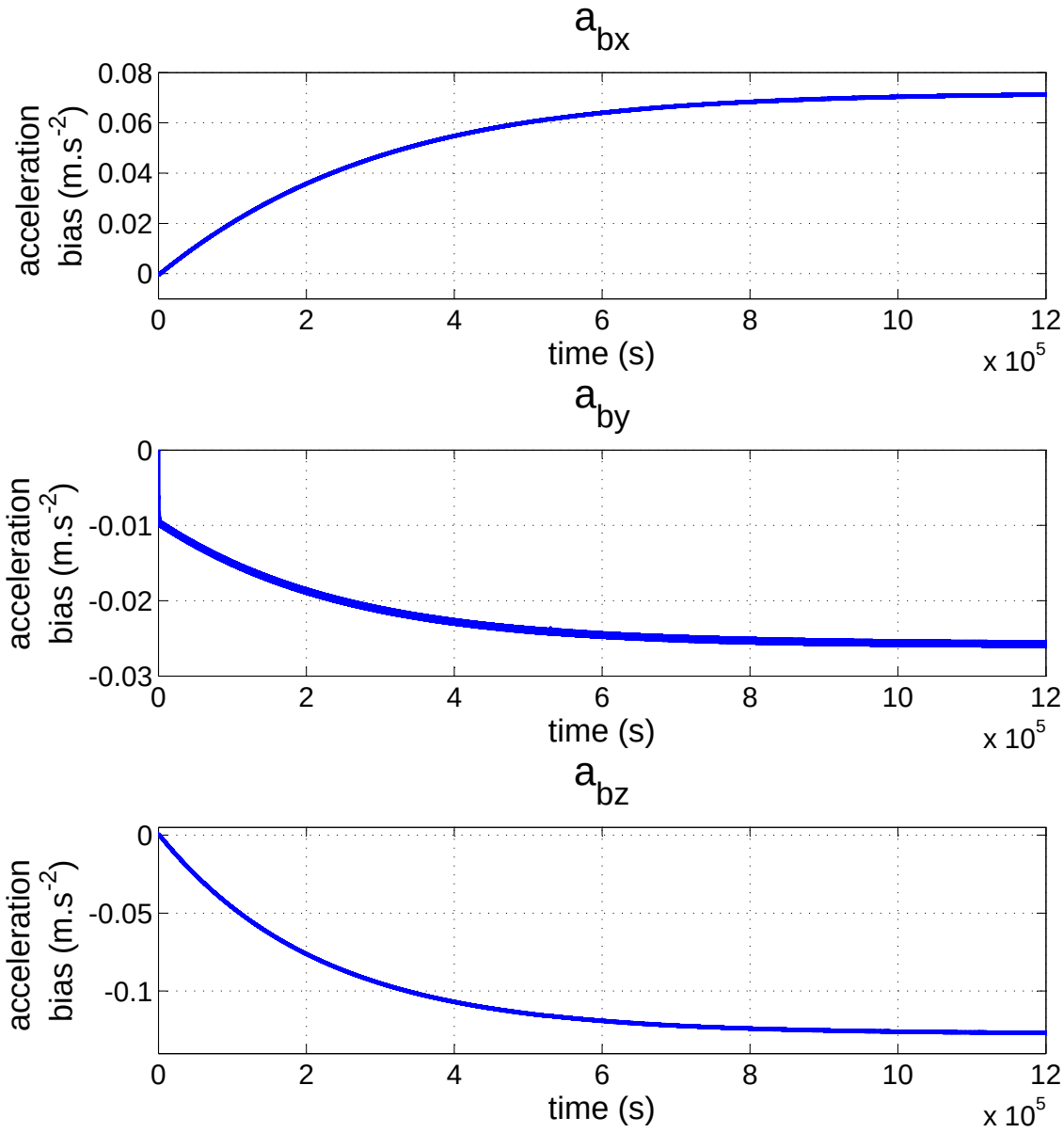


Figure 7.5: Bias estimation on acceleration by the observer (7.4) applied to loop-processing of RGBD data along with accelerometers, magnetometers, gyrometers and velocity measurements.

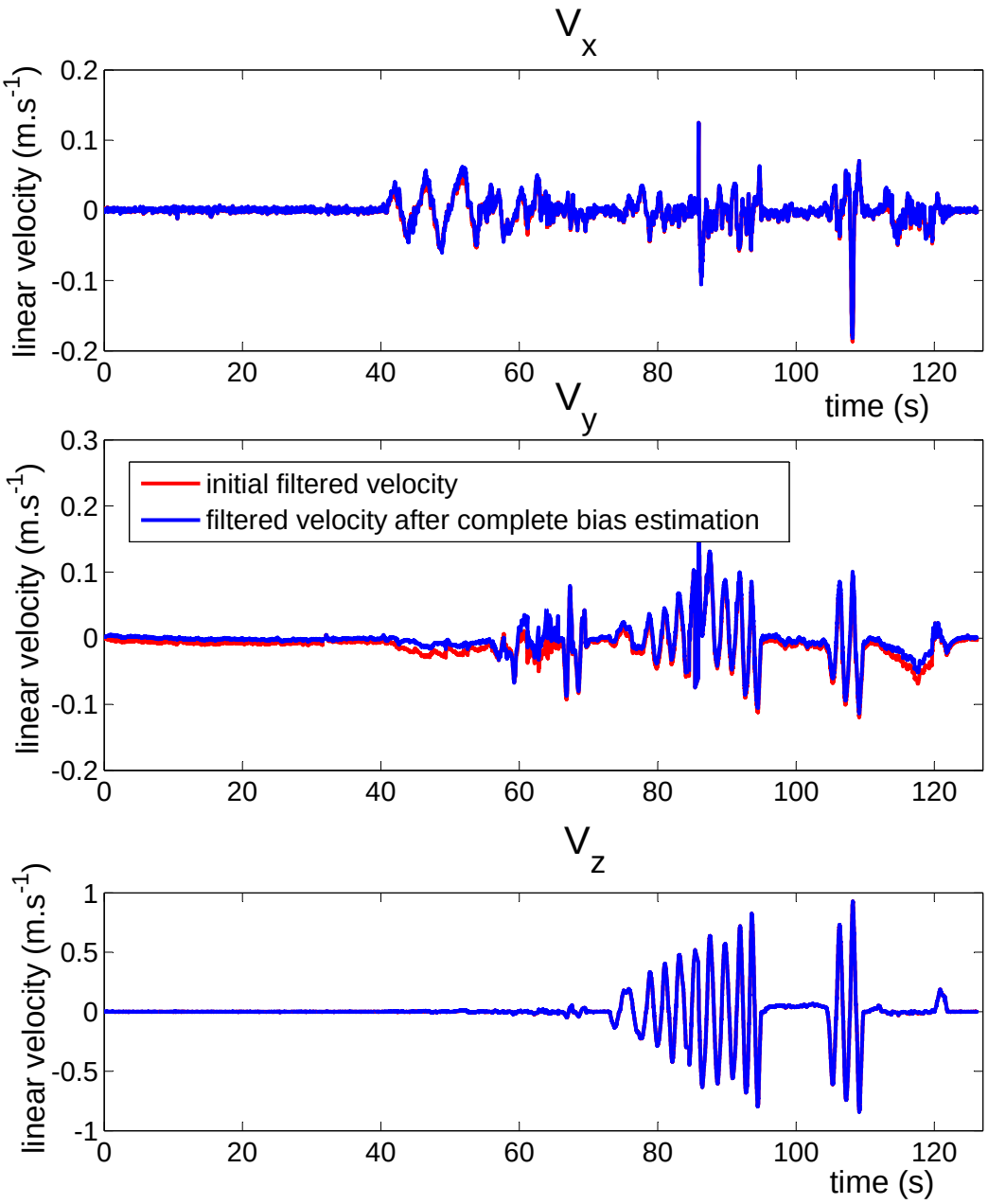


Figure 7.6: Comparison of the filtering of the linear velocity measurements by the observer 7.4 processing of RGBD data along with accelerometers, magnetometers, gyrometers and velocity measurements, using for the initial values of biases: zero (red); the values obtained by loop-processing (blue).

Part III

Simultaneous tracking and mapping for augmented reality

Localisation et cartographie simultanés pour la réalité augmentée

Chapter 8

Architecture of the original algorithm

PTAM

L'algorithme présenté dans cette dissertation est une modification de l'algorithme PTAM [Klein and Murray, 2007] décrit dans ce chapitre. La plupart des algorithmes de SLAM réalisent le suivi de trajectoire et la cartographie de l'environnement à chaque image [Davison et al., 2007, Eade and Drummond, 2006], ce qui implique qu'un suivi et une cartographie de mauvaise qualité vont se produire au même moment, et donc se dégrader mutuellement en un cercle vicieux. Au contraire, PTAM réalise un suivi et une cartographie parallèles *: la carte est optimisée dans un processus séparé, et seuls les bons points d'intérêt associés à un suivi de bonne qualité sont ajoutés à la carte. Une autre spécificité de cet algorithme est la méthode d'optimisation de la cartographie : tandis que la plupart des algorithmes de SLAM utilisent des méthodes de filtrage telles que Extended Kalman Filter (EKF) [Civera et al., 2008] ou filtrage particulaire [Montemerlo et al., 2003], qui sont toutes deux connues pour leur lourdeur en terme de temps de calcul, PTAM utilise l'ajustement de faisceaux, dont la complexité est en $O(m^3 + mn)$ où m est le nombre de poses de caméra et n est le nombre de points de la carte [Engels et al., 2006]. Cela permet de traiter en temps réel une très grande quantité de points d'intérêt visuels de basse qualité et faciles à détecter (plus précisément des coins Features from Accelerated Segment Test (FAST)), sans recourir à des descripteurs de points complexes. Cependant, la carte produite par cet algorithme n'est connue qu'à un facteur d'échelle près, ce qui est inévitable lorsque l'on traite des images produites par un système monoculaire. En outre, elle est uniquement composée de points d'intérêt discrets, ce qui ne permet pas de gérer en réalité augmentée les occultations qui peuvent avoir lieu entre un objet virtuel et l'environnement réel. Pour finir, cet algorithme a été conçu pour fonctionner sur de petits espaces de travail, alors qu'il serait nécessaire de cartographier une pièce entière.*

The algorithm presented in this dissertation is based on the original PTAM [Klein and Murray, 2007] described in this chapter. Most of the SLAM algorithms perform tracking and mapping refining at each frame[Davison et al., 2007, Eade and Drummond, 2006] which implies that low-quality tracking and mapping would occur at the same time and will

consequently degrade each other in a vicious circle. On the contrary, PTAM performs *parallel* tracking and mapping: the map is refined in a separate thread, and only good features associated to good tracking quality are added. Another specificity of this algorithm is the refinement method of the cartography: while most SLAM algorithms are based on EKF [Civera et al., 2008] or particle filtering [Montemerlo et al., 2003], which are both known to be a great computational burden, PTAM is based on bundle adjustment, whose complexity is $O(m^3 + mn)$ where m is the number of camera poses and n is the number of map points [Engels et al., 2006]. This enables to deal in real-time with a large amount of low-quality and easy-to-detect visual features (namely FAST corners), without the requirement of complex point descriptors. However, the map produced by this original algorithm intrinsically lacks a scale factor, which is unavoidable with monocular image processing. In addition, it is only composed of sparse features, which prevents the handling of occlusion of virtual objects in real images as required by augmented reality. Finally, it was first designed to map small workspaces, while we would like to be able to map at least a whole room.

8.1 Overview of the algorithm

PTAM is the acronym for Parallel Tracking and Mapping, an application initially proposed in [Klein and Murray, 2007] and later improved in [Klein and Murray, 2009], created by a research group of Oxford University. This open source system is one of the first to enable real-time processing: every step of tracking and mapping algorithms has been carefully chosen and optimized to ensure real-time capabilities of the system. In addition, trade off between performance and quality is adjustable, depending on the available resources. Only one camera is necessary (monocular system). As highlighted in Chapter 1, this yields an overall unknown scale factor on the produced map and in the localization within that map. Three distinct stages may exist:

- **Map initialization**: at the very beginning of the algorithm, this phase consists of the creation of a map. This phase is necessary since monocular system can not strictly speaking simultaneously estimate pose and structure without any external *a priori*. The map will be augmented all along the motion of the camera.
- **Tracking and Mapping generic algorithm**: once the map has been created, two threads run in parallel. Tracking consists of localizing (*estimating the pose of*) the camera with respect to the map. When new features of the environment appear along the motion of the camera, the tracking module notifies the mapping module to reinforce the cartography. If this is not the case, resources are available for the mapping module to *optimize the map*.
- **Relocalization**: sometimes, the tracking algorithm may not return a valid camera pose, generally because of the poor quality of the images (and consequently of the detected interest points). A specific method is implemented to enable the relocalization of the camera in the map.

These three stages are more precisely described in the following sections, to better underline the modifications implemented in the frame of this thesis. But first are introduced some general considerations.

The *map* consists of a set of camera poses and 3D-points, linked by *measurements*, or in other words by the position in the RGB images (when existing) of these 3D points at the camera poses. Feature tracking and SLAM as a general method consists of

1. detection of Two dimensional (2D) point features in the current view;
2. matching of point features with 2D features detected in other views or with 3D features of a map;
3. depending on the external inputs, estimation of camera pose (pose tracking), of 3D position of the 2D features (mapping), or of a combination of both (for example, homography or essential matrix estimation, as for map initialization).

PTAM uses FAST corner detector in the first step ([Rosten and Drummond, 2006]), which does not require as much processing time as more robust detectors such as Harris detector ([Harris and Stephens, 1988]) or SIFT extractor ([Lowe, 2004]). On the other hand, FAST does not provide a descriptor that could be used to match two 2D features (second step): matching is thus performed by patch comparison.

8.2 Map initialization

In order to start the SLAM loop, the algorithm must be initialized with some 3D features positioned in a fixed reference frame and a camera pose with respect to the same frame. Since the system is monocular, 3D positioning of features is performed by stereo vision. The procedure is the following:

- the user selects an initial pose,
- performs a slow motion (preferably a lateral translation) that allows for robust feature tracking,
- then selects a final pose.

Features are tracked from frame to frame all along the motion, and a relative pose estimation of the second pose with respect to the first one is computed, up to a scale factor. Two different methods are possible:

- **Essential matrix estimation**: if $\boldsymbol{x}$ and $\boldsymbol{x'}$ are the measurements of the same 3D point in two different views, their coordinates are related by

$$\boldsymbol{x'}^T \boldsymbol{E} \boldsymbol{x} = 0, \quad \text{with } \boldsymbol{E} = R[\boldsymbol{T}]_\times$$

  where $\boldsymbol{E}$ is the essential matrix, R and $\boldsymbol{T}$ are the difference of orientation and position (with the $SE(3)$ notation convention) between the two views, and $[\boldsymbol{T}]_\times$ is the matrix representation of the cross product with $\boldsymbol{T}$. $\boldsymbol{E}$, and then R and $\boldsymbol{T}$ (up to a scale factor) can be estimated by the five-point algorithm ([Nister, 2004],[Stewénius et al., 2006]) combined with RANdom SAmple Consensus (RANSAC) algorithm [Fischler and Bolles, 1981].
- **homography estimation**: if $\boldsymbol{x_i}$ and $\boldsymbol{x'_i}$ are the measurements of the same 3D points $\boldsymbol{X_i}$ in two different views, where the points $\boldsymbol{X_i}$ all belong to the same plane, their coordinates are related by

$$\boldsymbol{x'_i} = \boldsymbol{H} \boldsymbol{x_i}, \quad \text{with } \boldsymbol{H} = R + \frac{\boldsymbol{T} \cdot \boldsymbol{n}}{d}$$

where R and $\boldsymbol{T}$ have already been defined, and $\boldsymbol{n}$ and d are the normal and the distance to the plane, respectively. This requires for the user to start map initialization in front of a plane where features are present. The plane and its normal are estimated by RANSAC algorithm, then homography and R and $\frac{\boldsymbol{T}}{d}$ are estimated according to [Faugeras and Lustman, 1988].

Then, an external (arbitrary) scale factor is provided, and the robustly tracked features are positioned in the map by triangulation.

8.3 Pose estimation (tracking)

Once a map has been created, current pose estimation with respect to the map is possible, and is an essential step in the SLAM loop. Pose estimation is not given in a closed-loop form, but rather as the optimization of an initial pose. This requires

- a pose initialization (or pose prediction),
- a function, or cost, to optimize, (the cost must depend on the pose), and
- an optimization method.

The pose initialization is computed using a *motion model*. Since the system does not include velocity sensors or accelerometers, the motion model is arbitrarily chosen under the assumption of constant velocity with stability constraints (decaying alpha-beta model). This yields an estimate of linear and angular velocity $\boldsymbol{v}$ and $\boldsymbol{\omega}$, which are applied to the previous camera pose to compute a current pose prediction:

$$K_{pred} = \exp^{\sum_i \alpha_i \mathfrak{G}_i dt} K_{prev}$$

where the α_i are the components of $\boldsymbol{v}$ and $\boldsymbol{\omega}$ and the $\mathfrak{G}_i$ are the infinitesimal generators of $SE(3)$ (see Appendices B and C for more details). K_{prev} and K_{pred} denote the previous and the predicted camera poses (using the notations introduced in chapter 1). Finally, dt is the time interval between previous and current image acquisition.

From this predicted pose and the 3D points of the map, a set of potentially visible points is computed, and these points are projected on the image plane. In parallel, FAST detector is run on the current image, then the algorithm searches for matches between the projected points of the Potentially Visible Set (PVS) and the actual features of the image. The search is limited to a fixed region around the position of the projected features of the PVS. Figure 8.1 illustrates the PVS computation and the consequent match search. Details about FAST detector and patch scoring including warping and pyramidal search won't be given here, but can be found in the original publication ([Klein and Murray, 2007]). Once the match has been performed, the cost to minimize is defined as the projection error between the projected PVS and the detected features. Considering n pairs of matched features, x_d^j is the vector of coordinates in the image of the detected features and x_p^j is the vector of coordinates in the image of the matched projected point of the map, with $j \in (1, n)$. As described in Chapter 1, x_p^j is given in homogeneous coordinates by

$$\tilde{x}_p^j = P K_{pred} \tilde{X}_W^j$$

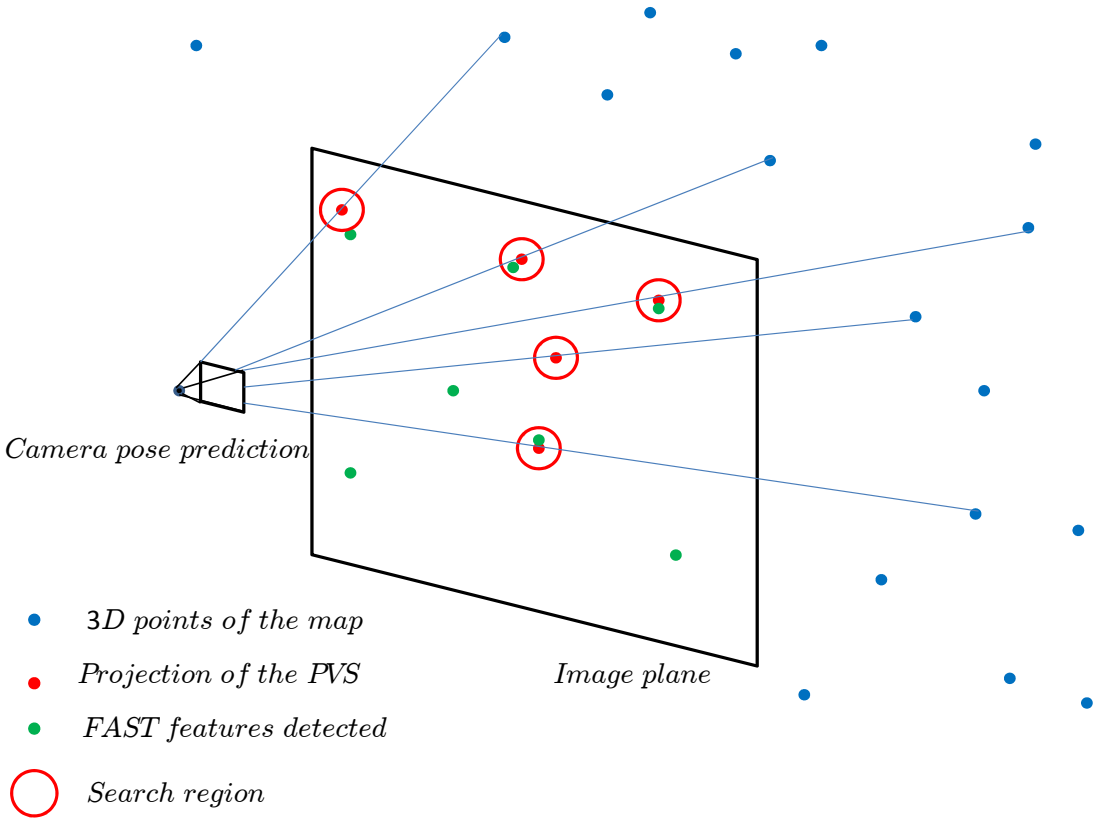


Figure 8.1: Projection of the Potentially Visible Set on the image plane and match search with dectected features.

where P is the intrinsic calibration matrix (under the assumption of a linear projection model, which can be adapted to non-linear models including distortions), and $\tilde{X}_W^j$ is the position of the corresponding 3D point of the map. The cost to minimize is given by

$$C(K) = \sum_j W_j \|\tilde{x}_p^j - \tilde{x}_d^j\|^2$$

where the W_j are weights iteratively computed to diminish the influence of *outliers* (defined as mismatched features) in the pose estimation (more details on these weights below). This method is known as *iteratively re-weighted least squares minimization*. Gauss-Newton iterations were employed for this non linear least-square minimization ([Hartley and Zisserman, 2000]). Only error, weights and Jacobian are required for this method.

- Error is as previously mentioned the reprojection error $\tilde{x}_p^j - \tilde{x}_d^j$.
- Weights are computed using any robust M-estimator function: for example, the Tukey's biweight function might be used , where the influence function parameters directly depend on the median of the errors distribution.
- To compute the Jacobian of the error with respect to the camera pose, the problem is formulated in terms of pose update, expressed as an element of the Lie algebra of $SE(3)$ and more specifically as the 6 components α_i of the pose update expressed with respect to the 6 infinitesimal generators $\mathfrak{G}_i$:

  $$K_{current} = K_{update} K_{pred}, \quad \text{with } K_{update} = \exp^{\sum_i \alpha_i \mathfrak{G}_i}, \quad \text{and } \alpha_i = \operatorname{argmin} C(\alpha_i).$$

  With this formulation, the derivative of the error with respect to the pose update is simply

  $$\sum_i \sum_j \frac{\partial(\tilde{x}_p^j - \tilde{x}_d^j)}{\partial \alpha_i} = \sum_i \sum_j P \mathfrak{G}_i K_{pred} \tilde{X}_W^j. \tag{8.1}$$

To summarize, the tracking process consists of

1. applying the motion model to the previous camera pose;
2. determining the PVS of map points;
3. finding currently visible matches with the projection of the PVS;
4. updating camera pose to minimize reprojection errors;
5. assessing the quality of the updated camera pose.

The last step of this process is essential for relocalization and mapping purposes. The number of map points projected and actually matched with FAST corners in the image is compared with the total number of points of the PVS. The ratio value is a measure of the quality of tracking. Below a certain threshold, current measurements can not be used to augment the map. Below an even lower threshold, the tracking is considered to be lost and the relocalization procedure is initiated.

8.4 Mapping

After initialization (described in Section 8.2), the map contains two *keyframes*, defined as specific and carefully chosen views, which contain the measurements (the 2D detected features) that link camera poses and 3D map points. Insertion of a new keyframe in the map enables the addition of new 3D points in the cartography by triangulation (Subsection 8.4.1): the precision of 3D positioning of these new points directly depend on the quality of pose estimation of the camera. The position of 3D points can be optimized when features have been detected in multiple views, along with camera poses associated to these views (Subsection 8.4.2).

8.4.1 Keyframe insertion

As the camera explores new parts of its environment, the map must be augmented to enable pose estimation: this is performed by keyframe insertion. A keyframe must satisfy the following criteria:

- the camera pose estimation associated to this view was assessed to be good;
- the time elapsed since previous keyframe insertion exceeds a given threshold;
- the linear distance to the closest keyframe's camera pose is sufficient to provide a good stereo baseline for triangulation (this distance depends on the mean depth of detected 3D map points).

If these conditions are met (verified by the tracking module), the mapping module aborts any operation other than keyframe insertion or places the keyframe in a waiting queue. Then keyframe is processed, which consists in thorough features detection (new features are considered now that real-time tracking constraints do not prevail). Another measurement of these features is necessary to compute their 3D position: these measurements and correspondences are searched in the closest keyframe by epipolar search, which consists of comparing the patch of a feature of the inserted keyframe to patches located along the epipolar line in the target view. Comparison is achieved through zero-mean sum of squared differences. When match is found, 3D position is computed by triangulation using the target keyframe camera pose and the inserted keyframe tracked camera pose.

8.4.2 Bundle adjustment

When no keyframe waits to be inserted in the map, the mapping module can achieve map optimization, which consists, as in pose update in the tracking module, of minimizing the difference between the projection of 3D map points in keyframes and their actual measurements. But, on contrary with pose updates, the minimization is performed by variations of both camera poses and 3D map points positions: this method, based on Levenberg-Marquadt algorithm [Marquardt, 1963], is called bundle adjustment and has been extensively studied (see [Hartley and Zisserman, 2000] for textbook implementation and [Snavely et al., 2008] for an example of application).

The main steps of bundle adjustment are recalled in the following, but first is summarized Levenberg-Marquadt iterative parameters estimation method.

Consider a functional relation that links parameters Π and measurements X, such that $f(\Pi) = m$. The goal is to give an estimate of Π that best fits the model given the measurements, starting from a initialization parameter Π_0. The residual error is defined as $\epsilon(\Pi) = f(\Pi) - m$. The goal is to minimize the (possibly weighted) squared error $\|\epsilon(\Pi)\|^2$. When f is nonlinear, the iterative method seeks for a small update δ_Π of the parameter, so that $f(\Pi + \delta_\Pi)$ can be approximated by $f(\Pi) + J\delta_\Pi$ where J is the Jacobian of f with respect to Π. The goal is to find the minimal argument δ_Π of $\|\epsilon(\Pi + \delta_\Pi)\|^2 = \|\epsilon(\Pi) + J\delta_\Pi\|^2$. First-order stationary equation yields the result as the solution of the following *normal* equation

$$J^T J\delta_\Pi = -J^T\epsilon(\Pi) \text{ (Gauss-Newton iterative method).}$$

Levenberg-Marquadt's method augments the normal equation by

$$(J^T J + \lambda I)\delta_\Pi = -J^T\epsilon(\Pi)$$

or even

$$(J^T J)_\lambda \delta_\Pi := (J^T J + \lambda \text{diag}(J^T J))\delta_\Pi = -J^T\epsilon(\Pi)$$

where the parameter λ varies depending on the achieved iteration.

This method is applied to bundle adjustment using the relation

$$\tilde{x}_i^j = PK_i\tilde{X}_W^j = f_{2D}(K_i, \tilde{X}_W^j). \tag{8.2}$$

The measurements formally consists of the concatenation of any measurement $\tilde{x}_i^j$ that could exist in a given set of keyframes. The vector of parameters Π is the concatenation of camera poses K_i associated to the keyframes (and more specifically poses updates expressed as the components of the Lie algebra element with respect to infinitesimal generators like in pose estimation 8.3) and 3D map points positions $\tilde{X}_W^j$ associated to measurements. Let us define

- J_K as the Jacobian of f_{2D} with respect to camera pose (see (8.1)),
- J_X as the Jacobian of f_{2D} with respect to 3D points position (computed by chain rule differentiation, taking into account the pinhole projection first, and then the distortion)
- δ_K as the update of camera poses,
- δ_X as the update of 3D points positions,
- ϵ is the reprojection difference, in other words the difference between the 2D position of the measurement and the 2D position of the projection of the associated 3D point in the image plane of the associated view.

The usual normal equations are given in the form

$$\begin{pmatrix} J_K^T J_K & J_K^T J_X \\ J_X^T J_K & J_X^T J_X \end{pmatrix} \begin{pmatrix} \delta_K \\ \delta_X \end{pmatrix} = \begin{pmatrix} J_K^T \epsilon \\ J_X^T \epsilon \end{pmatrix}$$

and transformed by Levenberg-Marquadt augmentation by

$$\begin{pmatrix} (J_K^T J_K)_\lambda & J_K^T J_X \\ J_X^T J_K & (J_X^T J_X)_\lambda \end{pmatrix} \begin{pmatrix} \delta_K \\ \delta_X \end{pmatrix} = \begin{pmatrix} J_K^T \epsilon \\ J_X^T \epsilon \end{pmatrix}.$$

The solution is given in a triangular form. First, δ_K is given as the solution of

$$((J_K^T J_K)_\lambda - J_K^T J_X (J_X^T J_X)_\lambda^{-1} J_X^T J_K)\delta_K = J_K^T \epsilon - J_K^T J_X (J_X^T J_X)_\lambda^{-1} J_X^T \epsilon$$

then δ_X is given as the solution of

$$(J_X^T J_X)_\lambda \delta_X = J_X^T \epsilon - J_X^T J_K \delta_K.$$

The minimization does not equally take into account all the measurements: weights are iteratively computed using a M-estimator influence function such as the Tukey's biweight objective function to reject the outliers influence in the optimization process (as for pose update, see section 8.3). The error to minimize is $\|\epsilon^T W \epsilon\|$, where W is a diagonal weighting matrix. The optimization procedure is modified accordingly.

8.5 Relocalisation

In case where the tracking was considered of poor quality for three successive frames, the system is declared "lost". A recovery procedure is initiated, described in [Williams et al., 2007]. The goal is to give a pose estimate that can be used as tracking initialization. The fastest solution is to compare a light version of the current image to similar versions of the views associated to keyframes of the map. Comparison is achieved through zero mean sum of squared differences. The "light" version is obtained by down sampling and blurring of the image and are named Small Blurry Image (SBI).

Chapter 9

Fusion of magneto-inertial, visual and depth measurements

Fusion de données RGBD, de mesures magnéto-inertielles et de mesures de vitesse par la méthode du flot géométrique.

Dans ce chapitre, nous décrivons comment PTAM a été amélioré par l'apport de données magnéto-inertielles et de profondeur : l'ambiguité du facteur d'échelle devient obsolète, tout comme le problème de l'occultation. En outre, les mesures de vitesse linéaire apportées par la méthode du flot géométrique (voir le Chapitre 6) peuvent être utilisées en même temps que les accéléromètres pour filtrer les mesures de position très bruitées fournies par l'algorithme de vision, et ainsi fournir une estimation de la position même quand les mesures visuelles ne sont pas disponibles. Une partie de ces améliorations ont été réalisées directement à Sysnav : toutes ont été possibles grâce à leur aide. L'algorithme initial PTAM a été implémenté et adapté pour être complété et rendu plus robuste par des capteurs supplémentaires, qui peuvent être classés en deux catégories : les capteurs magnéto-inertiels*, plus précisément les accéléromètres, les gyromètres et les magnétomètres ; le* capteur de profondeur*, qui est une sous-partie du capteur RGBD. Nous décrivons ici les capteurs et leur synchronisation, puis les améliorations apportées par la fusion d'un algorithme de vision et de données complémentaires.*

In this chapter, we describe how PTAM is improved when aided by magneto-inertial and depth data: the scale factor ambiguity becomes obsolete, as well as the occlusion issue. In addition, linear velocity measurements from our geometrical flow method (see Chapter 6) can be used along with accelerometers to filter the noisy position measurements provided by the vision algorithm, and to provide a position estimate even when vision measurements are not available. Part of these improvements were performed by Sysnav company: all of them were made possible thanks to their help. The original algorithm of PTAM was implemented and adapted to be completed and made more robust by additional sensors, that can be separated in two classes: *magneto-inertial sensors*, namely accelerometers, gyrometers and magnetometers; *depth sensor*,

as part of the RGBD sensor. We describe the sensors and their synchronization, then the improvements provided by fusion of a vision algorithm with complementary informations.

9.1 Sensors and synchronization

9.1.1 Depth sensor

We make use of the RGBD sensor integrated in the Kinect device developed by Microsoft for the Xbox 360 video game console and first released in November 2010. Although the technology on which this sensor is based has been known for years, this device is a technological outbreak in the way data is provided, and because of its relatively low cost. At 24.8 Hz, two images of a resolution of 640 by 480 pixels are jointly provided. The first image is a regular RGB image, but the second image is a *dense* depth image aligned with the RGB image. This yields that to any object in the field of view of the camera, a color and a 3D position (with respect to the camera frame $\mathcal{R}_{cam}$) is provided: on one hand this is a direct solution to the scale factor issue of monocular systems and on the other hand this theoretically allows to bypass any triangulation limitation. The image and depth sensors are actually regular CMOS sensor, the depth sensor being equipped with an infrared filter. Along with them, an infrared emitter sends a given pattern of infrared "dots". This pattern is projected onto the environment: depending on the depth of the objects on which the dots are projected, the pattern is more or less distorted. The distorted pattern is then acquired by the infrared sensible CMOS sensor, and disparity is computed from the distortion. Depth is then computed from this disparity, and stereo rectification enables to align RGB and depth images. Concerning the depth sensor, one can neither decide of the date of data acquisition, nor precisely know this date : one can only assume that acquisition rate is constant, that no data is lost during transmission by the USB or even the sensor's driver, and that the depth image and the RGB image of a specific frame were acquired at the same time. An example of RGBD data is given in Figure 9.1, in which areas where depth data is not valid can be seen (black areas, the depth data associated to these pixels is zero). Depth data is typically corrupted in presence of sun light or reflective surfaces, and of course if the environment is out of the range of the sensor (0.5-7 meters).

9.1.2 Magneto-inertial sensors

Over the past decades, MEMS have become more popular and more accurate, because they are small, they are cheap and they can be sufficient when high precision data is not required. For example, they are present in any current smartphone. The 3 accelerometers and 3 gyrometers used in our system are MEMS, gathered in a 6-DOF IMU distributed by Epson. It measures a few millimeters, consumes only 6 mA, and is operated at 125Hz. In addition to these sensors, magnetometers are Anistropic Magnetoresistances (AMR), and also operate at 125Hz. Synchronization between accelerometers, gyrometers and magnetometers is achieved independently of the depth sensor through a global numerical trigger, which ensures that data was acquired for all magneto-inertial sensors at the same date.

Figure 9.1: An example of aligned RGB and depth images acquired by the Kinect device

9.1.3 Synchronization

RGBD data and magneto-inertial data can be seen as two separate devices (RGBD sensor and magneto-inertial sensor) in which various data types are internally synchronized. The magneto-inertial sensor sends to the RGBD sensor a signal synchronized with magneto-inertial data that encodes a value varying from one frame to the next. The depth sensor's hardware has been modified to acquire this signal and to associate it with the current RGBD frame. When a RGBD frame sent by the RGBD sensor is received by the master system, synchronization with a magneto-inertial data frame is achieved by comparison of the signals.

9.2 Map initialization

Map initialization becomes easier when scale factor ambiguity is removed. The first 3D positions of map points can be computed from one single view: the position of a 2D point in a view gives the perspective line on which the 3D point is located, and the depth image gives the position on this line (to be precise, the position along the optical axis). Since accuracy of localization crucially depends on the quality of the map, only very distinctive map points are initialized, which is achieved through robust detection and description of features in several frames, and consistency check between several frames. In addition, map points insertion is restricted to features for which depth data is valid.

9.3 Mapping

Using data sources other than visual data can be of great profit for the mapping module. First, insertion of new map points is greatly simplified and made more robust when triangulation is not necessary (like in map initialization). Secondly, depth measurements can be added in the overall optimization by bundle adjustment.

9.3.1 Keyframe insertion

When the user explores large areas and does not want to be restricted to small workspaces for which the original algorithm was first designed, the principal differences arise in

- the number of keyframes and map points of the cartography;
- the range of camera poses admissible: in particular, orientation is submitted to larger variations when the user intends to map a whole room or building.

The problem of map size is critical in bundle adjustment, for which processing time evolves in $O(m^3 + m^2 n)$ where m is the number of keyframes and n the number of map points of the cartography: it must be addressed as soon as the decision on keyframe insertion is taken. This would tend to increase the value of the threshold of the linear distance from which a new keyframe is considered as necessary (see Subsection 8.4.1). In parallel, camera orientation is submitted to large variations: it is natural when exploring a large environment to turn the head to see as most new areas as possible. But precisely, orientation variation induces large view variations, which means that the number of features present in two consecutive views is likely to significantly decrease, possibly up to the point where no map points are visible in the current view: this leads then to the loss of tracking. In pure visual SLAM algorithms, a trade off is necessary to decide when a new keyframe should be inserted: the linear distance between the closest keyframe of the cartography must be sufficient to enable good triangulation quality, unless large variation of orientation occurs: in this later case, map points must be added to the map as soon as possible to ensure tracking continuity.

When depth data is available, the constraints on minimal linear distance between keyframes is looser, since triangulation is not anymore necessary. Criteria used to decide whether a keyframe should be inserted are changed according to these considerations:

- linear distance between keyframes must be over a minimal threshold to limit the map size growth, and below a maximal threshold to ensure good correspondence between detected features and sufficient view superposition;
- difference of orientation between keyframes must be as small as possible to ensure good correspondence between detected features and sufficient view superposition;
- difference of orientation between keyframes must be below a maximal threshold;
- if difference of orientation between the current view and all existing keyframes is over a minimal threshold, the current view should be inserted as soon as possible, since tracking is likely to be lost soon.

Map points insertion process depends on depth data availability. Considering a given 2D detected feature of the current view: like in the original algorithm, a feature match is searched on the epipolar line of the target view (the closest from the standpoint of linear distance), then the 3D position of the corresponding map point is computed:

- if depth data is available for both 2D features position and corresponding 3D positions are close, the consequent 3D position is the mean of both positions (expressed in the world frame)
- if depth data is available for neither of 2D features position, 3D position is computed by triangulation;

– if depth data is available for only one 2D feature position, the corresponding 3D point is reprojected in the other view; if this reprojection is close to the associated detected feature, the 3D point is added to the map.

9.3.2 Depth data in bundle adjustment

Bundle adjustment as described in Section 8.4.2 was only adapted to 2D measurements. But with the use of depth sensor, 3D measurements are available. One of the main weaknesses of bundle adjustment lies in the continuous drift of the scale factor: as camera poses and map points positions are simultaneously optimized, both can slowly corrupt each other if nothing stops that vicious circle. 3D measurements can put an end to this drift since it intrinsically contains a scale factor. To our knowledge, this method has first been published in [Scherer et al., 2012]. We describe here how it is adapted to our application. When depth data is available for a 2D feature detected in several different keyframes, a 3D measurement is automatically created, which is related to camera pose and 3D map point world position by

$$\tilde{X}_i^j = K_i \tilde{X}_W^j = f_{3D}(K_i, \tilde{X}_W^j). \tag{9.1}$$

which is similar to the 2D equation of projection (8.2). To use bundle adjustment in a similar way, one has

– to define the error $\|\epsilon\|^2$ to be minimized,
– to explicit the Jacobian of f_{3D} with respect to the camera pose K_i,
– to explicit the Jacobian of f_{3D} with respect to the map point position $\tilde{X}_W^j$.

Then bundle adjustment as described in Section 8.4.2 can be readily used.

The error to be minimized is simply the Euclidean norm of the difference ϵ between the 3D measurement and the 3D estimate computed from the 3D world position and the camera pose.

The Jacobian J_K of f_{3D} with respect to the camera pose is computed as in (8.1) by use of the infinitesimal generators of the Lie algebra of $SE(3)$, omitting the projection matrix P.

Finally, the Jacobian J_X of f_{3D} with respect to the 3D map point world position is extremely easy to compute, as it simply is the rotation term of the element of $SE(3)$ representing the camera pose K_i.

2D measurements and 3D measurements are used together in the optimization procedure: the size of the Jacobian matrices relative to f_{2D} and f_{3D} are different (sizes 2×6 and 3×6 respectively for J_K, and sizes 2×3 and 3×3 respectively for J_X), but they are in the algorithm always combined together to compute the pose updates δ_K (vectors of length 6) and map points position updates δ_X (vectors of length 3). However, the weights given to measurements of different types must be normalized so that their influence in the optimization is the same.

9.4 Pose estimation

The pose estimation is in its principle the same than in the original algorithm: prediction of the current camera pose from the previous camera pose, projection of the PVS on the image plane, pose update by minimization of the reprojection error. It has been modified to integrate

magneto-inertial and depth data in the pose prediction and in the computation of the PVS steps. The pose prediction consists of two distinct processes: linear velocity estimation ($\boldsymbol{v}$ the linear velocity expressed in camera frame $\mathcal{R}_{cam}$) and attitude estimation (quaternion q that expresses the passage from the world reference frame $\mathcal{R}_W$ to the current camera frame $\mathcal{R}_{cam}$).

Attitude estimation is achieved through the processing of measurements of magnetometers, accelerometers and gyrometers, as described in [Martin and Salaün, 2010], along with the estimation of gyrometers biases. Biases and scale factors of accelerometers are independently previously estimated. The principal assumption of this work consists of neglecting the linear acceleration of the moving system: $\partial_t \mathbf{V}(t) = 0$. This yields (see Appendix C.2 for notations and details on the usual motion equations under flat Earth assumption, e.g. (C.5)) that the specific acceleration $\boldsymbol{a}$ is simply a measurement of the gravity acceleration expressed in the camera frame ($\mathcal{R}_{cam}$) $q^* \mathbf{A} q$. In parallel, magnetometers provide the measurements of the magnetic field expressed in the camera frame $q^* \mathbf{B} q$ and gyrometers enable to express the attitude dynamics (C.4). An invariant non-linear observer has been designed to provide attitude and gyrometers biases estimation, for which local convergence is ensured. The difference of orientation between previous and current RGBD frames is then estimated (strictly from *magneto-inertial measurements*) and applied to the previous camera pose *estimated by visual measurements* to provide current attitude prediction.

Linear velocity estimation is achieved through depth images processing, as described in Chapter 6. This yields a linear displacement estimation, applied to the previous camera pose (with the help of attitude estimation to express the linear displacement in the world camera frame $\mathcal{R}_W$) to provide a prediction a camera pose position. The pose prediction algorithm is summarized in Figure 9.2: data, state variables and algorithms steps are respectively depicted in blue, green and red to highlight the data flow.

Computation of the PVS is improved using the magnetic field measurement. To each map point is attached the value of the magnetic field measured by he system at the time when this map point was first observed. At a given time, a map point can only be added to the PVS if this magnetic field vector and the current magnetic field measurement are close enough. This criterion enables to reject in a simple way the map points that belong to a non-visible surface. Consider for example a map point on a wall of a room; if the camera user goes to the next room and turns around to look at the other face of that wall, the map point is strictly speaking in its field of view since occlusion is not taken into account in the PVS construction. Yet, the u-turn operated ensures that magnetic field measurements associated to the two views are opposite: thanks to this selection criterion the map point will not be added to the PVS.

9.5 Tracking quality and relocalization

Sources of localization other than images are of great help when the quality of RGB data is too poor to enable robust feature detection and matching, which often occurs when the camera is submitted to high dynamics, or simply when the texture of the environment is too uniform. For this purpose, magneto-inertial and depth data can "take over" to fill data in when algorithm based on pure vision is not sufficient. This issue is addressed in the next section, dealing

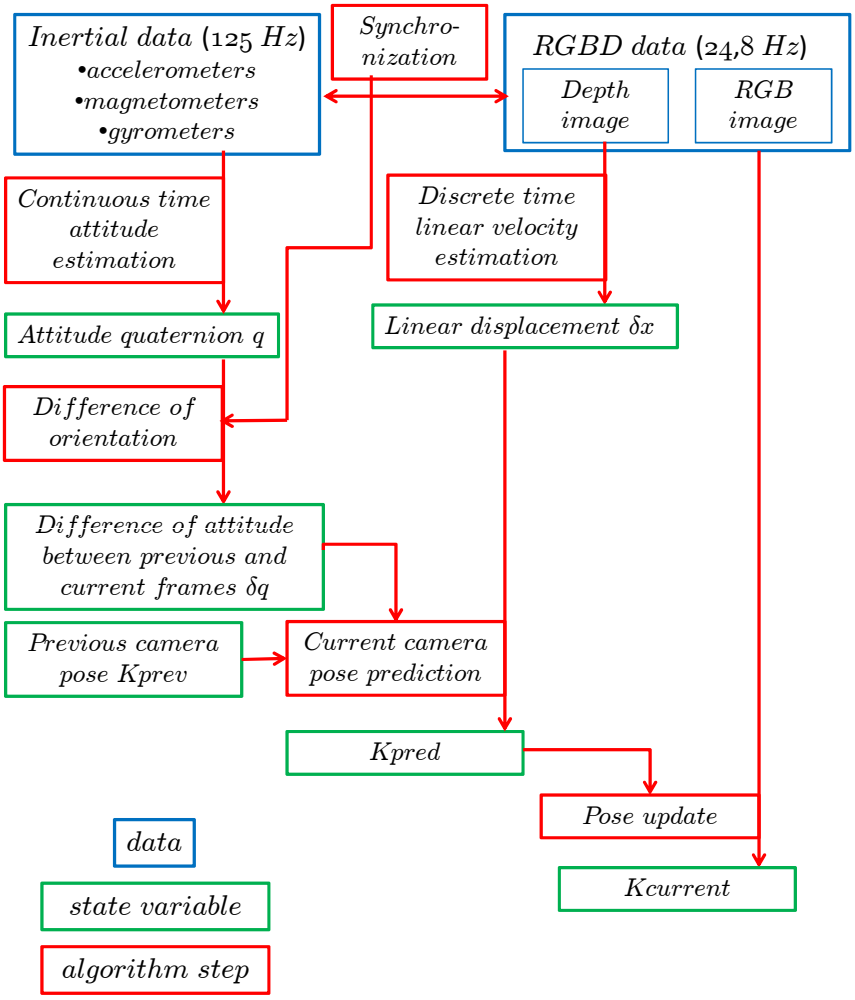


Figure 9.2: Representation of the pose estimation algorithm modified to take into account depth and magneto-inertial data

with localization for augmented reality. This section describes how magneto-inertial data, as redundant data, can assess of the tracking quality in one hand, and help in the recovery process in the other hand.

9.5.1 Assessing the visual tracking quality

Recall from Section 8.3 that in the original algorithm, tracking quality was estimated from the ratio of features of the PVS actually detected in the current frame. Along with this criterion, the absolute number of detected features is used: the minimum number of matched features necessary for pose estimation is theoretically 5. To these two purely visual criteria, a comparison of vision and magneto-inertial data is added.

Formally, the quality of visual tracking is judged from the standpoint of attitude accuracy. As we already mentioned in Section 9.4, processing of magneto-inertial data enables attitude estimation of the camera. This attitude is compared to the attitude estimated by visual tracking *on a relative basis*. Indeed, accelerometers enable the estimation of gyrometers biases, apart from the bias about the vertical axis. Magnetometers eliminate this ambiguity under an assumption of homogeneous magnetic field, which is generally not satisfied indoor. This yields a slow (bounded time-derivative) drift of the pitch angle: thus, pitch estimation lies in a cone of uncertainty, linearly growing in time.

The difference of attitude between the current time t and a given anterior time t_0 is computed by magneto-inertial attitude estimation (R_{in}) and by visual attitude estimation (R_{vis}). These attitude variations are then compared to each other. If they are close and stable in time during the time interval (t_0, t), the visual quality is considered to be good and the current frame will be used for the next frames for relative attitude computation (t will replace the anterior time t_0). Since vision is not anymore the only localization data, visual tracking loss is not as bad as in the original algorithm: criteria for good tracking quality are purposely difficult to meet to make sure that "bad" views are not chosen as keyframes, which would corrupt the entire map. As a consequence, tracking loss is more frequent and the relocalization procedure is optimized to be as robust and fast as possible.

9.5.2 Relocalization procedure

In the original algorithm, tracking can only be recovered if the camera is close enough to the camera pose of a keyframe of the cartography, which is a large limitation. On the other hand, the multiplication of the number of keyframes must be limited to avoid memory overload and large processing time devoted to optimization. This is why a new structure has been introduced, named Fast Small Blurry Image (FSBI). A FSBI is much lighter than a keyframe, as it only contains a SBI, a camera pose and a magnetic field measurement. A new FSBI is created when tracking is not lost and provided that a minimal linear distance of a few millimeters exist with respect to the already existing FSBIs. These small structures are kept in memory all along the camera motion and organized with respect to camera pose orientation. When tracking is lost, an attitude is estimated using magneto-inertial measurements: a difference of magneto-inertial attitudes since the last time when quality was considered to be good is computed and applied to

the last visual attitude. From this predicted attitude, a set of FSBIs are selected and compared in various ways to the current view:

1. the current and FSBI's magnetic field measurements are compared ;
2. if they are close enough, the SBIs corresponding to the current and to the selected FSBI are compared;
3. if they are close enough, a difference of camera poses between the FSBI and the current view is estimated by warping of the current SBI;
4. step 3 provides a camera pose estimate used as the initialization of tracking procedure (PVS projection and pose update);
5. the resulting camera pose attitude is compared to magneto-inertial attitude;
6. if they are close enough, tracking is recovered; otherwise, the FSBI selected in step 3 is removed from the list of FSBI to compare the current view with, and the procedure starts over from step 1.

9.6 Localization and augmented reality

The objective of augmented reality is to inlay the image of a virtual object in a real image, in a realistic way. Realistic can be defined using the following criteria:

- **occlusion:** the virtual 3D object (or part of it) must be hidden in the image when it is placed behind real objects;
- **relative displacement:** real image and virtual object must evolve in a consistent way along the camera motion;
- **stability:** if the camera user is immobile, the virtual object must appear immobile in the image;
- **persistence**: the image of the virtual object must continuously appear in the image, independently of the tracking state.

The first criterion involves a complete current knowledge of the geometry of the visible scene, which can be given by the complete and dense cartography of the environment. However, the computational burden associated to such a goal would prevent any real-time processing. The depth image actually gives a complete and easy solution: for a given position of the virtual object in the current camera frame $\mathcal{R}_{cam}$, occultation is immediate by comparison of the current real depth (measured) and the potential virtual depth (computed).

The three last criteria are associated with the problem of localization: indeed, for a given position of the virtual object in an external reference frame (the world $\mathcal{R}_W$), its position in the current view only depends on the current camera pose in the world $\mathcal{R}_W$. This means that localization must be accurate, stable and continuous (respectively, following the three last criteria), which can not be satisfied using visual data only since it implies ambiguity of the scale factor and does not prevent a tracking loss. We will describe how fusion between visual, magneto-inertial and depth measurements may solve this issue. Camera pose is described by a unitary quaternion q and a position X_C in the inertial reference frame $\mathcal{R}_W$. Its state is completed by a linear velocity $\boldsymbol{v}$ expressed in the camera frame $\mathcal{R}_{cam}$.

Like for pose prediction, the camera pose estimation is decoupled between attitude estimation and position estimation.

Figure 9.6 shows an implementation of the augmented reality system, which is intended to be worn on the head: the RGBD sensor is placed right on top of the eyes, and augmented reality glasses are fixed in front of the eyes, to project the augmented images. The magneto-inertial sensors lie on top of the head: all components are linked to an external master system.

Figure 9.3: Implementation of the augmented reality system, which is intended to be worn on the head: the RGBD sensor and augmented reality glasses are fixed respectively on top and in front of the eyes, to acquire real images and project the augmented images. The magneto-inertial sensors lie on top of the head: all components are linked to an external master system.

9.6.1 Attitude estimation

Attitude estimation is achieved using magneto-inertial and visual measurements. Magneto-inertial measurements (namely gyrometers, magnetometers and accelerometers) provide debiased measurements of angular velocity $\boldsymbol{\omega_m}$. Vision provide attitude measurements expressed by quaternions q_V. The proposed non linear observer was inspired by invariance theory (see [Bonnabel et al., 2008] for details on invariance theory, see [Mahony et al., 2008] for a $SO(3)$-invariant attitude observer):

$$\begin{cases} \partial_t \widehat{q} = \frac{1}{2}\widehat{q}\boldsymbol{\omega_m} + \widehat{q}K(\widehat{q}^* q_V - 1) \\ \widehat{q}(0) = q_V(0) \end{cases} \tag{9.2}$$

where 1 designates the quaternion $(1,0,0,0)^T$ and K is a correction gain. Local convergence is ensured by the following. Let us define $\xi := \widehat{q}^* q_V - 1$ the attitude error, and $\zeta := q\xi q^*$ the attitude error expressed in the inertial frame $\mathcal{R}_W$. The dynamics of this latter is given by

$$\begin{aligned} \partial_t \zeta &= (\partial_t q)\xi q^* + q(\partial_t \xi)q^* + q\xi(\partial_t q^*) \\ &= q(\boldsymbol{\omega_m} \times \xi)q^* + q(\partial_t \xi)q^* \end{aligned} \tag{9.3}$$

On the other hand, assuming that the attitude measurements q_V have same dynamics that the true attitude q ($\partial_t q_V = \frac{1}{2} q_V \boldsymbol{\omega_m}$), the dynamics of the error ξ is

$$\begin{aligned}\partial_t \xi &= -\widehat{q}^*(\partial_t \widehat{q})\widehat{q}^* q_V + \widehat{q}^* \partial_t q_V \\ &= -\tfrac{1}{2}(\boldsymbol{\omega_m} + K\xi)(\xi + 1) + \tfrac{1}{2}(\xi + 1)\boldsymbol{\omega_m} \\ &= \xi \times \boldsymbol{\omega_m} - \tfrac{1}{2} K\xi(\xi + 1)\end{aligned}$$

which in linearized around equilibrium ($\xi = 0$) by

$$\partial_t \xi = \xi \times \boldsymbol{\omega_m} - \frac{1}{2} K\xi \tag{9.4}$$

Plugging (9.4) into (9.3) yields

$$\partial_t \zeta = -\frac{1}{2} K\zeta.$$

Thus, a positive correction gain K ensures local exponential convergence of ζ towards zero. This means that our attitude estimate stays close to the visual attitude measurements when such measurements are available, is continuous and differentiable by construction, and is always available. This characteristics satisfy the conditions required for realistic augmented reality.

9.6.2 Position estimation

For position estimation, magneto-inertial and RGBD data are combined: first, to estimate the linear velocity $\widehat{\boldsymbol{v}}$, and then to estimate the position $\widehat{X_C}$ itself. The proposed observer for velocity and position is the following:

$$\begin{cases} \partial_t \widehat{\boldsymbol{v}} = \widehat{\boldsymbol{v}} \times \boldsymbol{\omega_m} + \boldsymbol{a_m} - \widehat{q}^* \mathbf{A} \widehat{q} + k_v(\boldsymbol{v_m} - \widehat{\boldsymbol{v}}) + k_{vx} \widehat{q}^*(X_m - \widehat{X_C})\widehat{q} \\ \partial_t \widehat{X_C} = \widehat{q}\widehat{\boldsymbol{v}}\widehat{q}^* + k_X(X_m - \widehat{X_C}) \\ \widehat{\boldsymbol{v}}(0) = \boldsymbol{v_m}(0) \\ \widehat{X_C}(0) = X_m(0) \end{cases} \tag{9.5}$$

where

- $\boldsymbol{a_m}$ and $\boldsymbol{\omega_m}$ are the measurements from accelerometers and gyrometers expressed in camera frame $\mathcal{R}_{cam}$,
- $\widehat{q}$ is the output of the previous attitude estimation (see Subsection 9.6.1),
- $\boldsymbol{v_m}$ is the output of the linear velocity estimation algorithm from depth data expressed in camera frame $\mathcal{R}_{cam}$ (see Chapter 6),
- X_m is the position measurement with respect to the inertial frame $\mathcal{R}_W$ (the output of the tracking algorithm).

To establish convergence results, let us define

- $\mathbf{V_m} \coloneqq q \boldsymbol{v_m} q^*$ the linear velocity measurement expressed in $\mathcal{R}_W$,
- $\widehat{\mathbf{V}} \coloneqq q\widehat{\boldsymbol{v}}q^*$ the linear velocity estimate expressed in $\mathcal{R}_W$,
- $\widetilde{\mathbf{V}} \coloneqq \widehat{\mathbf{V}} - \mathbf{V}$ the error of linear velocity,
- $\widetilde{X} \coloneqq \widehat{X_C} - X$ the error of position,
- $\xi \coloneqq q\widehat{q}^* - 1$ the absolute error of attitude.

Then, as in the previous section, the dynamics of $\widehat{\mathbf{V}}$ is given by

$$\begin{aligned}\partial_t \widehat{\mathbf{V}} &= q(\boldsymbol{\omega_m} \times \widehat{\boldsymbol{v}})q^* + q(\partial_t \widehat{\boldsymbol{v}})q^* \\ &= q\boldsymbol{a_m}q^* - (\xi+1)\mathbf{A}(\xi^*+1) + k_v(\mathbf{V_m} - \widehat{\mathbf{V}}) + k_{vx}(\xi+1)(X_m - \widehat{X_C})(\xi^*+1).\end{aligned} \tag{9.6}$$

Assuming that attitude error ξ is small, $\xi^* = -\xi$ (ξ can be identified with a vector of $\mathbb{R}^3$). The terms of equation (9.6) involving ξ can be linearized, yielding

$$\partial_t \widehat{\mathbf{V}} = q\boldsymbol{a_m}q^* - \mathbf{A} + 2\mathbf{A} \times \xi + k_v(\mathbf{V_m} - \widehat{\mathbf{V}}) + k_{vx}(X_m - \widehat{X_C}) - 2k_{vx}(X_m - \widehat{X_C}) \times \xi \tag{9.7}$$

and similarly

$$\partial_t \widehat{X_C} = \widehat{\mathbf{V}} + 2\widehat{\mathbf{V}} \times \xi - k_X(X_m - \widehat{X_C}). \tag{9.8}$$

Finally, assuming that the measurements X_m and $\boldsymbol{v_m}$ can be identified with true position and velocity X and $\boldsymbol{v}$, this yields the following dynamics for the errors $\widetilde{\mathbf{V}}$ and $\widetilde{X}$:

$$\begin{cases} \partial_t \widetilde{X} = \widetilde{\mathbf{V}} - k_X \widetilde{X} + 2\widehat{\mathbf{V}} \times \xi \\ \partial_t \widetilde{\mathbf{V}} = -k_{vx}\widetilde{X} - k_v \widetilde{\mathbf{V}} + 2\mathbf{A} \times \xi + 2k_{vx}\widetilde{X} \times \xi. \end{cases} \tag{9.9}$$

This system can be decomposed in a linear globally exponentially stable system (for positive correction gains k_X, k_v and k_{vx}), and perturbation terms which directly depend on the attitude error ξ. Provided that the correction gains are large enough and the attitude error ξ is small enough, the perturbations terms should be dominated by the exponential convergence of the errors of position and linear velocity towards 0.

Chapter 10

Experimental results

Validation quantitative de l'algorithme

Les résultats de ce nouvel algorithme sont présentés dans ce chapitre : la qualité de la reconstruction de trajectoire est estimée par comparaison avec une trajectoire de référence acquise par un système d'acquisition de mouvement ; la qualité de la cartographie peut uniquement être évaluée par des mesures manuelles. Les résultats sont suffisamment bons pour permettre l'addition de données virtuelles à l'image réelle, et un petit jeu a été conçu pour démontrer les capacités du système.

The results of this new algorithm and system are presented in this chapter: the tracking quality is evaluated by comparison with a reference trajectory acquired by a motion capture system; the quality of the cartography can only be evaluated by manual measurements. The results are good enough to perform visual augmentation of a real image with a virtual object, and a small game has been designed to demonstrate the systems capabilities.

10.1 Experimental conditions

To test our system and the associated algorithm, experiments have been performed in a room equipped with a *motion capture system* (distributed by Vicon). This system, widely used in research as well as in the entertainment industry, consists of a set of cameras pointing towards the center of the room, equipped with infrared optical filters and infrared LEDs. The positions and orientations of these cameras have to be carefully calibrated before any experiment. Reflecting dots (markers) are embedded on the system whose trajectory is to be estimated: the set of markers forms a rigid object, tracked by the cameras (infrared filtering ensures that only these dots are detected). Provided that these dots are detected by enough cameras, the position and orientation of the object can be estimated, with a millimeter resolution. The actual resolution of reconstruction depends on the number of detecting cameras. See [Aghajan and Cavallaro, 2009] for more details on motion capture systems. Figure 10.1 shows our system equipped with the reflective dots: at the forefront of the system is the RGBD sensor (an Xtion Pro distributed

by ASUS), and behind are fixed the inertial sensors (accelerometers, magnetometers and gyrometers). Figure 10.2 shows the room equipped with cameras of the Vicon motion capture system.

Note that only position and orientations are provided: velocities are postprocessed in the following for results comparison.

Note also that the motion capture system provides the attitude and position of the rigid body formed by the reflective markers attached to the system, with respect to an inertial fixed reference frame: both differ from the the reference frame $\mathcal{R}_{cam}$ that defines our system and the world $\mathcal{R}_W$. As a result, thorough evaluation of all the different reference frames must be achieved in order to enable trajectory comparison. This work, which might be fascinating as well as time-consuming, is not presented in this dissertation. Imprecisions might persist, yielding slight repeatable trajectory differences.

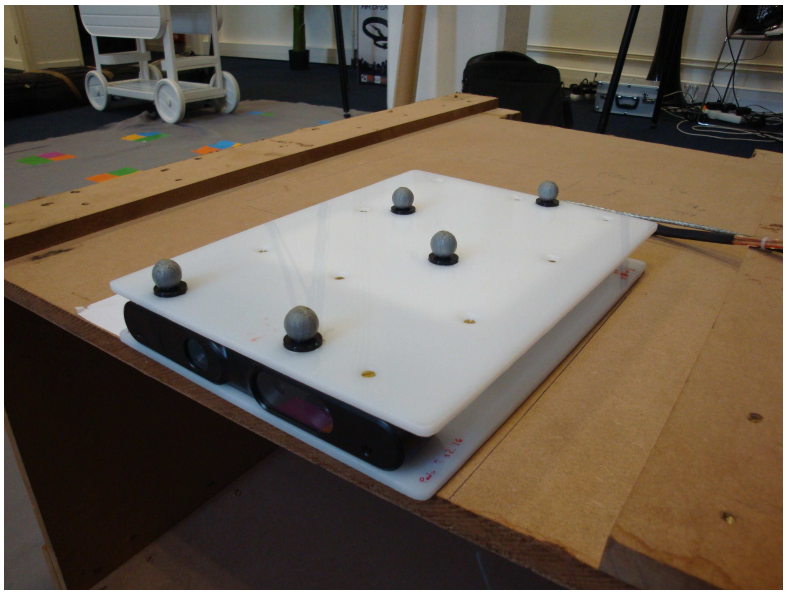


Figure 10.1: Our system equipped with reflective markers tracked by the Vicon system.

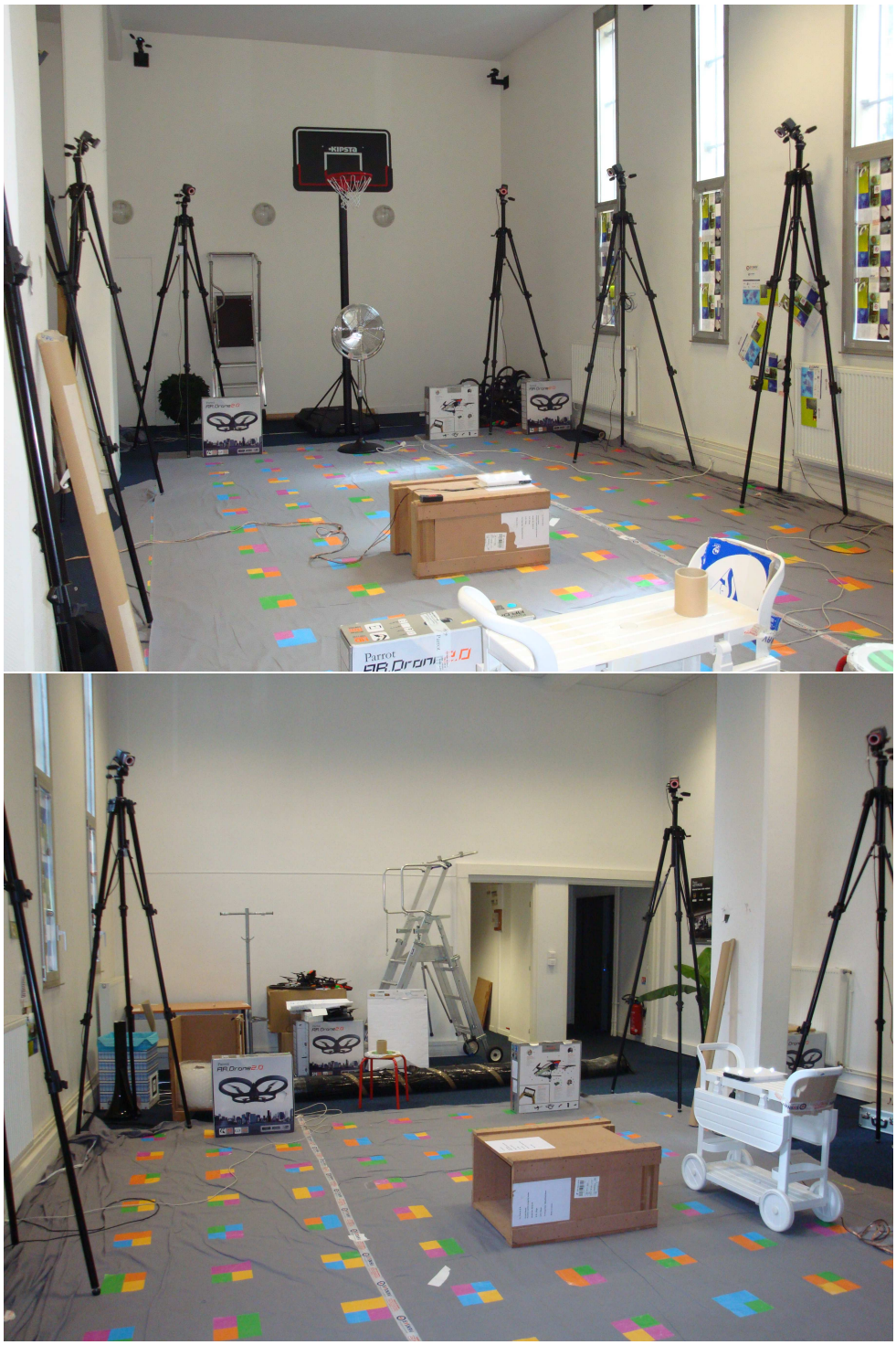


Figure 10.2: Views of the room equipped with a motion capture system (system Vicon©)

10.2 Linear velocity, position and attitude estimation

10.2.1 Translation along the lateral direction

In a first test, the system is placed from its starting point onto a trolley (shown in Figure 10.3), then the trolley moves back and forth along the lateral direction of the system (the x-axis): the system is only submitted to translations. Linear velocity estimation is the result of the fusion

Figure 10.3: The trolley used to ensure that our system is only submitted to pure lateral translation.

of linear velocity measurements (computed by geometrical flow method described in Chapter 6), filtered by accelerometers using inertial attitude estimation (see Subsection 9.6.1 for attitude estimation and system (9.5) for linear velocity filtering). An important part of the algorithm is the detection of inaccurate linear velocity measurements (caused by depth images of poor quality). Figures 10.4 and 10.5 show the results of linear velocity estimation, and more precisely the linear velocity measurements, detected as accurate (blue points) and inaccurate (red points). The filtered linear velocity is plotted in cyan and the reference (computed from motion capture positions) in black. A zoom on the linear velocity in the x-axis direction shows that most of the inaccurate linear velocity measurements are actually detected: for example between the times $\underline{t}$ = 160 sec and $\overline{t}$ = 240 sec, the high velocity phases are subject to inaccurate measurements: nevertheless, the accelerometers during these short periods are sufficient to provide open-loop integration, before the next accurate measurement provides the next velocity fix. The same phenomenon occurs between the times $\underline{t}$ = 260 sec and $\overline{t}$ = 330 sec.

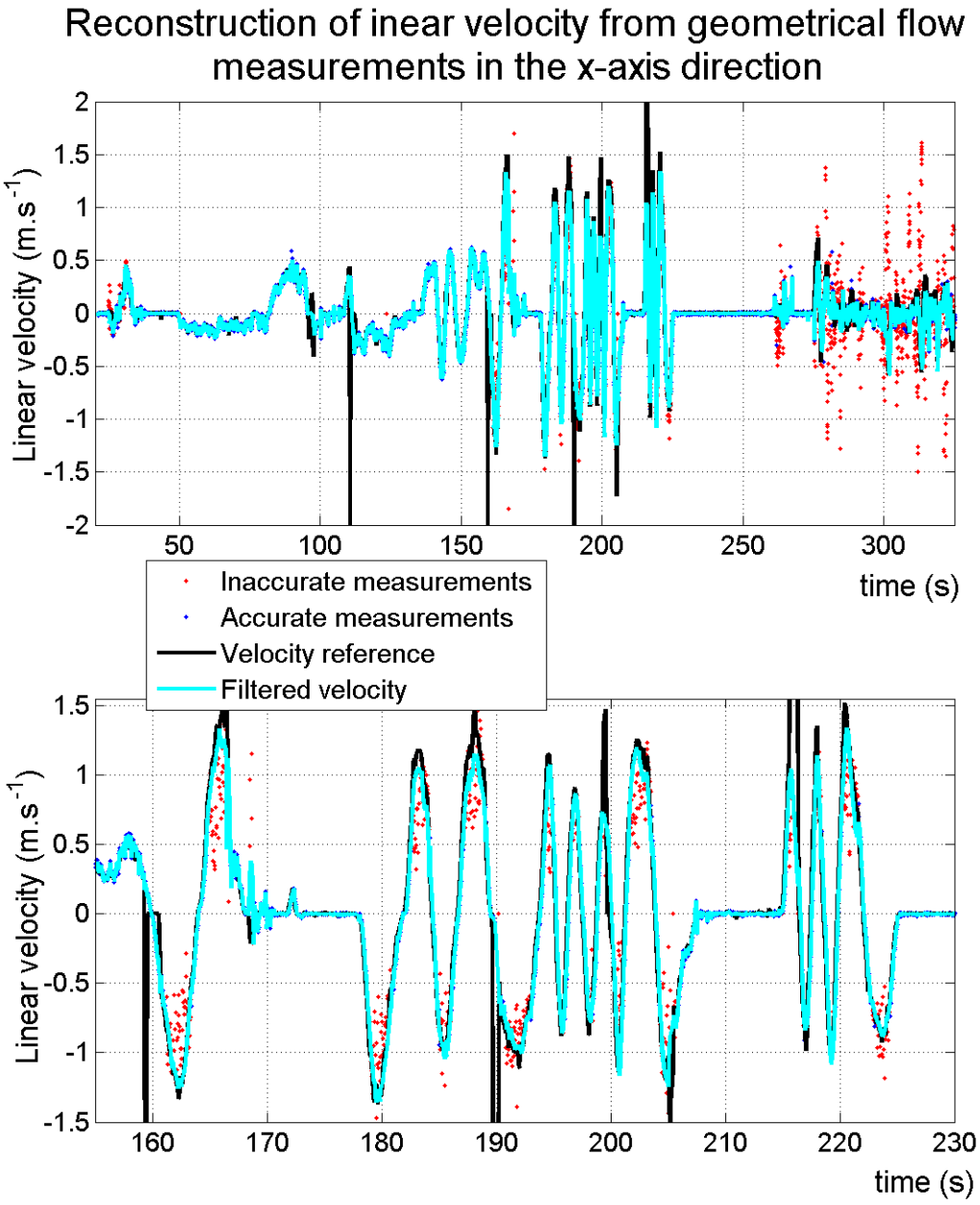


Figure 10.4: Linear velocity measurements are evaluated as accurate or inaccurate, depending on which they are used or not as velocity fixes in the linear velocity filter (9.5). When measurements are unavailable (red), accelerometers are integrated in open-loop. Overview (top) and zoom (bottom) on linear velocity along the lateral direction. Comparison with the reference trajectory (black) shows an excellent accuracy.

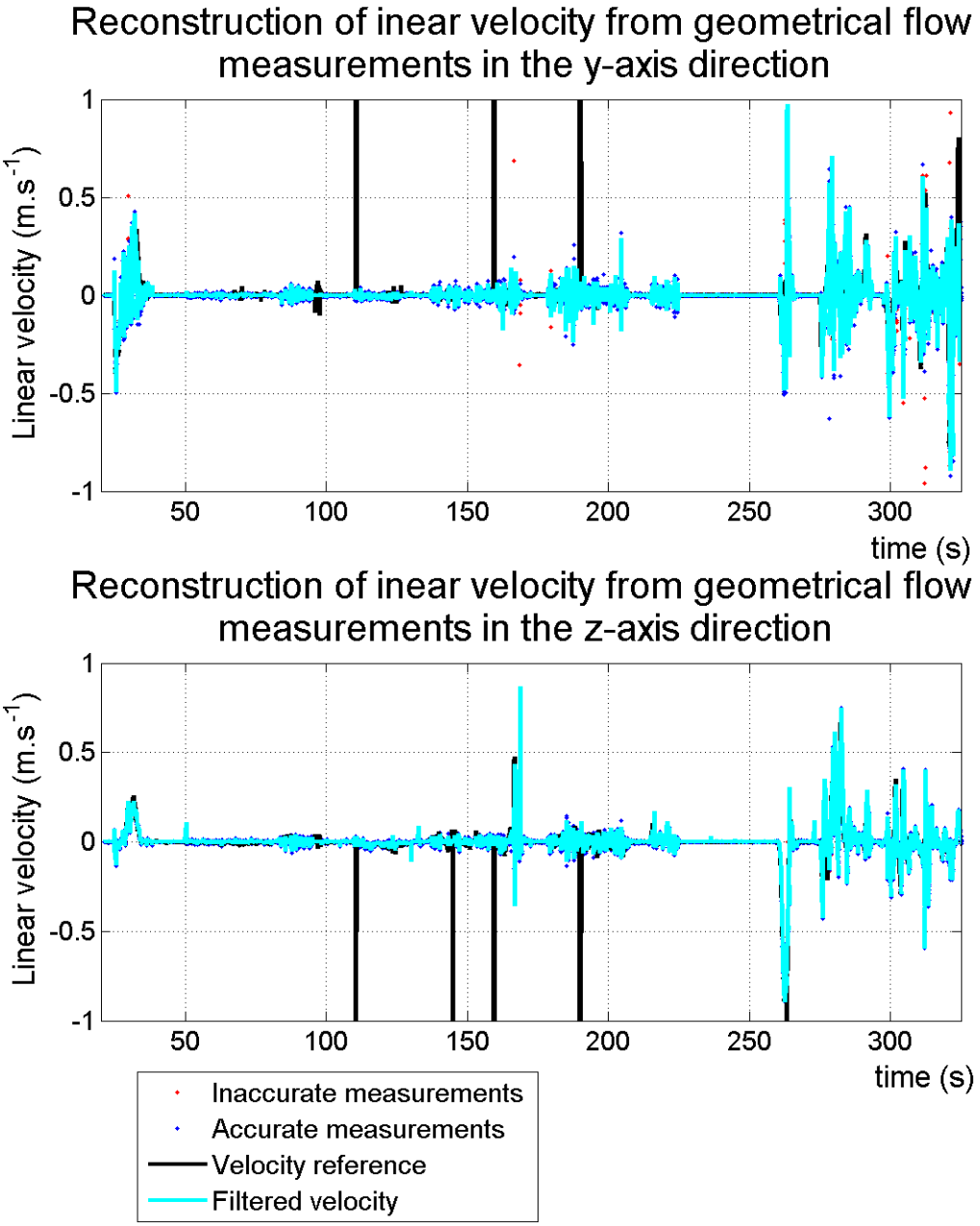


Figure 10.5: Linear velocity measurements are evaluated as accurate or inaccurate, depending on which they are used or not as velocity fixes in the linear velocity filter (9.5). When measurements are unavailable, accelerometers are integrated in open-loop. Linear velocity along the vertical direction (top) and optical axis (bottom) directions.

Filtered linear velocity is also compared to the velocity that was used in the original algorithm, computed by a motion model described in Section 8.3. Recall that this velocity is used to predict the current pose of the camera, which enables the computation of the PVS. The velocity is plotted (in blue) along with filtered velocity (in red) and velocity reference (in black) computed from motion capture positions in Figure 10.6. These results shows that the velocity of the motion model is noisier and less accurate than the filtered velocity, which should lead to better tracking results. Unfortunately, we were unable to prove that the impact of this improved prediction step actually improved the tracking accuracy.

Finally, the position is filtered as described by (9.5), using attitude estimation, filtered linear velocity (expressed in $\mathcal{R}_{cam}$) and position measurements (the output of the tracking algorithm). The result is shown in Figure 10.7. Recall that the position is expressed in the world reference frame $\mathcal{R}_W$, which explains that the larger variation of position occurs along the z-axis. These results show that the strong position outliers are canceled: the filter provides a continuous position even when the tracking algorithm based on RGB data processing is lost, which occurs in this test only once, at $\bar{t} \approx 170$ sec. The position estimation in the x-axis direction does not seem accurate in this test: at some times, the error exceeds 10 cm. However, it is repeatable. This error might actually be the artifact of an imprecision in the harmonization of the different reference frames, as explained in Subsection 10.1.

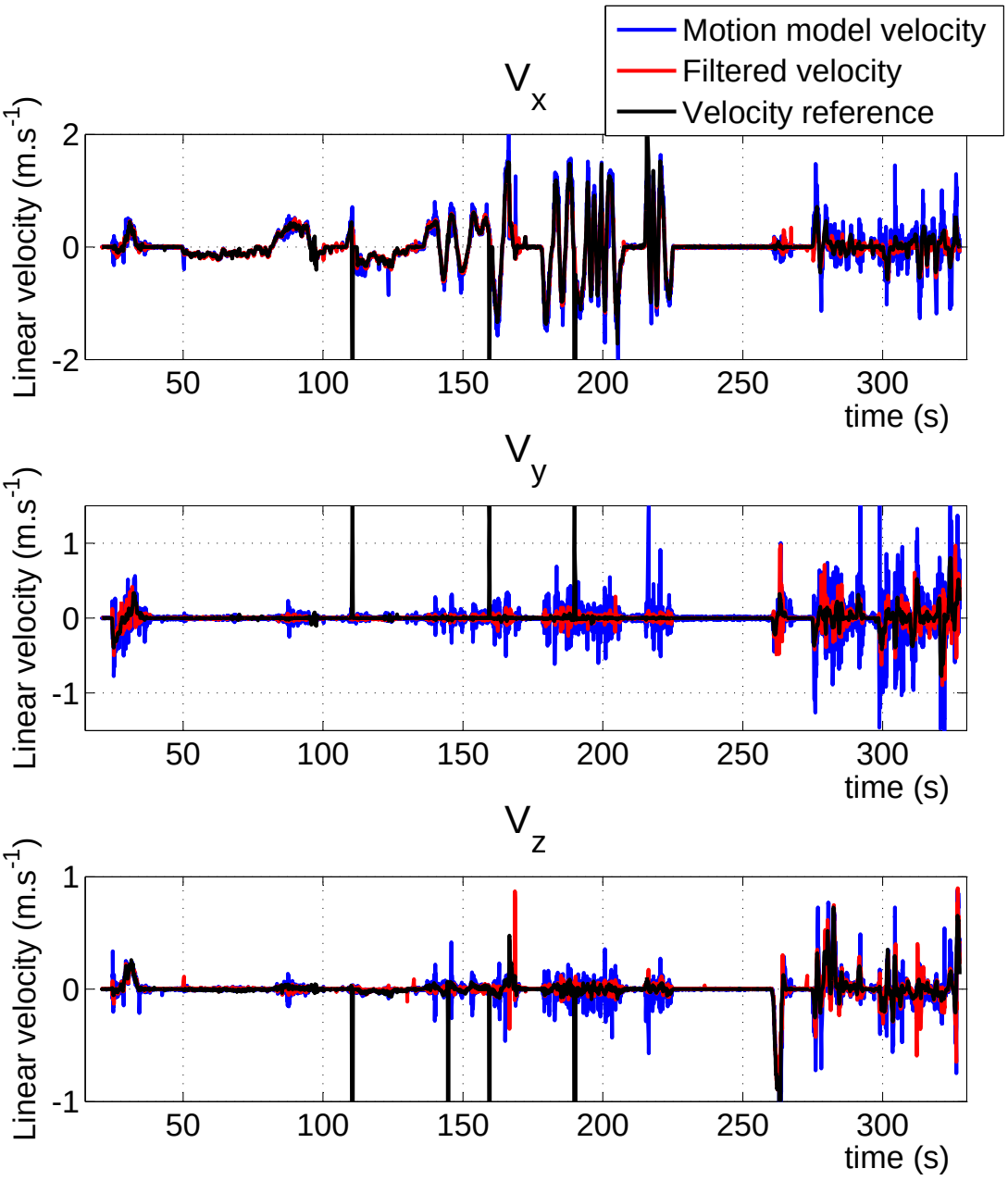


Figure 10.6: Comparison of the filtered linear velocity and the motion model velocity used in the prediction step of the tracking algorithm.

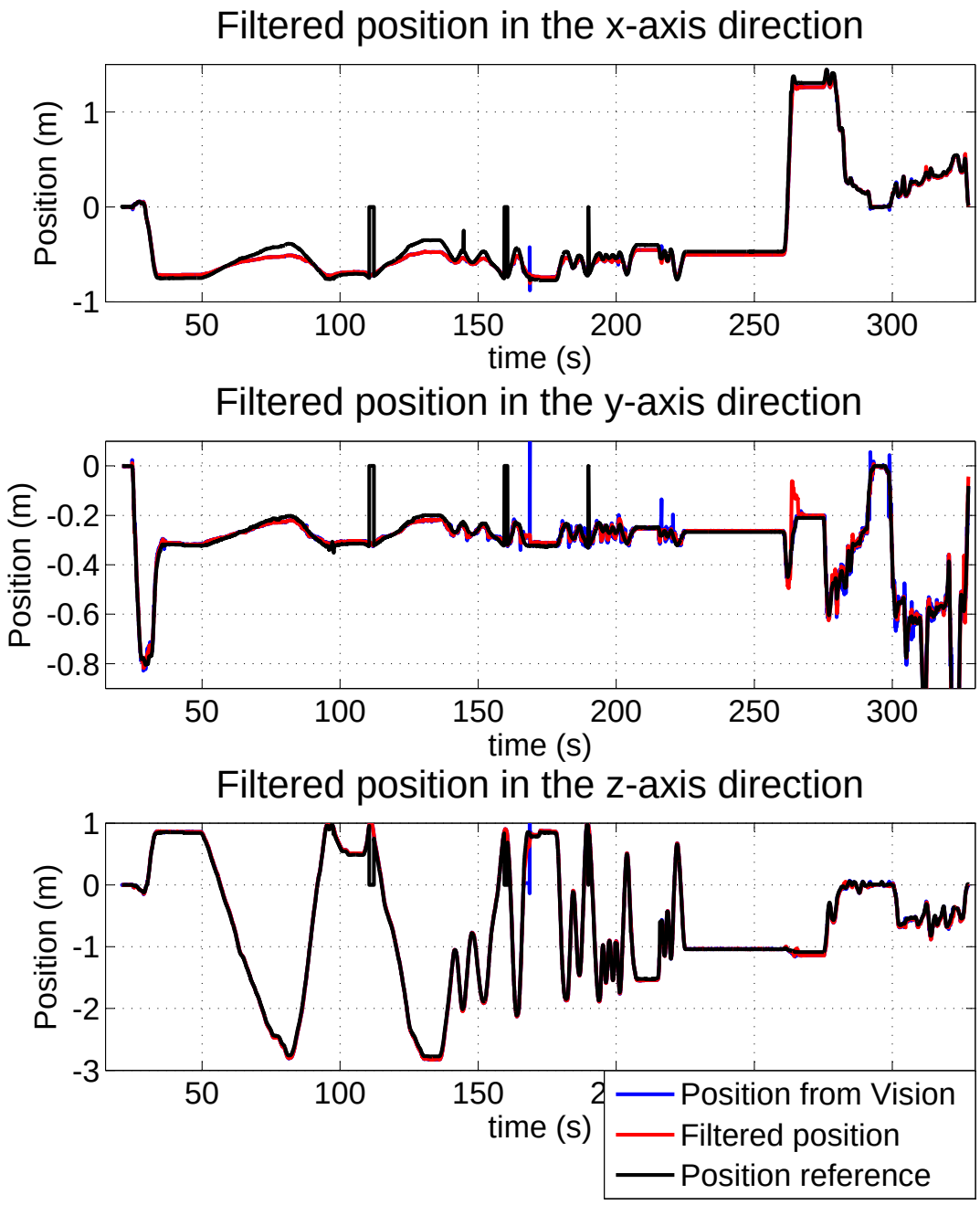


Figure 10.7: Comparison of the filtered camera position in the world reference frame $\mathcal{R}_W$ (in red) with the output of the tracking algorithm (in blue) and the position reference (in black).

10.2.2 Exploration of an entire room

In a second test, the system is handheld and the goal is to explore the entire room, with different dynamics: first, a slow loop is performed around the room, then motion with higher dynamics occur to test the robustness of the algorithm. The motion mostly occurs in a horizontal plane.

Just as for the previous test, linear velocities are estimated, and plotted along with the motion model and the reference linear velocities in Figure 10.8. Again, the filtered velocity is cleaner, except at very specific times where inaccurate velocity measurements were not successfully detected (for example at times $\bar{t} \approx 130$ sec or $\bar{t} \approx 190$ sec in the x-axis direction).

Then the attitude estimation is performed and results are plotted in Figure 10.9: recall that attitude estimation as described in Subsection 9.6.1 is based on gyrometers measurements and attitude measurements (outputs of the tracking algorithm). Along with these is plotted the attitude estimate based on inertial data only as described in Section 9.4. All of them are represented as Euler angles (see the Subsection C.1.3 in appendix for definition of these angles). The goal of this filter is completely achieved as it filters the high frequency dynamics of the tracking output as well as provides continuous attitude estimate when this measurement is not available: for example, at times $\bar{t} \approx 100$ sec or $\bar{t} \approx 130$.

Finally, position estimation is performed as described by (9.5) and explained for the previous test. The result is plotted in Figure 10.10, and a top view of the trajectory is shown in Figure 10.11. These results show that position estimation is generally accurately achieved, except at certain times, which are the combination of the loss of tracking and inaccurate and undetected linear velocity measurements.

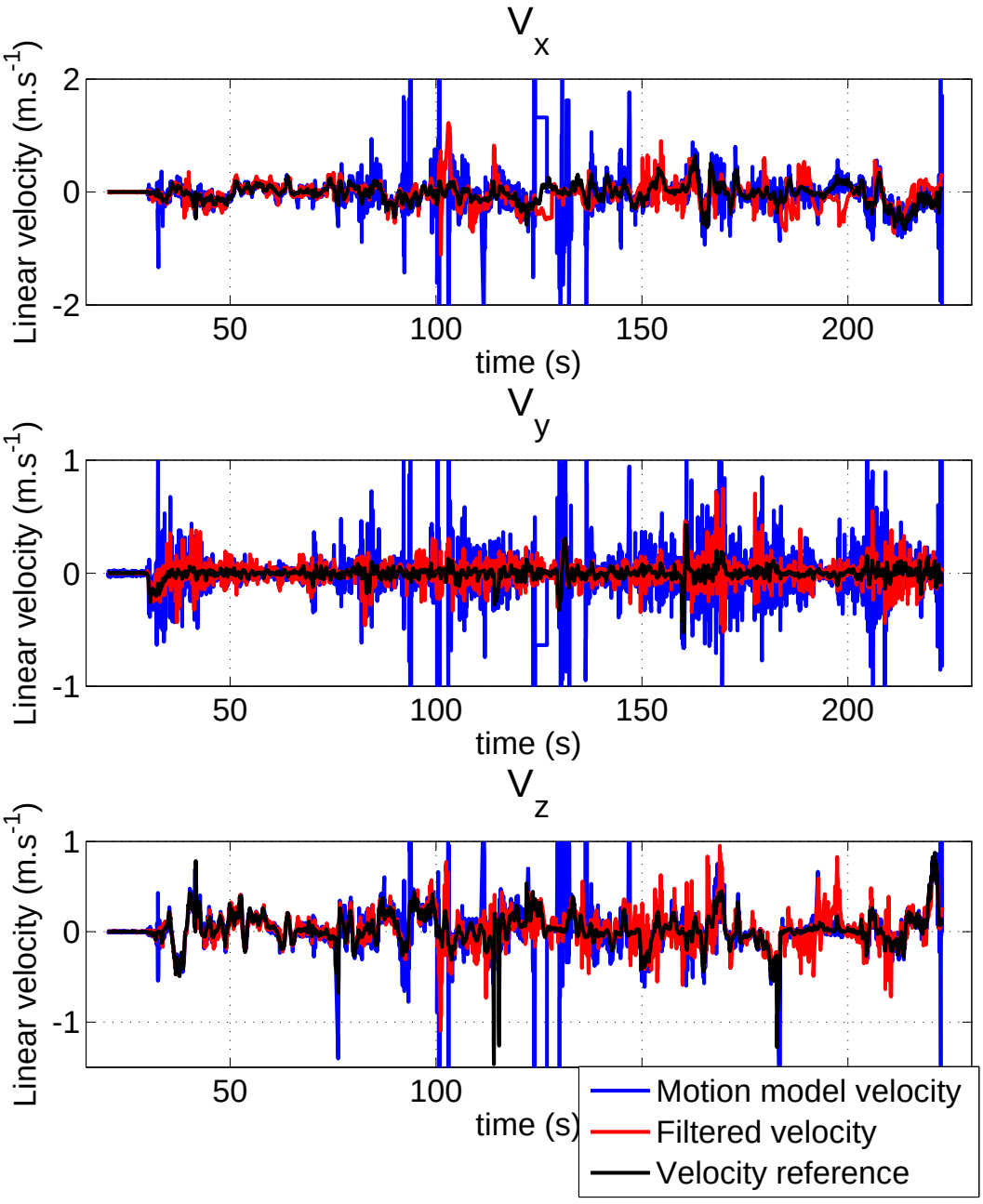


Figure 10.8: Comparison of the filtered linear velocity and the motion model velocity used in the prediction step of the tracking algorithm.

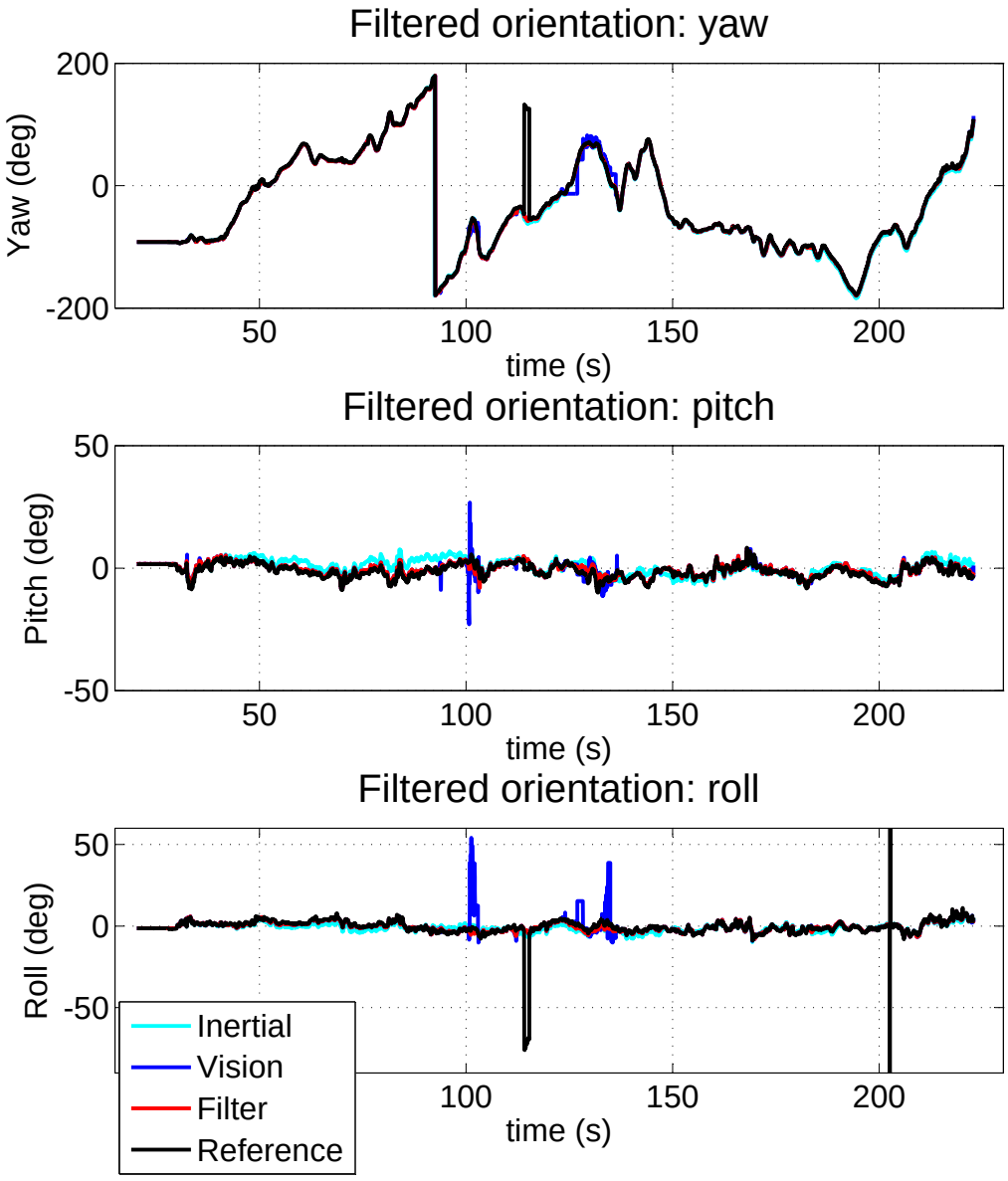


Figure 10.9: Attitude estimation, represented by Euler angles: in cyan, using inertial data only (see Section 9.4); in blue, the output of the tracking algorithm; in red, the filtered attitude (see Subsection 9.6.1); in black the reference.

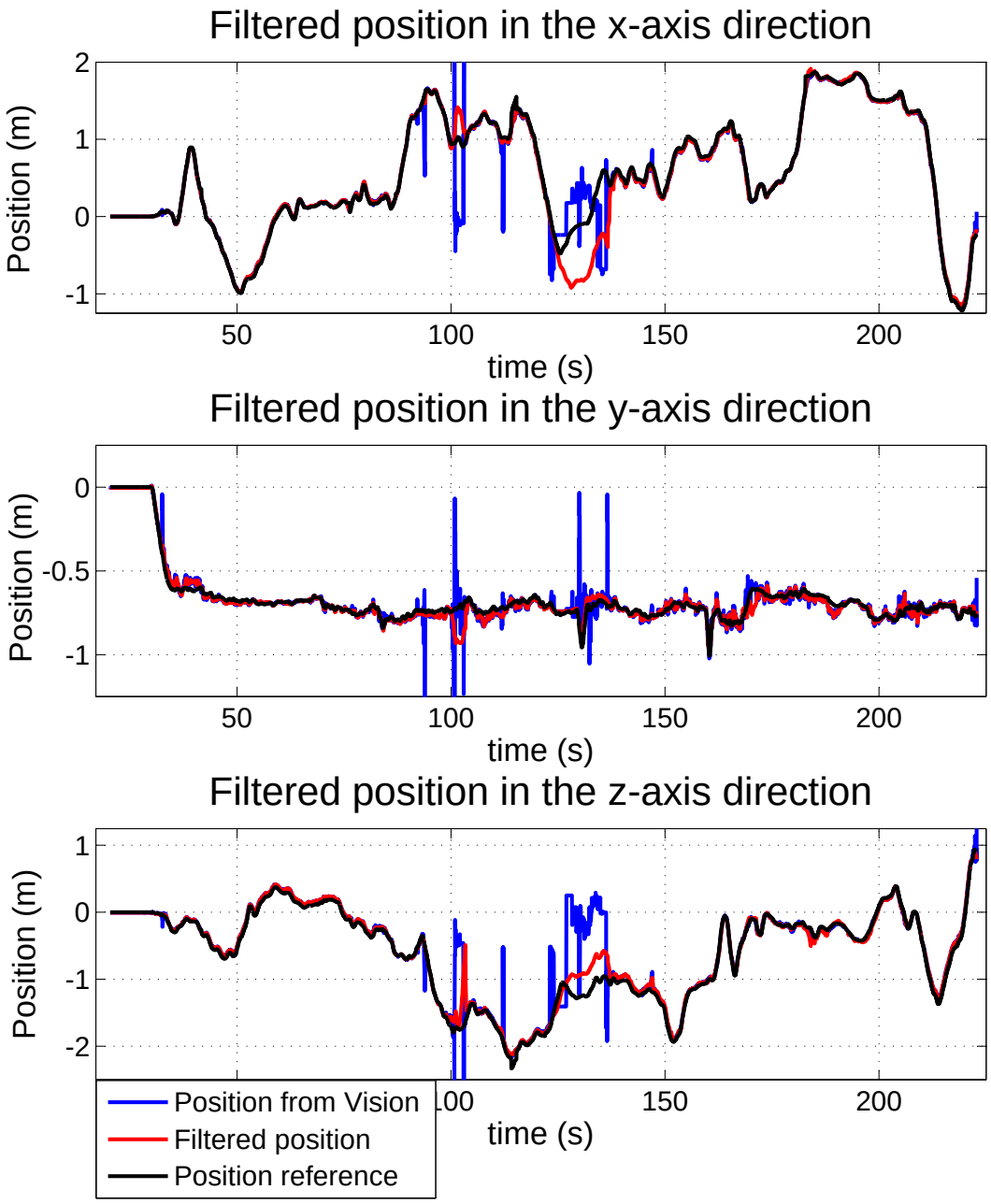


Figure 10.10: Comparison of the filtered camera position in the world reference frame $\mathcal{R}_W$ (in red) with the output of the tracking algorithm (in blue) and the position reference (in black).

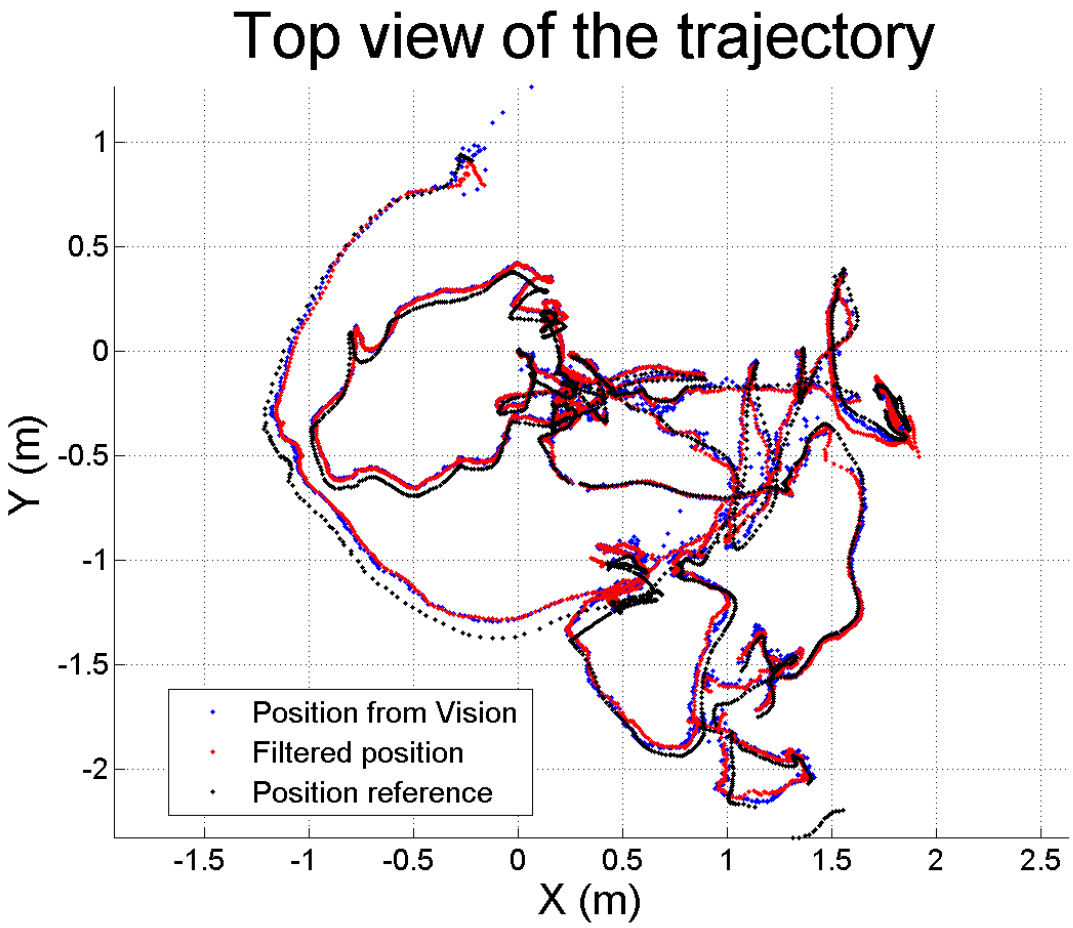


Figure 10.11: Top view of the trajectory of the camera during entire room exploration.

10.3 Cartography

A cartography can be reconstructed as a point cloud: from each depth image, a new point cloud can be generated, whose points positions in the world reference frame $\mathcal{R}_W$ are computed using current filtered camera position and attitude. To each of these points, a color is attributed using the RGB image, and the different point clouds are then aggregated to form the cartography. The result of such a reconstruction is shown in Figures 10.12 and 10.13. In the first figures, different views of the room are shown, along with the corresponding views of the point cloud formed by the map points of the Vision algorithm. In the second figure, an overview of the entire reconstruction is shown. Notice that the points corresponding to large depth measurements (e.g the distant walls of the room) form noisier point clouds: indeed, the noise distribution of the depth image directly depends on the depth value itself.

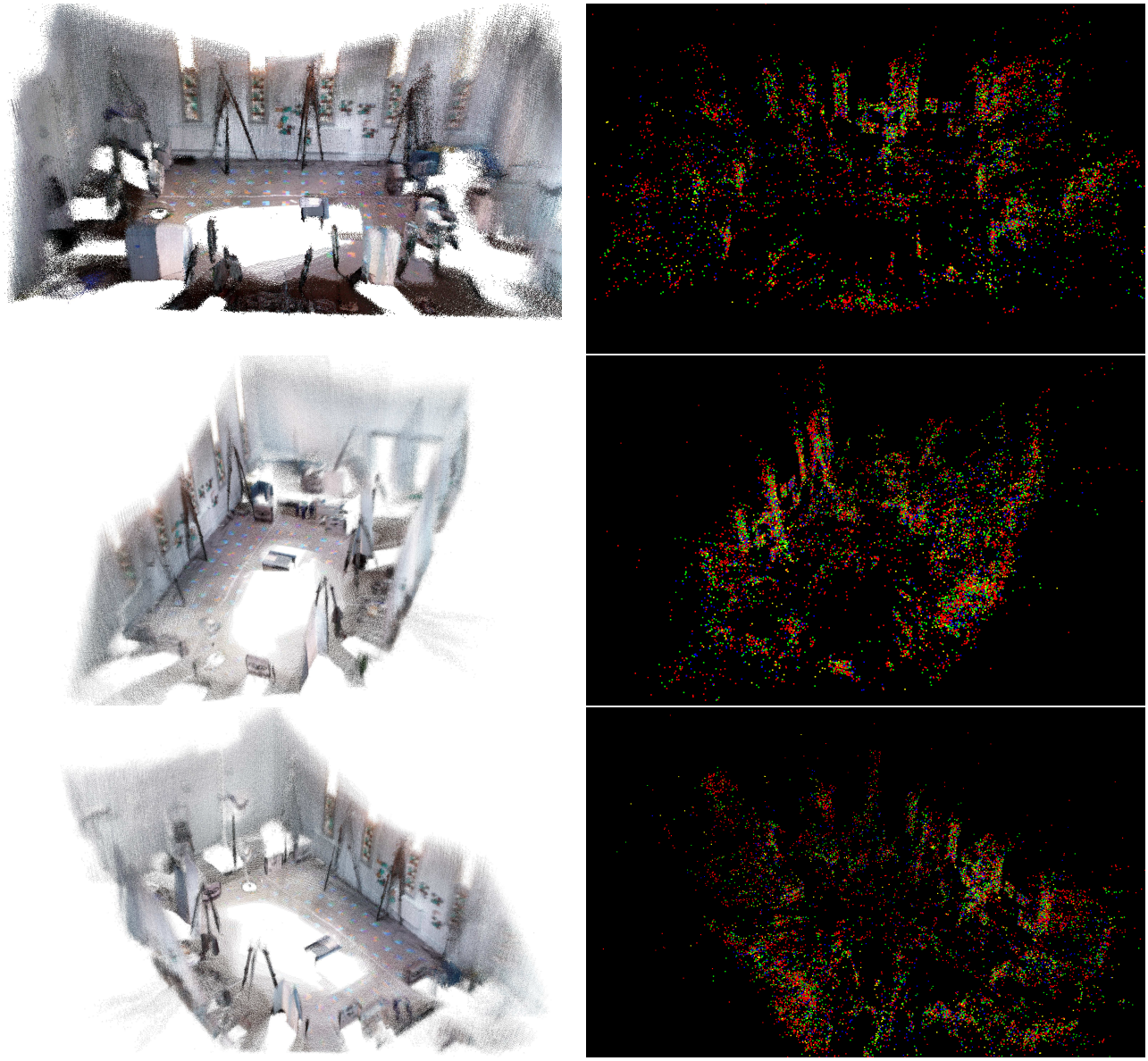


Figure 10.12: Details of the reconstructed cartography, computed using filtered position and depth measurements (left) and map points of the Vision algorithm (right).

Figure 10.13: Entire view of the reconstructed cartography, computed using filtered position and depth measurements. The reconstructed point cloud is very accurate, since relative errors on direct physical measurements do not exceed 1%. The overall geometry of the reconstructed scene is also very convincing: walls are flat and objects are not duplicated.

To assess the accuracy of the reconstructed cartography, distances were measured in the room using a laser telemeter. These various lengths, corresponding to physical distances such as space between windows or distance from a wall to another, are shown in Figure 10.14. The

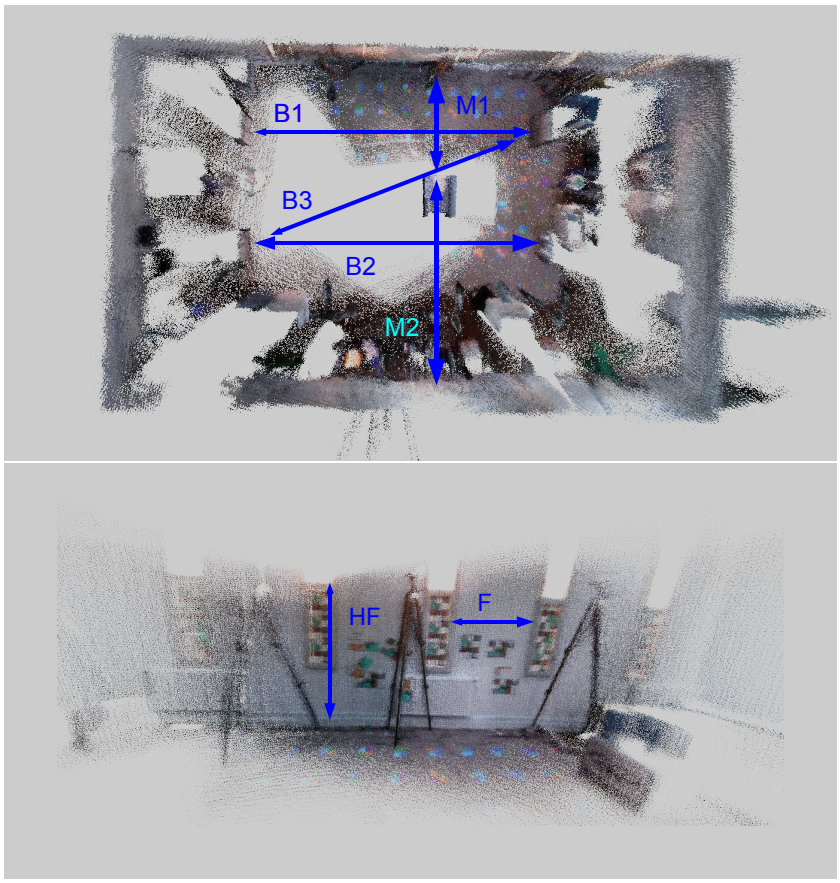


Figure 10.14: Definition of the meter-readings measured in the room by a laser telemeter.

same physical lengths are then estimated in the point cloud and compared to their actual values: comparison of both is shown in Table 10.1. Due to the noise in certain parts of the point clouds, especially the walls, the measurements can only be given with an uncertainty margin, which depends on the physical object being measured. The reconstructed point cloud is very accurate, since relative errors on the measurements do not exceed 1%. The overall geometry of the reconstructed scene is also very convincing: walls are flat and objects are not duplicated.

	Real (m)	Cartography (m)
B_1	5.92	5.94 ± 0.02
B_2	5.91	5.90 ± 0.01
B_3	6.45	6.42 ± 0.10
M_1	2.41	2.39 ± 0.05
M_2	4.21	4.12 ± 0.15
F	1.31	1.28 ± 0.03
HF	2.44	2.39 ± 0.06

Table 10.1: Comparison of some specific physical distances measured in reality and in the point cloud of the cartography.

Conclusion

In this thesis, we have presented different algorithms that meet the needs of visual augmented reality for a dense 3D perception of the surrounding environment on one hand and for a precise localization of the camera in this environment on the other hand.

The theoretical framework is based on a spherical camera model, which allows to express the dynamics of the light intensity and the depth by a system of $SO(3)$-invariant PDEs. The camera dynamics are made explicit in this system, as well as the depth of the environment surrounding the camera. It appears that linear velocity and depth are indistinguishable in the first PDE (the optical flow constraint), as expected for a monocular system; but the second PDE (the geometrical flow constraint) is not submitted to this ambiguity.

In the first part of this manuscript, the assumption of known camera dynamics is made to estimate the current dense depth field. Like for optical flow estimation, variational methods can be implemented to estimate the depth as the solution of the optical flow constraint where motion dynamics is made explicit. This formulation is advantageous because the fundamentally ill-posed problem of optical flow is now geometrically constrained by the knowledge of the camera dynamics. Accuracy of the result is highly dependent of the quality of the processed RGB images and of the value of the linear velocity. The indistinguishability of depth and velocity translates here as unobservability of the depth for a purely rotational motion. The geometrical flow constraint is thus exploited to design a non-linear asymptotic observer that refines the depth estimation based on optical flow or depth measurements.

The release of the Kinect device in November 2010 provided a low-cost and easy solution to the problem of dense depth estimation. In addition, it enabled to completely pose the converse problem, namely motion estimation from RGBD data as dense scalar fields. In a first time, the possibility to embed linear velocity sensors such as Pitot tubes or EM log sensors on the system as well as MEMS gyrometers led to the problem of bias estimation, unavoidable as soon as relatively low-cost sensors are involved. Interestingly, the observability of this question is strongly related to geometric properties of the environment, but not to the motion of the camera itself. Moreover, while these properties are quite intuitive, their meticulous demonstration requires a deep understanding of the Riemannian geometry of the unit sphere. The combined processing of regular images and depth images increase the probability that observability conditions may be satisfied for a given environment, but only depth images are necessary for the convergence of the estimation method. In a second time, a novel estimation method of linear and angular velocity by depth images processing was proposed. This technique is fast and does not lean on

features correspondence nor on explicit computation of spatial derivatives of the dense depth field. Exactly like for the previous observability problem of biases estimation, the quality of the results depend on properties of the environment. In addition, a confidence index is assigned to any measurement, which is a key component of good data fusion. These measurements are finally used to estimate biases on accelerometers and gyrometers in a third part, as well as to filter the linear velocity.

A practical implementation of a SLAM solution was tested for augmented reality: visual position measurements are issued from an adaptation of PTAM, where the missing scale factor is provided by the depth component of the processed RGBD data. Velocity measurements are obtained from our novel geometrical flow estimation method, and combined with these visual position measurements and inertial data to reconstruct a continuous and smooth trajectory that meet the augmented reality requirements. Cartography of the environment is still subject to long-time drifts or distortions when loop closure occurs, which is one of the biggest limitations of the actual system. This might be fixed by the addition of better features for detection and description: for example, the ORB detector and descriptor [Rublee et al., 2011] is comparable to SIFT in terms of accuracy, but runs about 100 times faster. The pose prediction of the tracking module is highly improved by our trajectory estimation, which means that the following pose estimation should not suffer from the reduce in the number of "low-quality" features and the addition of "good-quality" ones. In addition, better descriptors could be used in the relocalization module instead of the actual FSBI, and could be of great help for loop-closure detection [Angeli et al., 2008]. After loop-closure, the map should be globally corrected, by a distribution of error in the loop either uniformly or weighted by the path length, or by more sophisticated relaxation methods [Sprickerhof et al., 2011]. If the map can be efficiently corrected by loop-closure detection, the algorithm could benefit of two advantages. First, in a situation where pose estimation is submitted to large drifts, the local nature of pose optimization in the tracking module of PTAM may lead to the *duplication* of the environment in the cartography, and consequently to oscillations between two positions in the camera pose estimation, one for each representative of the cartography. On the contrary, if loop-closure is detected, this harmful effect would be avoided. Secondly, drifts of the cartography yield drifts in the visual attitude measurements: even if the cartography is initially aligned with the inertial reference frame, the visual measurements used in the attitude observer (9.2) eventually lead to erroneous attitude estimation. This has a direct consequence in the position observer (9.5), where the gravity acceleration is expressed in the camera frame thanks to this attitude estimate, and makes impossible the sole use of accelerometers when neither velocity nor position measurements are available.

To our knowledge, the work the most similar to our system in terms of results was achieved in the Kintinuous project, which is an extension of the Kinect Fusion work. Descriptions and videos are available online ([Whelan et al., 2012, Whelan et al., 2013a, Whelan et al., 2013b]). The team first dealt with the possibility to work on large areas, then improved the odometry part (originally only based on ICP, now augmented by the work of [Steinbrucker et al., 2011]), and finally (at the time of writing) added a place recognition module for loop-closure detection and post-correction of the map. This system, and especially the mapping module, almost entirely

runs on the Graphical Processing Unit (GPU) of the computer, which enables Massively Parallel Computation. The successive modifications applied to the Kintinuous project confirm our intuition that loop-closure detection is necessary, and suggest that significant improvements in real-time performances and increase of the map size should be inquired by parallel implementation on GPU of our algorithms.

Appendix

Appendix A

Differential geometry on the Riemannian unit sphere

To deal with vector fields, gradients and algebraic computations on the unit sphere $\mathbb{S}^2$, two different approaches can be adopted:

- the sphere $\mathbb{S}^2$ can be thought of as a subset of $\mathbb{R}^3$, where any point or vector has regular coordinates (e.g. Cartesian or spherical), and special constraints must be taken into account to make sure that vectors and points lie on that surface. For example, a point X has Cartesian coordinates $(x, y, z)^T$ where $x^2 + y^2 + z^2 = 1$.
- the sphere $\mathbb{S}^2$ can be thought of as a two-dimensional manifold, where vectors and metrics are defined with respect to *curves* living on $\mathbb{S}^2$.

Although the first approach is the most natural to begin with, computations can easily become tedious. The second approach, adopted in this thesis, requires first to understand the basics of Riemannian metrics and the differential geometry associated. In the following, we first recall these basics in a simplified way (more details can be found in [Stoker, 1969] or [Doubrovine et al., 1987]), then some useful relations of differential geometry are established.

A.1 Riemannian manifolds

A n-dimensional r-differentiable manifold $\mathcal{M}$ can be thought as a topological space which is locally equivalent to the Euclidean vector space $\mathbb{R}^n$. A precise definition involves a countable collection of *patches* covering $\mathcal{M}$. Any of these patches can be mapped into the unit ball of $\mathbb{R}^n$. Compositions of these mappings are consistent with respect to each other. The mappings are differentiable of order r with respect to the Euclidean coordinates of $\mathbb{R}^n$. Any composition of the mappings has non-vanishing Jacobian. Finally, $\mathcal{M}$ is *connected*.

A Riemannian manifold is a manifold for which a Riemannian metrics is defined and enables to define the concepts of lengths and vectors, necessary to work on intrinsic differential geometry. First, let us define a differentiable curve: it is the locus of points $p(t)$ in the manifold that results when the local coordinates u_i of the points of the locus are differentiable functions $u_i(t)$ of a

parameter t. To define the length of such a curve, let us a define the length of a line element dl, independent (by assumption) of the choice of local coordinates:

$$dl = \sum_{i,k} g_{ik} du_i du_k$$

where the functions g_{ik} are defined with respect to the coordinates, are differentiable and such that the quadratic form in the differentials is positive definite. To make sure that such a definition does not depend on the choice of local coordinates, it is required that the functions g'_{ik} associated to new coordinates u'_i must satisfy

$$g'_{ik} = \sum_{l,m} g_{lm} \frac{\partial u_l}{\partial u'_i} \frac{\partial u_m}{\partial u'_k}.$$

Then, the length of of a differentiable curve associated to $u_i(t)$ is defined by

$$l_{t_0}^{t_1} = \int_{t_0}^{t_1} \left(\sum_{i,k} g_{ik} \frac{du_i}{dt} \frac{du_k}{dt} \right)^{1/2} dt$$

and, more generally, a metric is defined using only the coefficients g_{ik}, so that the geometrical metric properties are invariant under change of local coordinates.

Then, vectors in the manifold $\mathcal{M}$ are identified with the tangent vectors of differentiable curves. For example, let us consider a curve $u_i(t)$ through a point p that corresponds to $t = 0$; its tangent vector at p can be written in the form $\frac{du_i}{dt}\big|_{t=0}$. Every differentiable curve through p defines such a vector $\mathbf{v}$. Conversely, every vector $\mathbf{v}$ is the tangent vector of a curve. These vectors form a linear vector space, called the tangent space of the manifold $\mathcal{M}$ at the point p.

These general matters are used in this thesis under the following form:

- a point on the unit sphere $\mathbb{S}^2$ is associated to a unit vector $\boldsymbol{\eta}$;
- considering a mapping $\mathbb{S}^2 \ni \boldsymbol{\eta} \mapsto h(\boldsymbol{\eta}) \in \mathbb{R}$, the gradient of h at point $\boldsymbol{\eta}$, denoted ∇h, lives in the tangent space at $\boldsymbol{\eta}$ denoted T_η;
- any differential geometry manipulation on h only has meaning in the tangent space T_η.

A.2 Differential calculus

In this section, we will establish some useful relations. Unless otherwise specified, $\mathbf{P}$ is a constant vector of $\mathbb{R}^3$. Let us define the mapping

$$\begin{array}{ccc} \mathbb{S}^2 & \to & \mathbb{R} \\ \boldsymbol{\eta} & \mapsto & f(\boldsymbol{\eta}) := \boldsymbol{\eta} \cdot \mathbf{P} \end{array}$$

We will first prove that

$$\nabla f = -\boldsymbol{\eta} \times (\boldsymbol{\eta} \times \mathbf{P}). \tag{A.1}$$

By definition of the differential of a mapping f defined on $\mathbb{S}^2$,

$$\nabla f \cdot \delta\boldsymbol{\eta} = df \quad \forall \delta\boldsymbol{\eta} \quad s.t. \quad \delta\boldsymbol{\eta} \cdot \boldsymbol{\eta} = 0. \tag{A.2}$$

The differential df in the direction $\delta\boldsymbol{\eta}$ writes $\mathbf{P}\cdot\delta\boldsymbol{\eta}$, which means that the projections of $\mathbf{P}$ and ∇f on T_η are the same.

$$\nabla(\boldsymbol{\eta}\cdot\mathbf{P})\cdot\delta\eta = \delta\eta\cdot\mathbf{P} \tag{A.3}$$

On the other hand, $\nabla f\cdot\boldsymbol{\eta}=0$ because ∇f at point $\boldsymbol{\eta}$ lives in the tangent space T_η, which means that ∇f equals its projection on T_η, and thus the projection of $\mathbf{P}$ on that space. This projection is written in the form $\mathbf{P}-(\boldsymbol{\eta}\cdot\mathbf{P})\boldsymbol{\eta}$, or equivalently $-\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\mathbf{P})$, which is the relation (A.1).

Now, we prove that

$$\Delta f = -2\eta\cdot\mathbf{P} \tag{A.4}$$

By definition, the Laplacian of a function is the divergence of its gradient. Thus

$$\Delta f = \nabla\cdot(-\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\mathbf{P})).$$

Let us define

$$\begin{array}{ccc}\mathbb{S}^2 & \to & T_\eta \\ \boldsymbol{\eta} & \mapsto & \mathbf{g}(\boldsymbol{\eta}) := -\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\mathbf{P}).\end{array}$$

The divergence of a vector field is the derivative of the area of a surface element as it evolves along the flow defined by the vector field. More precisely, as $\delta\boldsymbol{\eta_1}$ and $\delta\boldsymbol{\eta_2}$ are two infinitesimal vectors of T_η for $\boldsymbol{\eta}\in\mathbb{S}^2$, they define a surface element whose area is $(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta}$. As $\boldsymbol{\eta}$, $\delta\boldsymbol{\eta_1}$ and $\delta\boldsymbol{\eta_2}$ evolve according to

$$\frac{\partial\boldsymbol{\eta}}{\partial s} = \mathbf{g}(\boldsymbol{\eta}) \tag{A.5}$$

the divergence verifies:

$$\frac{\partial(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta}}{\partial s} = \nabla\cdot\mathbf{g}(\boldsymbol{\eta})(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta} \tag{A.6}$$

It is developped as

$$\frac{\partial(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta}}{\partial s} = (\frac{\partial\delta\boldsymbol{\eta_1}}{\partial s}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta} + (\delta\boldsymbol{\eta_1}\times\frac{\partial\delta\boldsymbol{\eta_2}}{\partial s})\cdot\boldsymbol{\eta} + (\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\frac{\partial\boldsymbol{\eta}}{\partial s}$$

Using (A.5) and the definition of $\mathbf{g}(\boldsymbol{\eta})$ applied to $\delta\boldsymbol{\eta_1}$, $\delta\boldsymbol{\eta_2}$ and $\boldsymbol{\eta}$:

$$\begin{aligned}\frac{\partial(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta}}{\partial s} =& ((-\delta\boldsymbol{\eta_1}\times(\boldsymbol{\eta}\times\mathbf{P})-\boldsymbol{\eta}\times(\delta\boldsymbol{\eta_1}\times\mathbf{P}))\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta} \\ &+(\delta\boldsymbol{\eta_1}\times(-\delta\boldsymbol{\eta_2}\times(\boldsymbol{\eta}\times\mathbf{P})-\boldsymbol{\eta}\times(\delta\boldsymbol{\eta_2}\times\mathbf{P})))\cdot\boldsymbol{\eta} \\ &+(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot(-\boldsymbol{\eta}\times(\boldsymbol{\eta}\times\mathbf{P}))\end{aligned}$$

Rearranging the terms, and since $\delta\boldsymbol{\eta_1}$ and $\delta\boldsymbol{\eta_2}\in T_\eta$ yields:

$$\frac{\partial(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta}}{\partial s} = -2(\boldsymbol{\eta}\cdot\mathbf{P})(\delta\boldsymbol{\eta_1}\times\delta\boldsymbol{\eta_2})\cdot\boldsymbol{\eta}$$

By identification with (A.6), this proves that

$$\nabla\cdot\mathbf{g}(\boldsymbol{\eta}) = -2(\boldsymbol{\eta}\cdot\mathbf{P})$$

which concludes the proof of the relation (A.4).

Finally, we prove that

$$\nabla \cdot (\boldsymbol{\eta} \times \mathbf{P}) = 0 \tag{A.7}$$

The method used to prove the relation (A.4) is applied to the vector field $f(\boldsymbol{\eta}) = \boldsymbol{\eta} \times \mathbf{P}$. This yields

$$\frac{\partial(\delta\boldsymbol{\eta_1} \times \delta\boldsymbol{\eta_2}) \cdot \boldsymbol{\eta}}{\partial s} = ((\delta\boldsymbol{\eta_1} \times \mathbf{P}) \times \delta\boldsymbol{\eta_2}) \cdot \boldsymbol{\eta} + (\delta\boldsymbol{\eta_1} \times (\delta\boldsymbol{\eta_2} \times \mathbf{P})) \cdot \boldsymbol{\eta} + (\delta\boldsymbol{\eta_1} \times \delta\boldsymbol{\eta_2}) \cdot (\boldsymbol{\eta} \times \mathbf{P}) . \tag{A.8}$$

Rearranging the terms yields the Jacobi identity:

$$\frac{\partial(\delta\boldsymbol{\eta_1} \times \delta\boldsymbol{\eta_2}) \cdot \boldsymbol{\eta}}{\partial s} = [(\delta\boldsymbol{\eta_1} \times (\delta\boldsymbol{\eta_2} \times \mathbf{P})) + (\delta\boldsymbol{\eta_2} \times (\mathbf{P} \times \delta\boldsymbol{\eta_1})) + (\mathbf{P} \times (\delta\boldsymbol{\eta_1} \times \delta\boldsymbol{\eta_2}))] \cdot \boldsymbol{\eta} = 0 \tag{A.9}$$

which by identification with (A.6) yields

$$\nabla \cdot f = 0.$$

Appendix B

Matrix Lie groups and Lie algebra

We will focus in this short description on matrix Lie groups on $\mathbb{R}$.

A matrix Lie group is any subgroup G of the group $GL(n,\mathbb{R})$ of invertible real matrices with the following property: if A_m is any sequence of matrices in G and A_m converges to some matrix A, then either $A \in G$, or A is not invertible. Thus a matrix Lie group is a closed subgroup of $GL(n,\mathbb{R})$. $GL(n,\mathbb{R})$ itself is a matrix Lie group. Other matrix Lie groups of interest will be studied in the following sections.

A Lie group is a differentiable manifold G which is also a group and such that the group product $G \times G \mapsto G$ and the inverse map $g \mapsto g^{-1}$ are differentiable. A matrix Lie group is a Lie group (as its name indicates, but not obviously from its definition).

Matrix Lie groups are in fact transformations that act on a vector space (e.g. $\mathbb{R}^n$), and composition of such transformations are simply matrix products.

The exponential of a square matrix X, denoted $\exp^X$ is defined as

$$\exp^X = \sum_{m=0}^{m=\infty} \frac{X^m}{m!}.$$

Such a definition is consistent since it is proven that the series converges. The matrix exponential is a continuous function of X.

The main properties of the matrix exponential are the following:

– $\exp^0 = I$

– $\exp^X$ is invertible and $(\exp^X)^{-1} = \exp^{-X}$

– if X and Y commute, $\exp^{X+Y} = \exp^X \exp^Y = \exp^Y \exp^X$.

The logarithm of a square matrix X, denoted $\log X$, is defined as

$$\log X = \sum_{m=1}^{m=\infty} (-1)^{m+1} \frac{(X-I)^m}{m}$$

whenever the series converge. The logarithm of a matrix X is defined and continuous on the set of square matrices with $\|X - I\| < 1$.

Let G be a matrix Lie group. The Lie algebra of G, denoted $\mathfrak{g}$, is the set of all matrices X such that $\exp^{tX}$ is in G for all real numbers t.

If G is a connected matrix Lie group, then every element A of G can be written in the form

$$A = \exp^{X_1} \exp^{X_2} \dots \exp^{X_m}$$

for some $X_1, X_2 \dots X_m$ in $\mathfrak{g}$.

Suppose G is a matrix Lie group with Lie algebra $\mathfrak{g}$. Then $\mathfrak{g}$ is the tangent space at the identity to G.

Any element X_I of the tangent space at the identity (and thus of $\mathfrak{g}$) can be written in the form

$$X_I = \sum_i \alpha_i \mathfrak{G}_i$$

where the $\mathfrak{G}_i$ are the *infinitesimal generators* of the Lie algebra and the α_i are real numbers. Consider an element A of G; a small variation around A acts as $\exp^{X_I}$: $A' = \exp^{X_I} A$ where A' is the new transformation. From the point of view of the vector space on which the Lie group G acts, consider two vectors $\mathbf{x}$ and $\mathbf{y}$ such that

$$\mathbf{y} = A\mathbf{x}.$$

$\mathbf{y}'$ is the result of a variation X_I at A acting on $\mathbf{y}$:

$$\mathbf{y}' = \exp^{X_I} A\mathbf{x} = \exp^{X_I} \mathbf{y}. \tag{B.1}$$

We can also formally write $\delta A := A' - A$. Then

$$\delta A = (\exp^{X_I} - I)A = \sum_{m=1}^{m=\infty} \frac{X_I^m}{m!} A. \tag{B.2}$$

Lie groups and Lie algebras have bee extensively studied. See for example [Hall, 2003] for a good introduction on these topics, and [Olver, 1999] for infinitesimal invariance theory applied to Lie groups.

Appendix C

3D rigid-body kinematics and quaternions

C.1 Attitude and angular velocity

The attitude (or orientation) of a rigid body can be represented in many ways: we will start with the most general one and give some other examples that are of great interest for specific applications.

C.1.1 The special orthogonal group

The orthogonal group $O(n)$ is the group of orthogonal matrices, that is, the column vectors that make up such a matrix are orthonormal. An equivalent definition is that $A \in O(n)$ if and only if $A^T A = I_n$ (or $A^{-1} = A^T$). Consequently, the determinant of orthogonal matrices is ± 1. The Special Orthogonal group $SO(n)$ is the subgroup of orthogonal matrices with determinant one. The elements of $SO(n)$ are the rotations of $\mathbb{R}^n$, and thus describe the orientation of a rigid body with respect to fixed external frame of reference. For example, every element of $SO(2)$ has the form

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad \text{with } \theta \in \mathbb{R}.$$

The Special Orthogonal group is a matrix Lie group. The Lie algebra of $SO(n)$ is the space of all matrices X of $M_n(\mathbb{R})$ such that $X^T = -X$: they are denoted $dt\Omega$ where Ω is written in the form

$$\Omega = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \text{ when } n = 3.$$

The infinitesimal generators at the identity (for $n = 3$) are

$$\mathfrak{G}_{\omega x} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathfrak{G}_{\omega y} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathfrak{G}_{\omega z} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{C.1}$$

Consider a vector expressed in a specific fixed coordinate frame $\mathcal{R}_W$ by $\mathbf{x}_w$, and in a moving coordinate frame $\mathcal{R}_c$ by $\mathbf{x}_c$. Let us denote R_w^c the rotation matrix from $\mathcal{R}_W$ to $\mathcal{R}_c$:

$$\mathbf{x}_c = R_w^c \mathbf{x}_w.$$

As $\mathcal{R}_c$ is moving, the effect of a small variation Ω at the position (orientation) R_w^c transforms $\mathbf{x}_c$ into $\mathbf{x}_c'$ such that

$$\mathbf{x}_c' = \exp^{\Omega dt} \mathbf{x}_c$$

(see equation (B.1)). Equivalently

$$\mathbf{x}_c' - \mathbf{x}_c = \sum_{m=1}^{m=\infty} \frac{(\Omega dt)^m}{m!} \mathbf{x}_c.$$

At first order in Ωdt, this yields

$$\mathbf{x}_c' - \mathbf{x}_c = \Omega \mathbf{x}_c dt, \text{which is usually written } \partial_t \mathbf{x}_c = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \times \mathbf{x}_c$$

where $\mathbf{a} \times \mathbf{b}$ stand for the usual cross product in $\mathbb{R}^3$. The vector $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$ is the so-called angular velocity. Similarly, a first order approximation of (B.2) yields

$$\partial_t R_w^c = \Omega R_w^c. \tag{C.2}$$

This relation is extended to higher orders using the Hausdorff formula and the adjoint representation of Lie groups and Lie algebras (see [Hall, 2003] and [Selig, 2005]). The matrix R_w^c that describes the relative attitude between two coordinate frames is often called the *direction cosine matrix*. It is mainly used in the robotics communities. Since the angular rates or angular velocities are provided by gyroscopes or gyrometers, it is often desired in inertial navigation to integrate relation (C.2) from a given initial attitude and given embedded sensor measurements to reconstruct the trajectory of a moving object. One must be cautious as the "orthogonal" property of the rotation matrix is easily lost by numerical integration . The proper integration method as described in [Radix, 1980] consists in right-multiplying the current orientation by a so-called *micro-rotator* defined by

$$\delta R = I + \frac{\sin\theta}{\theta}\Theta + \frac{1-\cos\theta}{\theta^2}\Theta^2$$

where Θ is the skew-symmetric matrix associated to the vector $\boldsymbol{\theta}$, $\theta = \|\boldsymbol{\theta}\|$ and $\boldsymbol{\theta}$ is obtained by integration of

$$\partial_t \boldsymbol{\theta} = \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\theta} \times \boldsymbol{\omega}, \quad \boldsymbol{\theta}(0) = \mathbf{0}, t \in [0, dt]. \tag{C.3}$$

This holds for short time intervals dt.

C.1.2 Quaternions

Quaternions were discovered by Hamilton in 1843: this fact was set in stone by the mathematician himself by carving the multiplication of the three basic imaginary units on the Broome Bridge in Dublin (although it appeared later that Gauss had made the same discovery a few years before). Any quaternion $q \in \mathbb{H}$ can be written in the form $q = q_0 + q_1 e_1 + q_2 e_2 + q_3 e_3$ where (q_0, q_1, q_2, q_3) are real numbers and the three imaginary units satisfy

$$e_1^2 = e_2^2 = e_3^2 = -1, \quad e_1 e_2 = e_3, \quad e_2 e_3 = e_1, \quad e_3 e_1 = e_2$$

and the commutation of any of these products yields an opposite result. This is an extension of the theory of complex numbers. Equivalently, a quaternion can be identified with a vector of $\mathbb{R}^4$, itself divided into a scalar (q_0) and a vector $(\mathbf{q})$ of $\mathbb{R}^3$. It follows that the product of two quaternions p and q is

$$pq = \begin{pmatrix} p_0 q_0 - \mathbf{p} \cdot \mathbf{q} \\ p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q} \end{pmatrix}.$$

The norm $\|q\|$ of a quaternion is the Euclidean norm of its representation as a vector of $\mathbb{R}^4$. The conjugate of a quaternion $q = (q_0, \mathbf{q})$ is $q^* = (q_0, -\mathbf{q})$ (like for the complex numbers, the imaginary part is the opposite). Since q and q^* have same norm $\|q\|$, the inverse q^{-1} of q is $\frac{q^*}{\|q\|^2}$. A vector $\mathbf{x}$ of $\mathbb{R}^3$ can be seen as an *imaginary* quaternion where the scalar part is simply zero.

Dot product and cross product are defined in $\mathbb{H}$ to coincide with the same operators in $\mathbb{R}^3$:

$$p \cdot q = \frac{1}{2}(pq^* + qp^*), \quad p \times q = \frac{1}{2}(pq^* - qp^*).$$

It appears then that the operator $R_q : \mathbf{x} \mapsto q^* \mathbf{x} q$ is a rotation as soon as $\|q\| = 1$, since it preserves lengths and angles. The corresponding rotation matrix R_q can be written in the form

$$R_q = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}.$$

Conversely, any rotation can be represented by two unitary opposite quaternions. Identification is made through the *axis-angle* representation of a rotation: any element R of $SO(3)$ can be represented by a scalar θ (the magnitude of rotation) and a unit vector $\mathbf{u}$ (the axis of rotation) (and equivalently by their opposite). Then the quaternion q representing the same rotation is $(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{u})$. More details on this representation and identification with $SO(3)$ can be found in [Radix, 1980] and [Savage, 2000].

It can be proven that the time derivative of the quaternion representing the attitude of a moving object with respect to a fixed frame is related to the angular velocity by

$$\partial_t q = \frac{1}{2} q \boldsymbol{\omega} \tag{C.4}$$

where $\boldsymbol{\omega}$ is the vector of angular velocity in $\mathbb{R}^3$. Numerical integration of quaternions is faster than for direct cosine matrices and quaternions are "lighter" in terms of memory space: they are

therefore favored for applications where real-time computation is necessary. But like for rotation matrices, in case of imprudent numerical computations, quaternions may no longer be unitary. A proper integration scheme is

$$q_{t+dt} = q_t \delta q$$

where δq is defined as

$$\delta q = (\cos\frac{\theta}{2}, \frac{\sin\frac{\theta}{2}}{\theta}\boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is computed by integration as in (C.3).

C.1.3 Euler angles

It is sometimes useful to see a direct cosine matrix as a succession of rotations about predefined axis. They are as many representations (sequences) as possibilities to choose the axis: in addition, the order of the sequence is also decisive since the group of rotations is not commutative. An element of $SO(3)$ is then uniquely defined by three angles of magnitude, three predefined axis, and a sequence. One possibility is to choose three fixed axis, but in the most popular choice, the axis of rotations are defined successively by the previous rotations: this yields the *Euler angles representation*. We detail here one of these representations for a coordinate frame where the axis are defined by the computer vision convention: the z axis is the optical axis, the y axis point downwards and the x axis points towards the right of the image (see figure C.1). Then, the convention adopted in this thesis is to define Euler angles (ψ, θ, ϕ) as

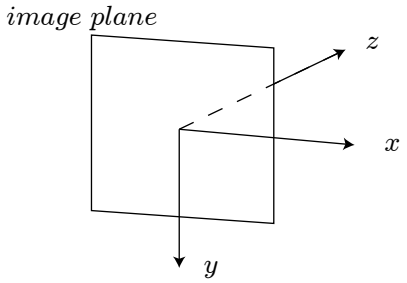


Figure C.1: Definition of the axis used to define the Euler angles of this dissertation

the successive rotations about axis y_0, then about the newly defined axis x_1, and finally about the newly defined axis z_2. This yields for (ψ, θ, ϕ) the usual names of *yaw, pitch* and *roll*. Figure C.2 shows the sequence of rotations that defines the Euler angles. The representation with Euler angles is particularly convenient because a given attitude R is simply defined by matrix product:

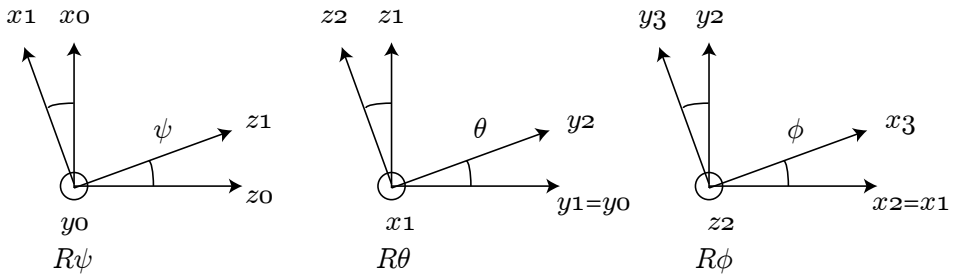


Figure C.2: Euler angles definition

$R = R_\phi R_\theta R_\psi$, with

$$R_\psi = \begin{pmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{pmatrix}, \quad R_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}, \quad R_\phi = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

It is often more meaningful to represent an orientation with angles which are easy to figure than with quaternions or matrices. However, computations with Euler angles are particularly tedious and highly non-linear since they involve trigonometric functions. Moreover, a singularity is encountered at $\theta = \pm\frac{\pi}{2}$. In this thesis, Euler angles are only used to represent an attitude with respect to another as an external output of navigation filters: we will never use them in the observers themselves. This is why relations between angular velocities and time derivatives of Euler angles are not given. They can be found in numerous textbooks including [Radix, 1980], [Savage, 2000].

C.2 Position, linear velocity and acceleration

The position of a rigid-body with respect to another reference frame $\mathcal{R}_W$ is simply the coordinates of the origin $C(t)$ of a reference frame $\mathcal{R}_{cam}$ attached to the moving body expressed in $\mathcal{R}_W$. In this thesis, $\mathcal{R}_W$ always refers to a reference frame attached to a physical point at the earth's surface. Earth's rotation is neglected and $\mathcal{R}_W$ is an inertial reference frame. In addition, Earth's curvature is also neglected. More details on reference frames commonly used in inertial navigation or in radio navigation can be found in [Zarrouati, 1987]. The linear velocity of the moving rigid-body relative to the inertial frame $\mathcal{R}_W$ can be expressed in both frames of reference:

- in $\mathcal{R}_W$, the linear velocity, denoted $\mathbf{V}(t)$ is simply $\partial_t C(t)$;
- in $\mathcal{R}_{cam}$, the linear velocity is denoted $\boldsymbol{v}(t)$ and is expressed as $\boldsymbol{v}(t) = q^* \mathbf{V}(t) q$, where q is the quaternion representing the attitude of the rigid-body at time t (see paragraph C.1.2).

The acceleration $\boldsymbol{\gamma}$ of the moving rigid-body relative to $\mathcal{R}_W$ expressed in $\mathcal{R}_{cam}$ satisfies

$$\partial_t \boldsymbol{v} = \boldsymbol{v} \times \boldsymbol{\omega} + \boldsymbol{\gamma}$$

where $\boldsymbol{\omega}$ is the angular velocity introduced in Section C.1. The reason why acceleration is only introduced in body coordinates is that the sensors used in this thesis are "strap-down" sensors, rigidly attached to the moving body frame. Body coordinates is the natural expression of these sensors. This is also the case for angular velocity. To be precise, embedded accelerometers provide measurements of the specific acceleration $\boldsymbol{a}$

$$\gamma = \boldsymbol{a} - q^* \mathbf{A} q \text{ where } \mathbf{A} = \begin{pmatrix} 0 \\ 9.81 \\ 0 \end{pmatrix}.$$

$q^* \mathbf{A} q$ is the gravity acceleration expressed in the body frame $\mathcal{R}_{cam}$. Finally, the linear velocity expressed in $\mathcal{R}_{cam}$ satisfies

$$\partial_t \boldsymbol{v} = \boldsymbol{v} \times \boldsymbol{\omega} + \boldsymbol{a} - q^* \mathbf{A} q. \tag{C.5}$$

The main issue of inertial navigation with strap-down systems is encrypted in this equation: a precise estimation of attitude is required to properly reconstruct linear velocity. In practice, this also means that errors in acceleration measurements are indistinguishable of imprecise estimation of attitude.

C.3 The Special Euclidean group

Another matrix Lie group of interest is the Special Euclidean group $SE(n)$. The Euclidean group $E(n)$ by definition consists of all one-to-one, onto, distance-preserving maps of $\mathbb{R}^n$ to itself. In fact, one can prove that every element of $E(n)$ can be written uniquely as an orthogonal linear transformation R followed by a translation $\mathbf{T}$: it maps a vector $\mathbf{x} \in \mathbb{R}^n$ into $R\mathbf{x} + \mathbf{T} \in \mathbb{R}^n$. It is denoted $(R, \mathbf{T})$. The product operation for $E(n)$ is the following:

$$(R_1, \mathbf{T}_1)(R_2, \mathbf{T}_2) = (R_1 R_2, \mathbf{T}_1 + R_1 \mathbf{T}_2).$$

The inverse of an element of $E(n)$ is given by

$$(R, \mathbf{T})^{-1} = (R^{-1}, -R^{-1}\mathbf{T}).$$

Since translations are not linear, $E(n)$ is not a subgroup of $GL(n, \mathbb{R})$, but it is isomorphic to a subgroup of $GL(n+1, \mathbb{R})$: to any element $(R, \mathbf{T})$ of $E(n)$, one associates the matrix

$$\begin{pmatrix} R & \mathbf{T} \\ 0_{1\times n} & 1 \end{pmatrix}.$$

Product operator and inversion are then simply expressed as matrix multiplication and inversion. $SE(n)$ is the subgroup of $E(n)$ where the element R is a member of $SO(n)$. It actually describes all rigid-body transformations of $\mathbb{R}^n$. To be more specific, it enables to linearly express the coordinates of a given fixed vector in a moving reference frame with respect to the coordinates of the same vector in a fixed reference frame: it is an extrinsic calibration matrix (see Section 1.1.1).

The Lie algebra of $SE(n)$ is the set of all matrices X of $M_{n+1}(\mathbb{R})$ of the form

$$\begin{pmatrix} \Omega & \mathbf{V} \\ 0_{1\times n} & 0 \end{pmatrix} \text{ such that } \Omega^T = -\Omega.$$

In $\mathbb{R}^3$, the infinitesimal generators at identity can be deduced from the Lie algebra of $SO(3)$ ((C.1)) and the three unit vectors

$$\mathfrak{G}_{vx} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathfrak{G}_{vy} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathfrak{G}_{vz} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

They can be written in the form

$$\mathfrak{G}_1 = \begin{pmatrix} \mathfrak{G}_{\omega x} & 0_{3\times 1} \\ 0_{1\times 3} & 0 \end{pmatrix}, \quad \mathfrak{G}_2 = \begin{pmatrix} \mathfrak{G}_{\omega y} & 0_{3\times 1} \\ 0_{1\times 3} & 0 \end{pmatrix}, \quad \mathfrak{G}_3 = \begin{pmatrix} \mathfrak{G}_{\omega z} & 0_{3\times 1} \\ 0_{1\times 3} & 0 \end{pmatrix},$$

$$\mathfrak{G}_4 = \begin{pmatrix} 0_{3\times 3} & \mathfrak{G}_{vx} \\ 0_{1\times 3} & 0 \end{pmatrix}, \quad \mathfrak{G}_5 = \begin{pmatrix} 0_{3\times 3} & \mathfrak{G}_{vy} \\ 0_{1\times 3} & 0 \end{pmatrix}, \quad \mathfrak{G}_6 = \begin{pmatrix} 0_{3\times 3} & \mathfrak{G}_{vz} \\ 0_{1\times 3} & 0 \end{pmatrix}.$$

Consider a vector expressed in a reference frame $\mathcal{R}_{cam}$ attached to a moving object with respect to its coordinates in another fixed reference frame $\mathcal{R}_W$ by

$$\mathbf{x}_c = K\mathbf{x}_w$$

where K is a member of the Special Euclidean Group $SE(n)$. A small variation of K, in other words of the position and/or orientation of the moving object, is represented by an element $(\Omega, \mathbf{V})$ of the Lie algebra, with can be written in the form

$$(\Omega, \mathbf{V}) \equiv \sum_i \alpha_i \mathfrak{G}_i$$

This variation has the following effect on $\mathbf{x}_c$ (see (B.1)):

$$\mathbf{x}'_c = \exp^{\sum_i \alpha_i \mathfrak{G}_i} \mathbf{x}_c. \tag{C.6}$$

Similarly (see equation (B.2)), a small variation of orientation and position applied to an element K of $SE(n)$ yields an new element K' which is given by

$$K' = \exp^{\sum_i \alpha_i \mathfrak{G}_i} K. \tag{C.7}$$

Bibliography

[Abbott and Powell, 1999] Abbott, E. and Powell, D. (1999). Land-vehicle navigation using GPS. *Proceedings of the IEEE*, 87(1):145–162.

[Abdursul et al., 2004] Abdursul, R., Inaba, H., and Ghosh, B. K. (2004). Nonlinear observers for perspective time-invariant linear systems. *Automatica*, 40(3):481–490.

[Ackerman, 2013] Ackerman, E. (2013). Google gets in your face: Google glass offers a slightly augmented version of reality. *IEEE Spectrum*, 50(1):26–29.

[Aghajan and Cavallaro, 2009] Aghajan, H. and Cavallaro, A. (2009). *Multi-camera networks: principles and applications*. Academic Press.

[Angeli et al., 2008] Angeli, A., Filliat, D., Doncieux, S., and Meyer, J.-A. (2008). Real-time visual loop-closure detection. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

[Asteriadis et al., 2013] Asteriadis, S., Chatzitofis, A., Zarpalas, D., Alexiadis, D., and Daras, P. (2013). Estimating human motion from multiple Kinect sensors. In *ACM International Conference Proceeding Series*.

[Audras et al., 2011] Audras, C., Comport, A., Meilland, M., and Rives, P. (2011). Real-time dense appearance-based SLAM for RGB-D sensors. In *Australasian Conf. on Robotics and Automation*.

[Azuma et al., 1997] Azuma, R. T. et al. (1997). A survey of augmented reality. *Presence*, 6(4):355–385.

[Baker and Nayar, 1998] Baker, S. and Nayar, S. K. (1998). A theory of catadioptric image formation. In *Computer Vision, 1998. Sixth International Conference on*, pages 35–42. IEEE.

[Baker and Nayar, 1999] Baker, S. and Nayar, S. K. (1999). A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196.

[Barreto, 2006] Barreto, J. P. (2006). A unifying geometric representation for central projection systems. *Computer Vision and Image Understanding*, 103(3):208–217.

[Bezick et al., 2010] Bezick, S. M., Pue, A. J., and Patzelt, C. M. (2010). Inertial navigation for guided missile systems. *Johns Hopkins APL technical digest*, 28:331–342.

[Bonnabel et al., 2008] Bonnabel, S., Martin, P., and Rouchon, P. (2008). Symmetry-preserving observers. *Automatic Control, IEEE Transactions on*, 53(11):2514–2526.

[Bonnabel and Rouchon, 2009] Bonnabel, S. and Rouchon, P. (2009). Fusion of inertial and visual : a geometrical observer-based approach. *2nd Mediterranean Conference on Intelligent Systems and Automation (CISA'09)*, 1107:54–58.

[Borrero and Marquez, 2013] Borrero, A. and Marquez, J. (2013). Interaction of real robots with virtual scenarios through augmented reality: Application to robotics teaching/learning by means of remote labs. *International Journal of Engineering Education*, 29(3):788–798.

[Boutteau, 2010] Boutteau, R. (2010). *Reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions.* PhD thesis, Université de Rouen.

[Bovcon et al., 2013] Bovcon, N., Vaupotic, A., Klemenc, B., and Solina, F. (2013). "Atlas 2012" augmented reality: A case study in the domain of fine arts. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7946 LNCS:477–496.

[Brenner, 2005] Brenner, C. (2005). Building reconstruction from images and laser scanning. *International Journal of Applied Earth Observation and Geoinformation*, 6(3-4):187–198.

[Brezis, 1983] Brezis, H. (1983). *Analyse foncionnelle*. Masson,Paris.

[Bristeau et al., 2011] Bristeau, P., Callou, F., Vissière, D., and Petit, N. (2011). The navigation and control technology inside the AR. drone micro UAV. In *IFAC World Congress*, volume 18, pages 1477–1484.

[Calderita et al., 2013] Calderita, L., Bandera, J., Bustos, P., and Skiadopoulos, A. (2013). Model-based reinforcement of Kinect depth data for human motion capture applications. *Sensors (Switzerland)*, 13(7):8835–8855.

[Chambolle, 2004] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1):89–97.

[Chambolle and Pock, 2010] Chambolle, A. and Pock, T. (2010). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, pages 1–26.

[Chatfield, 1997] Chatfield, A. (1997). *Fundamentals of High Accuracy Inertial Navigation*, volume 174. Aiaa.

[Civera et al., 2008] Civera, J., Davison, A., and Montiel, J. (2008). Inverse depth parametrization for monocular SLAM. *Robotics, IEEE Transactions on*, 24(5):932 –945.

[Correa et al., 2012] Correa, D. S. O., Sciotti, D. F., Prado, M. G., Sales, D. O., Wolf, D. F., and Osório, F. S. (2012). Mobile robots navigation in indoor environments using Kinect sensor. In *Critical Embedded Systems (CBSEC), 2012 Second Brazilian Conference on*, pages 36–41. IEEE.

[Dahl et al., 2010] Dahl, O., Wang, Y., Lynch, A. F., and Heyden, A. (2010). Observer forms for perspective systems. *Automatica*, 46(11):1829–1834.

[Daniel Herrera et al., 2012] Daniel Herrera, C., Kannala, J., and Heikkilä, J. (2012). Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064.

[Davison et al., 2007] Davison, A., Reid, I., Molton, N., and Stasse, O. (2007). Monoslam: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067.

[de Sorbier et al., 2012] de Sorbier, F., Shiino, H., and Saito, H. (2012). Violin pedagogy for finger and bow placement using augmented reality. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–5. IEEE.

[Devernay and Faugeras, 2001] Devernay, F. and Faugeras, O. (2001). Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24.

[Dixon et al., 2003] Dixon, W., Fang, Y., Dawson, D., and Flynn, T. (2003). Range identification for perspective vision systems. *Automatic Control, IEEE Transactions on*, 48(12):2232 – 2238.

[Dorveaux, 2011] Dorveaux, E. (2011). *Navigation magnéto-inertielle: principes et application à un système podométrique indoor*. PhD thesis, École Nationale Supérieure des Mines de Paris.

[Doubrovine et al., 1987] Doubrovine, B., Novikov, S. P., Fomenko, A. T., and Kotliar, V. (1987). *Géométrie contemporaine: méthodes et applications*. Ed. Mir.

[Du et al., 2011] Du, H., Henry, P., Ren, X., Cheng, M., Goldman, D. B., Seitz, S. M., and Fox, D. (2011). Interactive 3D modeling of indoor environments with a consumer depth camera. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 75–84. ACM.

[Dube and Zell, 2011] Dube, D. and Zell, A. (2011). Real-time plane extraction from depth images with the randomized hough transform. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1084–1091. IEEE.

[Eade and Drummond, 2006] Eade, E. and Drummond, T. (2006). Scalable monocular SLAM. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 469–476. IEEE.

[Edwards, 2013] Edwards, C. (2013). Better than reality? *Engineering and Technology*, 8(4):28–31.

[Endres et al., 2012] Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W. (2012). An evaluation of the RGB-D SLAM system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1691–1696. IEEE.

[Engels et al., 2006] Engels, C., Stewénius, H., and Nistér, D. (2006). Bundle adjustment rules. *Photogrammetric computer vision*, 2.

[Faugeras, 1993] Faugeras, O. (1993). *Three-dimensional computer vision: a geometric viewpoint*. the MIT Press.

[Faugeras and Lustman, 1988] Faugeras, O. D. and Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(03):485–508.

[Faurre, 1971] Faurre, P. (1971). Navigation inertielle et filtrage stochastique. *Méthodes mathématiques de l'informatique. Dunod*.

[Fischler and Bolles, 1981] 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*, 24:381–395.

[Gaskell, 1998] Gaskell, I. (1998). *Vermeer Studies: studies in the history of art, Center for Advanced Study in the Visual Arts: symposium papers XXXIII*. National gallery of art.

[Gavish et al., 2013] Gavish, N., Gutiérrez, T., Webel, S., Rodríguez, J., Peveri, M., Bockholt, U., and Tecchia, F. (2013). Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, (ahead-of-print):1–21.

[Georgiev et al., 2011] Georgiev, K., Creed, R. T., and Lakaemper, R. (2011). Fast plane extraction in 3D range data based on line segments. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3808–3815. IEEE.

[Geyer and Daniilidis, 2000] Geyer, C. and Daniilidis, K. (2000). A unifying theory for central panoramic systems and practical implications. In *Computer Vision ECCV 2000*, pages 445–461. Springer.

[Girbacia et al., 2013] Girbacia, F., Butnariu, S., Orman, A., and Postelnicu, C. (2013). Virtual restoration of deteriorated religious heritage objects using augmented reality technologies. *European Journal of Science and Theology*, 9(2):223–231.

[Gold et al., 1998] Gold, S., Rangarajan, A., Lu, C.-P., Pappu, S., and Mjolsness, E. (1998). New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031.

[Gonzalez-Jorge et al., 2013] Gonzalez-Jorge, H., Riveiro, B., Vazquez-Fernandez, E., Martínez-Sánchez, J., and Arias, P. (2013). Metrological evaluation of Microsoft Kinect and Asus Xtion sensors. *Measurement: Journal of the International Measurement Confederation*, 46(6):1800–1806.

[Graham et al., 2013] Graham, M., Zook, M., and Boulton, A. (2013). Augmented reality in urban places: Contested content and the duplicity of code. *Transactions of the Institute of British Geographers*, 38(3):464–479.

[Grasmair, 2010] Grasmair, M. (2010). Non-smooth variational methods in imaging and inverse problems.

[Grossberg and Nayar, 2001] Grossberg, M. D. and Nayar, S. K. (2001). A general imaging model and a method for finding its parameters. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 108–115. IEEE.

[Guerrero and Uribe-Quevedo, 2013] Guerrero, C. and Uribe-Quevedo, A. (2013). Kinect-based posture tracking for correcting positions during exercise. *Studies in health technology and informatics*, 184:158–160.

[Guichard and Morel, 2004] Guichard, F. and Morel, J.-M. (2004). Contrast invariant image analysis and PDEs. *preprint*.

[Guo et al., 2013] Guo, X., Lopez, L., Yu, Z., Steiner, K., Barner, K., Bauer, T., and Yu, J. (2013). A portable immersive surgery training system using RGB-D sensors. *Studies in health technology and informatics*, 184:161–167.

[Hall, 2003] Hall, B. (2003). *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer.

[Hamel et al., 2011] Hamel, T., Mahony, R., Trumpf, J., Morin, P., and Hua, M. (2011). Homography estimation on the special linear group based on direct point correspondence. In *Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 7902–7908.

[Han and Seo, 2013] Han, T. and Seo, Y.-H. (2013). Mixed reality system for virtual interior design. *International Journal of Smart Home*, 7(3):133–142.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK.

[Hartley and Zisserman, 2000] Hartley, R. and Zisserman, A. (2000). *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press.

[Henry et al., 2012] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research*, 31(5):647–663.

[Heyden and Dahl, 2009] Heyden, A. and Dahl, O. (2009). Provably convergent on-line structure and motion estimation for perspective systems. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 751 –758.

[Horn and Schunck, 1981] Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

[Huang et al., 2011] Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2011). Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Int. Symposium on Robotics Research (ISRR),(Flagstaff, Arizona, USA)*.

[Izadi et al., 2011] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al. (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.

[Kahn, 2013] Kahn, S. (2013). Reducing the gap between augmented reality and 3D modeling with real-time depth imaging. *Virtual Reality*, 17(2):111–123.

[Kaplan and Hegarty, 2005] Kaplan, E. D. and Hegarty, C. J. (2005). *Understanding GPS: principles and applications*. Artech house.

[Karagiannis and Astolfi, 2005] Karagiannis, D. and Astolfi, A. (2005). A new solution to the problem of range identification in perspective vision systems. *Automatic Control, IEEE Transactions on*, 50(12):2074 – 2077.

[Khan et al., 2013] Khan, W., Xiang, Y., Aalsalem, M., and Arshad, Q. (2013). Mobile phone sensing systems: A survey. *IEEE Communications Surveys and Tutorials*, 15(1):402–427.

[Khoshelham and Elberink, 2012] Khoshelham, K. and Elberink, S. (2012). Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454.

[Klein and Murray, 2007] 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.

[Klein and Murray, 2009] Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 83–86. IEEE.

[Lefèvre, 2007] Lefèvre, W. (2007). *Inside the Camera Obscura: Optics and Art Under the Spell of the Projected Image*. Max-Planck-Institut für Wissenschaftsgeschichte.

[LeVeque, 1992] LeVeque, R. J. (1992). *Numerical methods for conservation laws*. Birkhauser.

[Li and Hartley, 2007] Li, H. and Hartley, R. (2007). The 3D-3D registration problem revisited. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

[Li et al., 2013] Li, X.-J., Xie, B., and Ye, F. (2013). Research and application of online product display technology based on augmented reality. *Information Technology Journal*, 12(6):1134–1142.

[Lim and Lee, 2009] Lim, H. and Lee, Y. S. (2009). Real-time single camera SLAM using fiducial markers. In *ICCAS-SICE, 2009*, pages 177–182. IEEE.

[Lin et al., 2013] Lin, C.-M., Huang, C.-Y., Yang, D.-Y., Lin, T.-C., Chung, C.-C., and Lin, J.-H. (2013). An android-based interactive museum exhibit system using wireless sensing and augmented reality technologies. *Advanced Materials Research*, 684:509–512.

[Lohmiller and Slotine, 1996] Lohmiller, W. and Slotine, J.-J. (1996). On metric observers for nonlinear systems. In *Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on*, pages 320–326.

[Lohmiller and Slotine, 1998] Lohmiller, W. and Slotine, J. J. (1998). On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696.

[Lothe et al., 2010] 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.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Luca et al., 2008] Luca, A. D., Oriolo, G., and Giordano, P. R. (2008). Feature depth observation for image-based visual servoing: Theory and experiments. *I. J. Robotic Res.*, pages 1093–1116.

[Mahony and Hamel, 2005] Mahony, R. and Hamel, T. (2005). Image-based visual servo control of aerial robotic systems using linear image features. *Robotics, IEEE Transactions on*, 21(2):227–239.

[Mahony et al., 2008] Mahony, R., Hamel, T., and Pflimlin, J.-M. (2008). Nonlinear complementary filters on the special orthogonal group. *Automatic Control, IEEE Transactions on*, 53(5):1203–1218.

[Maisto et al., 2013] Maisto, M., Panella, M., Liparulo, L., and Proietti, A. (2013). An accurate algorithm for the identification of fingertips using an RGB-D camera. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(2):272–283.

[Marquardt, 1963] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441.

[Martin and Salaün, 2010] Martin, P. and Salaün, E. (2010). Design and implementation of a low-cost observer-based attitude and heading reference system. *Control Engineering Practice*, 18(7):712–722.

[Matthies et al., 1989] Matthies, L., Kanade, T., and Szeliski, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238.

[Mitiche and reza Mansouri, 2004] Mitiche, A. and reza Mansouri, A. (2004). On convergence of the horn and schunck optical-flow estimation method. *IEEE Transactions on Image Processing*, 13:848–852.

[Montemerlo et al., 2003] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico. IJCAI.

[Mousavi Hondori et al., 2013] Mousavi Hondori, H., Khademi, M., Dodakian, L., Cramer, S., and Lopes, C. (2013). A spatial augmented reality rehab system for post-stroke hand rehabilitation. *Studies in health technology and informatics*, 184:279–285.

[Munaro et al., 2013] Munaro, M., Ballin, G., Michieletto, S., and Menegatti, E. (2013). 3D flow estimation for human action recognition from colored point clouds. *Biologically Inspired Cognitive Architectures*.

[Newcombe and Davison, 2010] Newcombe, R. A. and Davison, A. J. (2010). Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1498–1505. IEEE.

[Nister, 2004] Nister, D. (2004). An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756 –770.

[Nistér et al., 2004] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652. IEEE.

[Olver, 1999] Olver, P. J. (1999). *Classical invariant theory*, volume 44. Cambridge University Press.

[Ong et al., 2008] Ong, S., Yuan, M., and Nee, A. (2008). Augmented reality applications in manufacturing: A survey. *International Journal of Production Research*, 46(10):2707–2742.

[Peyré, 2013] Peyré, G. (2013). Advanced signal processing. https://www.ceremade.dauphine.fr/peyre/numerical-tour/book/AdvancedSignalProcessing.pdf.

[Pollefeys, 1999] Pollefeys, M. (1999). *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, KATHOLIEKE UNIVERSITEIT LEUVEN.

[Pollefeys et al., 2004] Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., and Koch, R. (2004). Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232.

[Radix, 1980] Radix, J. (1980). Strap-down inertial systems. *Toulouse Cepadues Editions*, 1.

[Rosten and Drummond, 2006] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer.

[Rublee et al., 2011] 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.

[Rudin et al., 1992] Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268.

[Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE.

[Ruzanka et al., 2013] Ruzanka, S., Chang, B., and Behar, K. (2013). There's an app for that shirt! evaluation of augmented reality tracking methods on deformable surfaces for fashion design. In *IS&T/SPIE Electronic Imaging*, pages 86490D–86490D. International Society for Optics and Photonics.

[Santoso and Gook, 2012] Santoso, M. and Gook, L. (2012). ARobot: Development of 3rd person shooting game and handheld augmented reality. In *Proceedings - VRCAI 2012: 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 55–58.

[Sassano et al., 2010] Sassano, M., Carnevale, D., and Astolfi, A. (2010). Observer design for range and orientation identification. *Automatica*, 46(8):1369–1375.

[Savage, 2000] Savage, P. (2000). *Strapdown Analytics*. Strapdown Associates.

[Scherer et al., 2012] Scherer, S. A., Dube, D., and Zell, A. (2012). Using depth in visual simultaneous localisation and mapping. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5216–5221. IEEE.

[Selig, 2005] Selig, J. (2005). Lie groups and lie algebras in robotics. In *Computational Noncommutative Algebra and Applications*, pages 101–125. Springer.

[Serre, 1999] Serre, D. (1999). *Systems of Conservation Laws 1: Hyperbolicity, entropies, shock waves*, volume 1. Cambridge University Press.

[Shiva and Raajan, 2013] Shiva, G. and Raajan, N. (2013). Augmented reality: The future tense of 3D advertisement. In *2013 International Conference on Computer Communication and Informatics, ICCCI 2013*.

[Skog and Handel, 2009] Skog, I. and Handel, P. (2009). In-car positioning and navigation technologies; a survey. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):4 –21.

[Smisek et al., 2011] Smisek, J., Jancosek, M., and Pajdla, T. (2011). 3D with Kinect. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1160.

[Smith et al., 1990] Smith, R., Self, M., and Cheeseman, P. (1990). *Estimating uncertain spatial relationships in robotics*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA.

[Snavely et al., 2008] Snavely, N., Seitz, S. M., and Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210.

[Soatto et al., 1994] Soatto, S., Perona, P., Frezza, R., and Picci, G. (1994). Motion estimation via dynamic vision. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 4, pages 3253 –3258.

[Speidel et al., 2013] Speidel, S., Krappe, S., Röhl, S., Bodenstedt, S., Müller-Stich, B., and Dillmann, R. (2013). Robust feature tracking for endoscopic pose estimation and structure recovery. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 8671.

[Sprickerhof et al., 2011] Sprickerhof, J., Nüchter, A., Lingemann, K., and Hertzberg, J. (2011). A heuristic loop closing technique for large-scale 6D SLAM. *Automatika–Journal for Control, Measurement, Electronics, Computing and Communications*, 52(3).

[Steinbrucker et al., 2011] Steinbrucker, F., Sturm, J., and Cremers, D. (2011). Real-time visual odometry from dense RGB-D images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 719–722. IEEE.

[Stewénius et al., 2006] Stewénius, H., Engels, C., and Nistér, D. (2006). Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294.

[Stoker, 1969] Stoker, J. J. (1969). Differential geometry. *Pure and Applied Mathematics*, 20.

[Stowers et al., 2011] Stowers, J., Hayes, M., and Bainbridge-Smith, A. (2011). Altitude control of a quadrotor helicopter using depth map from microsoft Kinect sensor. In *2011 IEEE International Conference on Mechatronics, ICM 2011 - Proceedings*, pages 358–362.

[Strasdat et al., 2010a] Strasdat, H., Montiel, J. M. M., and Davison, A. (2010a). Scale drift-aware large scale monocular SLAM. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain.

[Strasdat et al., 2010b] Strasdat, H., Montiel, J. M. M., and Davison, A. J. (2010b). Real-time monocular SLAM: Why filter? In *ICRA*, pages 2657–2664.

[Szeliski, 2010] Szeliski, R. (2010). *Computer vision, Algorithms and Applications*. Springer.

[Taguchi et al., 2012] Taguchi, Y., Jian, Y.-D., Ramalingam, S., and Feng, C. (2012). SLAM using both points and planes for hand-held 3D sensors. In *ISMAR 2012 - 11th IEEE International Symposium on Mixed and Augmented Reality 2012, Science and Technology Papers*, pages 321–322.

[Uliana et al., 1997] Uliana, M., Andreucci, F., and Papalia, B. (1997). The navigation system of an autonomous underwater vehicle for antarctic exploration. In *OCEANS '97. MTS/IEEE Conference Proceedings*, volume 1, pages 403 –408 vol.1.

[Vissiere, 2008] Vissiere, D. (2008). *Guidance, Navigation and Control Solutions for Unmanned Heterogenous Vehicles in a Collaborative Mission.* PhD thesis, MINES ParisTech, Paris, France.

[Vissière et al., 2007] Vissière, D., Martin, A., and Petit, N. (2007). Using spatially distributed magnetometers to increase IMU-based velocity estimation in perturbed areas. In *Proc. of the 46th IEEE Conf. on Decision and Control*, pages 4924–4931.

[Vissière et al., 2008] Vissière, D., Martin, A., and Petit, N. (2008). Système fournissant la vitesse et la position d' un corps en utilisant les variations du champ magnétique évaluées grâce aux mesures de un ou des magnétomètres et de une ou des centrales inertielles. *Patent FR2914739 (A1).*

[Vogt and Shingles, 2013] Vogt, F. and Shingles, L. (2013). Augmented reality in astrophysics. *Astrophysics and Space Science*, pages 1–14.

[Wedel et al., 2008] Wedel, A., Pock, T., Braun, J., Franke, U., and Cremers, D. (2008). Duality tv-l1 flow with fundamental matrix prior. In *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, pages 1–6. IEEE.

[Wedel et al., 2009] Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2009). An improved algorithm for TV-L1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Springer.

[Wei et al., 2013] Wei, Y.-M., Kang, L., Yang, B., and Wu, L.-D. b. (2013). Applications of structure from motion: A survey. *Journal of Zhejiang University: Science C*, 14(7):486–494.

[Whelan et al., 2013a] Whelan, T., Johannsson, H., Kaess, M., Leonard, J., and McDonald, J. (2013a). Robust real-time visual odometry for dense RGB-D mapping. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Karlsruhe, Germany.

[Whelan et al., 2012] Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. (2012). Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia.

[Whelan et al., 2013b] Whelan, T., Kaess, M., Leonard, J., and McDonald, J. (2013b). Deformation-based loop closure for large scale dense RGB-D SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS*, Tokyo, Japan. (to appear).

[Williams et al., 2007] Williams, B., Klein, G., and Reid, I. (2007). Real-time SLAM relocalisation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

[Woun and Tan, 2012] Woun, B. and Tan, G. (2012). Deployment of augmented reality interactions in games. In *Proceedings of the 2012 International Conference on Artificial Intelligence, ICAI 2012*, volume 2, pages 1034–1037.

[Yang et al., 2013] Yang, M.-D., Chao, C.-F., Huang, K.-S., Lu, L.-Y., and Chen, Y.-P. (2013). Image-based 3D scene reconstruction and exploration in augmented reality. *Automation in Construction*, 33:48–60.

[Zarrouati et al., 2012a] Zarrouati, N., Aldea, E., Pierre, R., et al. (2012a). Estimation dense de profondeur combinant approches variationnelles et observateurs asymptotiques. In *Actes de la conférence RFIA 2012*.

[Zarrouati et al., 2012b] Zarrouati, N., Aldea, E., and Rouchon, P. (2012b). Robust depth regularization explicitly constrained by camera motion. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3606–3609. IEEE.

[Zarrouati et al., 2012c] Zarrouati, N., Aldea, E., and Rouchon, P. (2012c). SO(3)-invariant asymptotic observers for dense depth field estimation based on visual data and known camera motion. In *American Control Conference, Montreal*. (arXiv:1103.2539v2).

[Zarrouati et al., 2012d] Zarrouati, N., Hillion, M., and Petit, N. (2012d). Curvilinear velocity estimation using low-quality stereo-vision systems and a gyrometer. In *American Control Conference (ACC), 2012*, pages 4108–4115. IEEE.

[Zarrouati, 1987] Zarrouati, O. (1987). *Trajectoires spatiales.* Cepadues-Editions.

[Zarrouati-Vissière et al., 2013] Zarrouati-Vissière, N., Beauchard, K., and Rouchon, P. (2013). Inertial-sensor bias estimation from brightness/depth images and based on SO(3)-invariant integro/partial-differential equations on the unit sphere. *submitted to SIAM Journal on Control and Optimization*. (arXiv:1309:2193).

[Zarrouati-Vissiere et al., 2012] Zarrouati-Vissiere, N., Rouchon, P., and Beauchard, K. (2012). Rotational and translational bias estimation based on depth and image measurements. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 6627–6634. IEEE.

[Zarrouati-Vissière et al., 2013a] Zarrouati-Vissière, N., Beauchard, K., and Rouchon, P. (2013a). 6 degree-of-freedom motion estimation from depth images and magneto-inertial data for trajectory reconstruction. *submitted to Robotics, IEEE Transactions on.*

[Zarrouati-Vissière et al., 2013b] Zarrouati-Vissière, N., Rouchon, P., Caruso, D., Hillion, M., and Beauchard, K. (déposé le 19 novembre 2013b). Mesure de mouvement par traitement de la profondeur. In *Demande de brevet n^o 1361347*.

[Zhang, 1994] Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334.

[Zhang, 2001] Zhang, Z. (2001). Microsoft easy camera calibration tool. http://research.microsoft.com/en-us/um/people/zhang/calib/.

[Zhang, 2012] Zhang, Z. (2012). Microsoft Kinect sensor and its effect. *IEEE Multimedia*, 19(2):4–10.

[Zhang and Faugeras, 1992] Zhang, Z. and Faugeras, O. D. (1992). Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7(3):211–241.

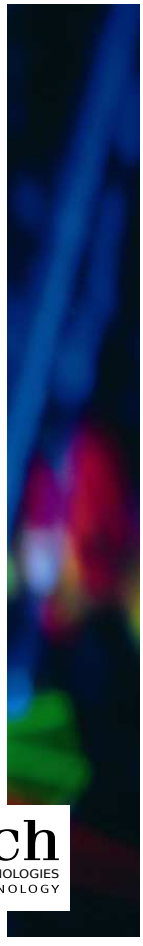
[Zollmann and Reitmayr, 2012] Zollmann, S. and Reitmayr, G. (2012). Dense depth maps from sparse models and image coherence for augmented reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, pages 53–60.

La réalité augmentée:
fusion de vision et navigation.

Résumé: Cette thèse a pour objet l'étude d'algorithmes pour des applications de réalité visuellement augmentée. Plusieurs besoins existent pour de telles applications, qui sont traités en tenant compte de la contrainte d'indistinguabilité de la profondeur et du mouvement linéaire dans le cas de l'utilisation de systèmes monoculaires. Pour insérer en temps réel de manière réaliste des objets virtuels dans des images acquises dans un environnement arbitraire et inconnu, il est non seulement nécessaire d'avoir une perception 3D de cet environnement à chaque instant, mais également d'y localiser précisément la caméra. Pour le premier besoin, on fait l'hypothèse d'une dynamique de la caméra connue, pour le second on suppose que la profondeur est donnée en entrée: ces deux hypothèses sont réalisables en pratique. Les deux problèmes sont posés dans le contexte d'un modèle de caméra sphérique, ce qui permet d'obtenir des équations de mouvement invariantes par rotation pour l'intensité lumineuse comme pour la profondeur. L'observabilité théorique de ces problèmes est étudiée à l'aide d'outils de géométrie différentielle sur la sphère unité Riemanienne. Une implémentation pratique est présentée: les résultats expérimentaux montrent qu'il est possible de localiser une caméra dans un environnement inconnu tout en cartographiant précisément cet environnement.
Mots clés: Réalité augmentée, systèmes de navigation, vision, algorithmes SLAM, estimation de biais, profondeur, observabilité, géométrie différentielle, capteurs MEMS, données RGBD

Augmented reality:
the fusion of vision and navigation.

Abstract: The purpose of this thesis is to study algorithms for visual augmented reality. Different requirements of such an application are addressed, with the constraint that the use of a monocular system makes depth and linear motion indistinguishable. The real-time realistic insertion of virtual objects in images of a real arbitrary environment yields the need for a dense 3D perception of this environment on one hand, and a precise localization of the camera on the other hand. The first requirement is studied under an assumption of known dynamics, and the second under the assumption of known depth: both assumptions are practically realizable. Both problems are posed in the context of a spherical camera model, which yields $SO(3)$-invariant dynamical equations for light intensity and depth. The study of theoretical observability requires differential geometry tools for the Riemannian unit sphere. Practical implementation on a system is presented and experimental results demonstrate the ability to localize a camera in a unknown environment while precisely mapping this environment.
Keywords: Augmented reality, navigation systems, vision, SLAM algorithms, bias estimation, depth, observability, differential geometry, MEMS sensors, RGBD data



MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY