

Stereo-Vision-Aided Inertial Navigation

PhD Thesis

submitted in partial fulfillment for the degree of

Dr. rer. nat.

to the Department of Mathematics and Computer Science
of the Freie Universität Berlin



by

Denis Grießbach

Berlin, 2014

Gutachter:

Prof. Dr. Raúl Rojas

Prof. Dr. rer. nat. Ralf Reulke

Datum der Disputation: 21.05.2015

Abstract

Reliable information about position and attitude is an essential requirement for many applications. The work expounded in this paper aims at a tight integration of low-cost inertial navigation and stereo vision to obtain this information. The method I present here is based on passive measurements and does not rely on external referencing. Thus, it provides a navigation solution for unknown indoor and outdoor environments.

Special attention is paid to a stereo vision-based system, capable of providing egomotion measurements with six degrees of freedom. Natural landmarks are extracted and tracked in consecutive image pairs aided by inertial measurements to constrain the correspondence problem. This effectively reduces the computational effort and avoids uncertainties that stem from mismatches, resulting in a robust tracking algorithm that runs in real time. In turn, the extracted egomotion is used to constrain the inertial sensor drift. In addition, the measured gravity serves as a vertical reference, stabilizing the navigation solution.

Based on dead reckoning, inertial navigation is widely used and has been studied in almost every aspect. To correct the inertial sensor errors, these systems are periodically provided with external data, e.g. a Global Positioning System (GPS) signal. The reliable short term properties of inertial data are complemented by the long-term stability of the external measurement. Although such methods do work very well, a similar solution is needed for navigating in difficult environments with erroneous or no external reference whatsoever. In such situations, using independent measurement systems like barometers, odometers, or vision-based systems is especially advantageous.

Hence, I present an approach for a heterogeneous multi-sensor system that involves both hardware and software, wherein aspects like synchronization, registration, and calibration of the sensor system are considered. As the optical system is of major importance, I've developed a new method that provides fast and reliable camera calibration. Herein, I also present my extensive analysis of possible error sources throughout the system. The result of this integration of stereo vision and inertial navigation is then proven in various pedestrian navigation tasks.

Zusammenfassung

Das Wissen um Position und Lage spielt für viele Anwendungen eine entscheidende Rolle. Diese Arbeit zeigt die enge Verknüpfung von inertialer Navigation und optischen Informationen, um dieses Wissen zu gewinnen. Die vorgestellte Methode basiert auf rein passiven Messungen und ist unabhängig von externen Referenzen. Damit eignet sie sich für die Navigation in unbekanntem Umgebungen sowohl in Innen- als auch in Außenbereichen.

Ein Hauptaugenmerk liegt auf der Gewinnung von Bewegungsinformationen aus optischen Systemen. Unterstützt durch inertielle Bewegungsdaten werden natürliche Merkmale einer Umgebung verfolgt, um daraus die Eigenbewegung einer Stereokamera abzuleiten, die dazu beiträgt, die inertielle Sensordrift zu kompensieren. Die Nutzung der inertialen Messungen liefert einen signifikanten Beitrag zur Vermeidung von Fehlzuordnungen und zur Reduzierung von Rechenleistung. Die gemessene Erdbeschleunigung dient als vertikale Referenz, welche die Navigationslösung zusätzlich stabilisiert.

Die inertielle Navigation, ein Koppelnavigationsverfahren, hat eine große Bedeutung und ist Gegenstand vieler Forschungsarbeiten. Bei der Koppelnavigation wird aus der zuletzt bekannten Position sowie der gemessenen Geschwindigkeit und Zeit die aktuelle Position bestimmt. Zusätzliche langzeitstabile externe Messungen, wie z.B. GPS, ergänzen die guten Kurzzeiteigenschaften der inertialen Navigation und begrenzen aufsummierte Fehler. Obwohl das Verfahren für viele Anwendungen sehr gut funktioniert, zeigt es Schwächen, wenn die stützende Messung fehlerhaft oder nicht verfügbar ist. Der Einsatz unabhängiger Systeme wie z.B. optischer Sensoren, Barometer oder Odometer stellt daher eine sinnvolle Ergänzung dar.

Zunächst stelle ich einen allgemeinen Ansatz für ein Multisensor-System zur Positions- und Lagemessung vor. Hierbei beleuchte ich die gesamte Systemkette, beginnend mit der Auslegung der Hardware-Komponenten über die Datenerfassung und die Kalibrierung bis zur Ableitung höherwertiger Informationen aus fusionierten Sensordaten. Insbesondere die detaillierte Betrachtung möglicher Fehlerquellen liefert einen wichtigen Beitrag zum Systemverständnis. Anhand einiger Navigationsaufgaben im Innen- und Außenbereich stelle ich beispielhaft das Ergebnis einer Integration von optischen- und inertialen Messdaten dar.

Contents

Preface	v
Notation	vii
1 Introduction	1
1.1 Motivation	2
1.2 Contributions	3
1.3 Organization of the Thesis	4
2 Related Work	5
2.1 General Navigation	6
2.2 Visual Navigation	7
2.3 Vision-Aided Inertial Navigation	9
3 Inertial Navigation	11
3.1 Inertial Navigation System	11
3.1.1 Coordinate Frames	12
3.1.2 Attitude Algorithm	13
3.1.3 Position Algorithm	14
3.2 Inertial Measurement Unit	15
3.2.1 Sensor Error Model	16
3.2.2 Gyroscope	18
3.2.3 Accelerometer	21
3.3 Inclination Sensor	24
4 Computer Vision	25
4.1 Single Camera View	25

4.1.1	Camera Model	26
4.1.2	Lens Distortion	28
4.2	Stereo Camera View	31
4.2.1	Epipolar Geometry	31
4.2.2	Triangulation	32
4.3	Pose Estimation	34
4.3.1	Feature Extraction	36
4.3.2	Feature Matching	37
4.4	Calibration and Registration	40
4.4.1	Diffraction Optical Element	40
4.4.2	Single View Calibration	42
4.4.3	Stereo Registration	43
4.5	Experimental Setup	45
5	Sensor Fusion	47
5.1	Sensor Registration	47
5.1.1	IMU to Inclinometer Sensor	48
5.1.2	IMU to Camera	48
5.1.3	Time Line Registration	49
5.2	Navigation Filter	49
5.2.1	Error State Filter	50
5.2.2	State Transition Model	51
5.2.3	Measurement Model	53
6	System Integration and Application	55
6.1	Hardware Concept	55
6.2	Software Framework	57
6.2.1	Middleware	59
6.2.2	Application Network	59
6.3	Experimental Results	60
6.3.1	Ground Truth Data	61
6.3.2	System Initialization	61
6.3.3	Experimental Data Sets	63

7	Conclusion and Future Work	71
7.1	Summary	71
7.2	Future Work	72
A	Coordinates Transformation	75
A.1	Homogeneous Coordinates Transformation	75
A.2	Attitude Representations	76
A.2.1	Euler Angle	76
A.2.2	Quaternion	78
A.2.3	Unit Quaternion	79
B	Uncertainty Calculus	83
B.1	Uncertainty Mapping	83
B.1.1	Linear Equations	83
B.1.2	Non-Linear Single Valued Equations	83
B.1.3	Non-Linear Vector Valued Equations	84
B.2	Least Square Solutions	85
B.2.1	Linear Equations	85
B.2.2	Non-Linear Equations	86
B.2.3	Homogeneous Equation	87
B.3	Kalman Filter	87
B.3.1	Continuous System Model	88
B.3.2	Linear Kalman Filter	88
	Bibliography	91
	Acronyms	101
	Declaration	103

Preface

The drive behind this research into a vision-aided navigation solution came from when I had simply been looking for a technology that provides positioning for an indoor photogrammetric application. For outdoor environments, it is common practice to use GPS-aided inertial navigation solutions. Surprisingly, however, there was no similar option for indoor environments. My target from the beginning was therefore to create a general navigation solution, which functioned without external references. Furthermore, only passive low-cost sensors should be used. Neither prior knowledge of the environment nor limitations on the movement are assumed in my investigations.

Although it seems straightforward to use stereo vision to aid in inertial navigation, I was certainly not aware of how many subjects need to be dealt with to finally present a solution. This research therefore was only possible thanks to the great teamwork and the commitment of everyone involved.

First of all, I would like to thank Anko Börner who gave me the opportunity to research the subject at the Institute of Optical Sensor Systems at the German Aerospace Center (DLR) in Berlin. I would also like to thank Professor Raúl Rojas from the Department of Mathematics and Computer Science at the Freie Universität Berlin for supporting and supervising my work. My special thanks go to Dirk Baumbach with whom I spent countless hours discussing the core filter algorithms and who has been a tremendous help over the years.

I would also like to thank the whole team, in particular Jürgen Wohlfeil for the great support on the software framework, David Krutz and Maximilian Buder working on the FPGA design, André Choinowski responsible for the hardware assembly, and Martin Bauer for his excellent work on the camera calibration method. Finally, I like to thank Sergey Zuev for managing the working group and all students and interns working on the details which made it run in the end.

Notation

Throughout this thesis the following notation is used:

- \mathbf{M} : Matrices are denoted in upper case bold letters.
- \mathbf{v} : Vectors are denoted in lower case bold letters.
- \mathbf{M}_n^b : Matrix transforming from n -frame to b -frame.
- \mathbf{v}^n : Vector given in n -frame coordinates.
- \mathbf{v}_{ab}^n : Quantity of the b -frame w.r.t the a -frame, given in n -frame coordinates.
- Operators:

$\tilde{[\cdot]}$	homogeneous vector
$\hat{[\cdot]}$	apparent quantity (estimated value, measured value)
$\dot{[\cdot]}$	temporal derivative
$\check{[\cdot]}$	augmented vector/matrix
$\ \cdot\ $	euclidean norm
$[\cdot]^{-1}$	matrix inverse
$[\cdot]^T$	matrix transposed
$[\cdot]^*$	quaternion conjugate
\circ	quaternion multiplication
\star	convolution
\times	cross product
$[\cdot]_{\times}$	skew-symmetric matrix from vector

Chapter 1

Introduction

Reliable information about *position* and *attitude*, also referred to as *pose* or *orientation*, is an essential requirement for many applications, especially those accomplishing navigation¹ tasks. Combined *dead reckoning* and *position fixing* is the preferred method for achieving reliable navigation solutions. Dead reckoning estimates the current position using a previous position fix and measurements of velocity, direction, and time traveled. Based on dead reckoning, an Inertial Navigation System (INS) calculates orientation from a known initial position, velocity, and attitude using inertial sensors – accelerometers and gyroscopes – which measure acceleration and angular velocity with respect to an inertial reference frame. Due to integrated errors, this leads to unbound error growth if not corrected by position fixes, which can be obtained by *line of position* measurements or radio navigation, e.g. Global Navigation Satellite System (GNSS).

If such fixes are unavailable or not accessible, vision can provide valuable information in terms of passive measurement to serve as a reference. Visual information is increasingly at navigation's disposal due to recent advances in image sensor technology, which enable affordable, lightweight, high resolution cameras. Therefore, I pay particular attention to a stereo vision system here, which can be used for egomotion estimation. My method leverages the complementary strengths of inertial and visual measurements by combining them, thereby reducing inertial sensor drift while supporting the visual system with inertial information.

¹ Navigation: The process of planning, recording, and controlling the movement of a craft or vehicle from one place to another. (Bowditch, 1802)

1.1 Motivation

A very strong indication for the efficacy of inertial measurements combined with vision is the fact that human evolution, and that of some animals, has developed inline with this strategy for perception and control of bodily movement. Figure 1.1 shows the inner ear and the vestibular system, while the latter consists of semicircular canals and the otolith organ (Gillingham and Previc, 1993). The three semicircular canals sense angular velocities and are arranged at mutually perpendicular angles. Linear acceleration and angular position (inclination) are sensed by the otoliths (utricle and saccule).

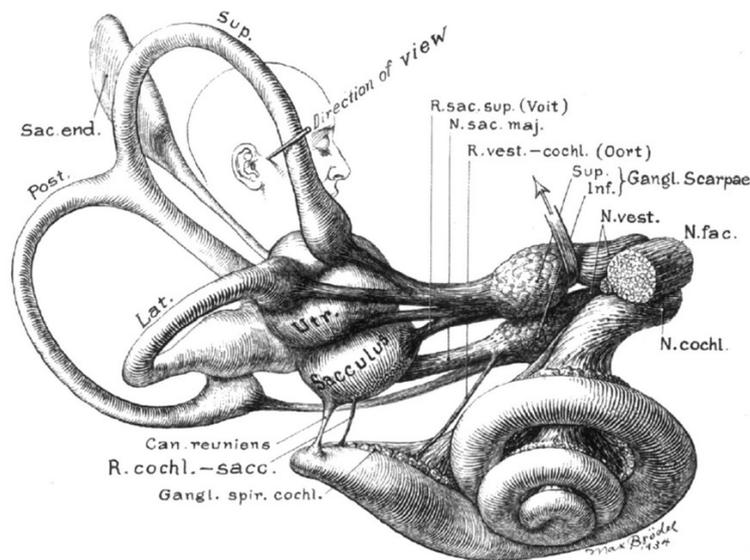


Figure 1.1: Vestibular system with lateral semicircular canal (Lat.), posterior semicircular canal (Post.), superior semicircular canal (Sup.), utricle (Utr.) and saccule. (Schubert and Minor, 2004)

Inertial sensors can be seen as an equivalent to the vestibular system. Humans use vision, reinforcing the inertial sensing system, for a reliable navigation. Without visual information or auditory aid, humans are unable to walk straight for longer distances; instead, they start walking in circles (Souman et al., 2009). If blindfolded, most people, when asked to cross a soccer field, will drift to a sideline before reaching the other end of the field. Bestaven et al. (2012) rules out any biomechanical asymmetries and supposes that this behavior is caused by the vestibular system. Similar to an inertial navigation system without aid,

accumulated noise causes erroneous navigation. On the other hand, the vestibular sensory organ helps stabilize vision when the head moves (Schubert and Minor, 2004). Therefore, combining very frequent inertial measurements – which are prone to large drift errors – with longer-term stable visual measurements is only a natural step.

Although my major objective is to achieve local navigation for indoor scenarios, I intend to advance a general solution as well. This requires the basic condition that no a priori knowledge about the environment can be assumed. It also implies a passive system that neither interferes with other systems nor modifies the environment. The incorporation of additional sensors must be supported by a flexible and open framework. This ensures reliability, robustness, and integrity for the navigation solution.

1.2 Contributions

My work put forth in this publication focuses on the aspect of navigation that is concerned with determining position and attitude. The main contribution I make is a low-cost Vision-Aided Inertial Navigation System (VINS), which provides local navigation in unknown environments. No a priori assumptions about the environments are made, e.g. no maps or pre-installed equipment are required.

- My method takes low-cost inertial sensors and couples them with stereo vision systems to take advantage of the complementary properties of both systems. The visual data aids the inertial navigation solution, while the inertial measurements are used to constrain the correspondence problem and track feature points simultaneously. Possible difficulties and solutions with respect to calibration and registration of the sensor systems are discussed below.
- I approach the matter from a systemic perspective, describing the complete processing chain for a tight integration of inertial measurements and visual data. Special emphasis is given to an overall error modeling. Reliable measures of uncertainty are derived throughout the data processing chain. I also show how the propagation of uncertainties for visual measurements can be greatly simplified.

- Furthermore, I have developed essential contributions for a novel calibration method for optical systems, which uses a holographic pattern. This method allows for accurate, fast, and reliable camera calibration. I also refer to an extension for stereo systems.
- To process a high level navigation solution from low level sensor data in real time, I present a scalable framework for a heterogeneous multi-sensor system with regard to hardware and software integration. The system is designed to be easily enhanced with either additional hardware or software.

1.3 Organization of the Thesis

This publication is organized as follows: Chapter 2 reviews state-of-the-art methods available in the field of navigation with an emphasis on visual and inertial navigation. I discuss possible solutions, including their assets and drawbacks.

Determining position and attitude by means of inertial navigation is recapitulated in chapter 3. I give an overview of inertial sensor technologies and their properties. Methods of sensor characterization and calibration are discussed based on a generalized sensor model.

Chapter 4 describes the overall process for extracting egomotion information from a stereo vision system. This includes a new method for intrinsic camera calibration with holographic patterning, camera registration, and egomotion determination as well as feature extraction and matching methods. Here, I show the advantage of deeply integrating visual and inertial measurements.

The combination of inertial and vision data leads to a VINS. Chapter 5 shows the registration of the sensor systems including synchronization issues. Solving these is an important prerequisite for fusing the data within a Kalman filter.

I propose a design for a heterogeneous multi-sensor system in chapter 6. It includes a hardware concept as well as a software framework for real time data handling and data processing. It is appropriate for calculating complex hierarchical data flows to produce high level information from low level sensor data. The capabilities of the proposed VINS are demonstrated with real data on a pedestrian indoor navigation task.

I draw conclusions and discuss directions for possible future work in chapter 7.

Chapter 2

Related Work

Navigation is defined as having two elements. The first is the determination of position and attitude, called *positioning* or *localization*; whereas, the second is mainly used in a topological sense. The second refers to *routing* and *guidance*. It deals with path planning and obstacle avoidance and leading an object to a given destination (Hofmann-Wellenhof et al., 2003). My work addresses the first element, which is concerned with positioning, particularly with methods appropriate for indoor applications.

Navigation has been the subject of research for many years which has produced a huge amount of contributions for a wide field of applications. They can be classified with the following categories:

- with or without external reference (infrastructure)
- map-based or map-less methods
- active or passive measurements

Although this publication focuses on positioning from passive measurements in non-cooperative indoor environments, I also give a general overview of different positioning methods and their possible shortcomings. For a comprehensive overview of these techniques, however, please refer to Mautz (2012). Inertial and visual navigation methods are reviewed here in greater detail.

2.1 General Navigation

Many solutions proposed for indoor navigation require infrastructure. Local area networks, which work with pseudolites (pseudo-satellites), RFID¹ Tags, UWB², or BLE³, have to be established first. Others make use of existing infrastructure, e.g. IEEE 802.11⁴ networks. Both may use fingerprint-based methods and need an offline procedure to create a probability radio map of the environment. Position is then determined by mapping the online measurement to this fingerprint (Kjærgaard, 2010). Dynamic environments require radio map updates and thus cause problems for positioning. Non-fingerprint-based solutions usually work with pseudolites or UWB systems and use triangulation for positioning. This method may suffer from occlusions and multi-path effects.

Another method is *map matching* in which a set of features, a map, determined by radar, vision, or other measurements is matched with a known map (Kayton and Fried, 1969). Elfes (1987) proposes using sonar range measurements for mapping and navigation. Laser range scanners that use a rotating laser beam to measure distances, have been applied in outdoor environments (Guivant et al., 2000) as well as for indoor scenarios (Fox et al., 1999). Clark and Dissanayake (1999) propose using millimeter wave radar for land vehicle navigation tasks. An active vision system is presented with the Kinect sensor from Microsoft (2011), which uses the PrimeSense 3D sensing technology (PrimeSense, 2010). A dense depth map of the environment is derived from infrared vision observing a coded diffractive infrared pattern that is projected onto the scene. Biswas and Veloso (2012) propose to extract planes from the derived 3D point cloud and map them to a known 2D representation of the environment to determine one's own position. Very recently, Google released the Project Tango (Google, 2014), a mobile-phone-based 3D reconstruction development kit which uses a PrimeSense-like technology optimized for low power consumption. The outdoor use of these active infrared solutions is very restricted due to interference from normal daylight. Generally, active systems are less scalable as they may interfere with similar signal sources, e.g. in case of multiple devices. Systems that either need infrastructure or use active measurements are not in the scope of this thesis, please refer to S. Chao (2009) and Torres-Solis et al. (2010) for further reading.

¹ Radio-Frequency Identification ² Ultra-Wideband ³ Bluetooth Low Energy ⁴ WLAN

Inertial navigation has been used for many years in aerospace and naval navigation (Kayton and Fried, 1969; Stovall, 1997; Titterton and Weston, 2004). A deep integration of GPS and strapdown-INS for airborne applications was demonstrated by Wendel (2003). Although this works very well, it shows major weaknesses in GNSS-denied environments with erroneous or no GNSS data, e.g. in urban or dense structured areas. Furthermore, this method does not work in indoor environments as needed for robotic applications or indoor navigation. With the development of small, lightweight, and low-cost Microelectromechanical System (MEMS) inertial sensors and their increasing performance (Schmidt, 2010), inertial navigation is finding ever greater robotic and consumer applications (Viéville and Faugeras, 1989). MEMS-based INS have to deal with difficult error characteristics, e.g. large bias instabilities, high noise, and g-sensitive gyroscopes resulting in poor navigation if left uncorrected (Walchko, 2002; Woodman, 2007). Redundant measurements have to be employed to restrain inertial sensor drift.

2.2 Visual Navigation

Increasing computational power and tremendous improvements in camera technology have opened the doors for vision-based navigation. To maintain an overview of the numerous contributions, DeSouza and Kak (2002) suggested a taxonomy which subdivides vision-based navigation into *map-based*, *map-building-based*, and *map-less* navigation. Solutions with single cameras – including omnidirectional cameras –, stereo cameras, or trifocal systems have been set up. Visual navigation techniques for mobile robots are reviewed by Bonin-Font et al. (2008).

Map-based Map-based navigation or map matching depends on a metric or topological map of the environment which has to be matched with an observed scene for localization. Map features can be represented as landmarks (Atiya and Hager, 1993) or occupancy grid cells (Moravec and Elfes, 1985; Cox, 1991). Dellaert et al. (1999) proposed using the condensation algorithm (conditional density propagation) also called Monte Carlo localization or particle filter for global self-localization. This probabilistic approach allows for non-Gaussian probability distributions and therefore provides multi-hypothesis tracking. A comprehensive review is found in Thrun et al. (2000) and Filliat and Meyer (2003).

Map-building-based If a map does not exist, the SLAM⁵ problem can be addressed (Durrant-Whyte and Bailey, 2006). Besides of using laser-range scanners, sonar, or radar sensors, visual SLAM systems have been set up with single cameras (Burschka and Hager, 2004; Karlsson et al., 2005; Davison et al., 2007) and stereo camera systems (Sim and Little, 2006; Konolige et al., 2010). Although the SLAM technology can be considered mature, there are still unsolved problems, e.g. regarding computational complexity or the handling of dynamic scenes. For a good overview of SLAM and possible problems, refer to Thrun (2002) and Aulinas et al. (2008).

Map-less Map-less navigation neither needs an a priori description of the environment nor builds a map for navigation purposes. The main techniques used are *optical flow*, *appearance-based* navigation, and *feature-based* tracking.

With optical flow, single pixels are tracked in consecutive images (Horn and Schunck, 1981). Inspired by insects' flight, the resulting optical flow field is used for reactive navigation strategies, e.g. for obstacle detection or to achieve corridor centering (Coombs and Roberts, 1993). An overview of insect-inspired navigation strategies is found in Srinivasan et al. (2004). The application of optical flow for robotic navigation is reviewed in H. Chao et al. (2014).

The appearance-based method usually consists of an offline learning phase and an online navigation phase. Matsumoto et al. (1996) store an image view sequence in a trial run. To achieve position and attitude, the current image is matched to the image sequence with a template matching technique. Cummins and Newman (2007) proposed extracting an observation model, e.g. SIFT⁶ descriptors from the learning images. A Bayesian network is later used to estimate the location, including the detection of loop closures. The additional learning phase of this method is a disadvantage.

Feature-based methods extract distinctive points or lines from an image and track them in consecutive images. In similarity to the term wheel odometry, the egomotion determination from these features is called Visual Odometry (VO) (Nistér et al., 2004). Single view approaches apply the eight-point algorithm (Longuet-Higgins, 1981) or the five-point algorithm (Nistér, 2004) to solve for the essential matrix (eq. (4.2.2)), which can be decomposed to recover the relative

⁵ Simultaneous Localization and Mapping ⁶ Scale-Invariant Feature Transform

pose up to a scale factor. The absolute scale factor has to be known initially or estimated from additional measurements, e.g. by a laser altimeter (E. Johnson and Mathies, 1999).

Using stereo vision avoids scale ambiguity and allows for extraction of the complete egomotion. The work of Moravec (1980) is one of the first describing a typical workflow, including feature extraction, stereo matching, triangulation, feature tracking, and egomotion estimation. This approach was extended with advanced modeling of triangulation errors by Matthies and Shafer (1987). Olson et al. (2003) included a compass for periodical orientation updates to achieve linear error growth in the accumulated position. Feature tracking, finding a homologous point set in two images, has to deal with the correspondence problem caused by potentially unknown and unrestricted egomotion. To solve it in real time, the possible search areas have to be constrained, either by restricting the motion or predicting it with additional information. Another way is to use multi-resolution pyramids for a more efficient tracking.

Hirschmüller (2003) and Nistér (2004) proposed to use feature matching rather than feature tracking. Features are extracted in every image and only matches between these features are allowed. Furthermore, an outlier detection scheme was included to increase the robustness of the algorithm. An adoption is shown by Howard (2008), which operates directly on dense disparity images. Assuming a constant velocity motion model, Kitt et al. (2010) applied a Kalman filter for further accuracy improvements.

2.3 Vision-Aided Inertial Navigation

Besides the advantages in the field of visual navigation in the last decade, considerable efforts have been made in the development of inertial sensor technology (Schmidt, 2010). The potential benefits of combining inertial and visual information was already recognized by Viéville and Faugeras (1990). The authors proposed using the two complementary sensing modalities to achieve a more accurate and robust navigation. In general, systems can be distinguished between loosely coupled ones where the visual module works independently and tightly coupled systems in which the inertial measurements are used to support the tracking process and vice versa.

Single Camera View You et al. (1999) proposed a tight coupling of gyroscopes and vision for augmented reality. The gyroscope data was used to predict feature positions in order to constrain the correspondence problem. The result of the optical-flow-based tracking module is used to correct for accumulated drift in the inertial data. Huster (2003) demonstrated a system using gyroscopes, acceleration sensors, and bearing observation from a single feature for underwater vehicle navigation. A more complex approach, fusing inertial sensors, a laser altimeter, and visual information from multiple features for a planetary lander was shown in Roumeliotis, A. Johnson, et al. (2002). Some authors proposed using visual information to bridge GPS drop-outs in a GPS-aided INS for airborne and terrestrial vehicle applications (A. Brown and Sullivan, 2002; George and Sukkarieh, 2007; Randeniya et al., 2010). Another single view approach is using inertial sensors and omnidirectional vision with optical flow (Stratmann and Solda, 2004) or feature tracking methods (Strelow, 2004; Diel, 2005).

Stereo Camera View The study of human vision and vestibular system reveals an analogy to stereo vision and inertial sensors (Lobo, 2002). Inertial clues can be used as a vertical reference for ground plane estimation and vertical line detection or to estimate the focal distance of the camera (Lobo, 2002; Lobo and Dias, 2003, 2004). Based on this work, Corke et al. (2007) proposed a framework for Structure from Motion (SfM) aided by inertial information. Zhu et al. (2006) integrated a loosely coupled system with GPS, Inertial Measurement Unit (IMU), and stereo vision fused within an extended Kalman filter for outdoor applications. Oskiper et al. (2007) employed an additional rearview stereo system in order to increase the robustness of the visual measurement for indoor environments. To bridge failing visual measurements, a loosely coupled IMU is used. A tightly coupled SfM system used for indoor environments is demonstrated in Veth and Raquet (2007) and Miller et al. (2010). In this approach a few feature points are selected, incorporated to the filter, and tracked as long as they are visible. Their positions and uncertainties are projected onto the image to constrain the search area.

Chapter 3

Inertial Navigation

Based on dead reckoning, inertial navigation is orientation by means of known initial position, velocity, and attitude. It uses inertial sensors – accelerometers and gyroscopes –, which measure acceleration and angular velocity with respect to (w.r.t.) an inertial reference frame. In an inertial frame, inert objects move rectilinearly with constant velocity through space. According to Newton's First Law an external force is needed for a change of motion. Newton's Second Law states that this force is proportional to the acceleration.

The main advantage of inertial sensors and thus inertial navigation is that they are self-contained, robust, and independent from external signals and disturbance. On the other hand, dead reckoning accumulates small errors in the inertial measurements, leading to unbound error growth. This chapter looks at a simplified navigation approach for short-term, local navigation that is suitable for indoor environments. For a more detailed reading refer to Titterton and Weston (2004), which served as a basis for this chapter.

3.1 Inertial Navigation System

An INS consists of accelerometers, gyroscopes, and a computer for continuous calculation of position and attitude. Three accelerometers, and respectively three gyroscopes, are mounted with their sensitive axes at mutually perpendicular angles within an IMU. Accelerometers measure the specific force to which acceleration and the gravitational force contribute. Changes in velocity and position are calculated by integrating the measured accelerations w.r.t. time. These changes

are added to the previously known position to obtain the present position.

Rotational motion of the accelerometers w.r.t. the inertial reference frame have to be sensed in order to resolve the accelerometer measurements in terms of the reference frame for integration. Gyroscopes measure the angular velocity or rotation rate. Integrating this measurement w.r.t. time gives the change in attitude, which is added to the previously known attitude to determine the present attitude.

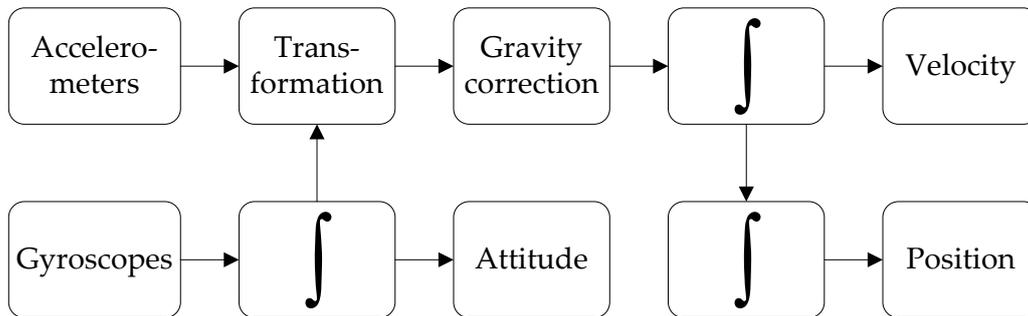


Figure 3.1: Schematic strapdown INS

An INS rigidly attached to the measured object is called a strapdown-INS. In comparison to platform-stabilized systems, these are not isolated from rotational movements, which provides lower mechanical complexity, but increases the computational effort. An additional problem is the need for gyroscopes that measure the full dynamic range of the body rotation with the same accuracy as platform-stabilized systems. With small, lightweight, and low-cost MEMS inertial sensors, navigation systems become possible for a wide range of applications.

3.1.1 Coordinate Frames

The definition of different reference frames is essential for the derivation of inertial navigation equations. Subscripts, e.g. for inertial sensors measuring the b -frame w.r.t the i -frame, have been omitted for a better readability in the following.

Earth-centered Inertial Frame (i -frame) This coordinate frame has its origin at the center of the Earth, with non-rotating axes w.r.t. the fixed stars. The z -axis is the rotational axis of the Earth which is assumed to be invariant.

Earth-centered Fixed Frame (*e*-frame) Similar to the inertial frame, it is earth-centered but has fixed axes w.r.t. the earth. The x -axis points to the intersection of the Greenwich meridian and the Earth's equatorial plane.

Navigation Frame (*n*-frame) This local coordinate system has an arbitrary origin and its axes are aligned north, east, and down (NED), also referred to as Local Tangent Plane (LTP). An alternative convention uses axes aligned to east, north, and up (ENU).

Body Frame (*b*-frame) This coordinate frame is aligned to the roll, pitch, and yaw axes of the measured device.

3.1.2 Attitude Algorithm

The relative rotation between two coordinate frames can be described in various ways (see appendix A). Using the quaternion representation for the attitude update algorithm has some advantages. It is more precise, easier to check for consistency and computationally less demanding. Small numerical errors induced by the update process are corrected by normalizing the quaternion. For the quaternion update with an angular velocity ω , the following differential equation has to be solved:

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \circ \mathbf{p}, \quad (3.1.1)$$

with $\mathbf{p} = (0, \omega^T)$. The quaternion multiplication is expressed in matrix form as

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{W} \mathbf{q}, \quad (3.1.2)$$

with

$$\mathbf{W} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \quad (3.1.3)$$

Assuming the angular velocity to be constant over the update interval, the equation (3.1.2) is solved by using the exponential, where $y(t) = C \exp(kt)$ is a solution to the differential equation $dy/dt = ky$.

$$\mathbf{q}_{k+1} = \exp\left(\frac{1}{2} \mathbf{W} \Delta t\right) \mathbf{q}_k \quad (3.1.4)$$

Expanding the exponential term and writing it in quaternion form again leads to

$$\mathbf{q}_{k+1} = \mathbf{q}_k \circ \mathbf{r}_k, \quad (3.1.5)$$

with \mathbf{r}_k forming a quaternion for a rotation about a vector $\boldsymbol{\sigma} = \boldsymbol{\omega}\Delta t$, of magnitude $\sigma = |\boldsymbol{\sigma}|$ with

$$\mathbf{r}_k = \begin{pmatrix} a_c \\ a_s \boldsymbol{\sigma} \end{pmatrix}, \quad (3.1.6)$$

where

$$a_c = \cos\left(\frac{\sigma}{2}\right) = 1 - \frac{(0.5\sigma)^2}{2!} + \frac{(0.5\sigma)^4}{4!} - \dots \quad (3.1.7)$$

$$a_s = \frac{\sin(\sigma/2)}{\sigma} = 0.5 \left(1 - \frac{(0.5\sigma)^2}{3!} + \frac{(0.5\sigma)^4}{5!} - \dots \right). \quad (3.1.8)$$

3.1.3 Position Algorithm

According to fig. 3.1, the position is obtained by integrating the velocity which itself is determined from accelerations \mathbf{a}^b . For an observer moving in the vicinity of earth the following differential equation (Titterton and Weston, 2004) has to be solved:

$$\dot{\mathbf{v}}^n = \mathbf{R}_b^n \mathbf{a}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_{eb}^n + \mathbf{g}^n. \quad (3.1.9)$$

It includes a correction for the Coriolis acceleration with $\boldsymbol{\omega}_{ie}^n$, the angular velocity of the earth frame w.r.t the inertial frame and the transport rate $\boldsymbol{\omega}_{en}^n$, the angular velocity of the navigation frame w.r.t the earth frame. Both can only be determined if the position of the navigation frame w.r.t the earth frame and the direction north is known. In case the navigation task is restricted to a short period within a local area, they can be neglected, especially when using low-cost MEMS inertial sensors with high noise levels.

An incremental velocity is derived by integrating the measured acceleration for the interval t_k to t_{k+1} with

$$\Delta \mathbf{v}^n = \int_{t_k}^{t_{k+1}} \mathbf{R}_b^n \mathbf{a}^b dt. \quad (3.1.10)$$

As the rotation matrix varies continuously within the update interval, the integral is approximated as follows:

$$\Delta \mathbf{v}^n \approx \mathbf{R}_{b,k}^n \left(\Delta \mathbf{v}_{k+1}^b + \frac{1}{2} \boldsymbol{\alpha}_{k+1}^b \times \Delta \mathbf{v}_{k+1}^b + \frac{1}{2} \int_k^{k+1} \left(\Delta \boldsymbol{\alpha}^b \times \mathbf{a}^b - \boldsymbol{\omega}^b \times \Delta \mathbf{v}^b \right) dt \right) \quad (3.1.11)$$

with

$$\Delta \mathbf{v}^b = \int_{t_k}^t \mathbf{a}^b dt \quad (3.1.12)$$

$$\Delta \boldsymbol{\alpha}^b = \int_{t_k}^t \boldsymbol{\omega}^b dt, \quad (3.1.13)$$

where $\Delta \mathbf{v}_{k+1}^b = \Delta \mathbf{v}^b$ and $\Delta \boldsymbol{\alpha}_{k+1}^b = \Delta \boldsymbol{\alpha}^b$ evaluated over the interval t_k to t_{k+1} . It contains the velocity change measured by the IMU, a rotation correction term and a dynamic integral term. The dynamic term equals zero if \mathbf{a}^b and $\boldsymbol{\omega}^b$ remain constant over the update interval. A detailed derivation of eq. (3.1.11) can be found in Titterton and Weston (2004). Using the simplified eq. (3.1.9), the velocity vector is updated as follows:

$$\mathbf{v}_{k+1}^n = \mathbf{v}_k^n + \Delta \mathbf{v}^n + \mathbf{g} \Delta t \quad (3.1.14)$$

$$= \mathbf{v}_k^n + \mathbf{R}_{b,k}^n \left(\mathbf{a}_{k+1}^b \Delta t + \frac{1}{2} \boldsymbol{\omega}_{k+1}^b \Delta t \times \mathbf{a}_{k+1}^b \Delta t \right) + \mathbf{g} \Delta t. \quad (3.1.15)$$

The position within the navigation frame can now be obtained by integrating the velocity w.r.t the navigation frame.

$$\mathbf{p}^n = \int_0^t \mathbf{v}^n dt \quad (3.1.16)$$

For the interval t_k to t_{k+1} , the integral is approximated with the trapezoidal rule.

$$\mathbf{p}_{k+1}^n = \mathbf{p}_k^n + \frac{\mathbf{v}_k^n + \mathbf{v}_{k+1}^n}{2} \Delta t \quad (3.1.17)$$

3.2 Inertial Measurement Unit

An IMU consists of three gyroscopes and three accelerometers, each set with their sensitive axes placed at mutually perpendicular angles. Recently, sets

of magnetometers and a barometer have been added to boost performance of attitude and height calculation. All measurements are synchronized and preprocessed within the IMU and provided via a common interface to the user. The manufacturer usually provides a calibration sheet quantifying the error characteristic of the build in sensors. For the experimental part, the tactical grade MEMS-IMU *ADIS-16488* from Analog Devices Inc. is used here. A detailed characterization is given in tables 3.1 and 3.2.

3.2.1 Sensor Error Model

All inertial sensor data are measurements and therefore subject to errors. Measurements of both sensor sets show deterministic and stochastic errors (IEEE, 2004, R2010). The measurement \hat{x} is modeled with

$$\hat{x} = SMx + b_x + n, \quad (3.2.1)$$

where the bias term b_x is added to the true value x . Zero-mean Gaussian noise is subsumed in n . The linear error in the ratio between input and output signal, called scale factor error, is described by the matrix S with

$$S = I_{3 \times 3} + \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} 10^{-6}, \quad (3.2.2)$$

where $s_{(\cdot)}$ are the scale factor errors of the particular axes, given in parts per million (ppm). Derivations from the linear error approximation are resumed as a scale factor non-linearity. The misalignment matrix M corrects for non-orthogonality of the sensor axes, given as small angle approximation from eq. (A.2.1) with

$$M = \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix}. \quad (3.2.3)$$

Defining the output coordinate system, so that the z -axis coincides with the sensor z -axis and the output y -axis lies in the plane spanned by the sensor y -axis

and z-axis, this can be simplified according to Grewal et al. (1990) with

$$M = \begin{bmatrix} 1 & -\psi & \theta \\ 0 & 1 & -\phi \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.2.4)$$

Furthermore, a constant bias term x_b is added. It is the average over a specified time of signal output, measured at specified operating conditions that have no correlation with input rotation or acceleration (IEEE, 2001).

So far, the model only includes fixed error components obtained by a calibration procedure. It does not take into account temperature dependent variations, switch-on to switch-on variations, and in-run variations. They all influence the navigation system's accuracy. In fact, they define the performance of the system. Especially in-run variations of bias and scale factor, called bias stability and scale factor stability, are limiting factors and therefore used to classify inertial sensors (see fig. 3.3 and fig. 3.6). Compensation mechanisms are shown in chapter 5.

Allan Variance

To determine the character of a random process, the Allan variance method is used as a measure of frequency stability. A sequence of stationary sampled data is divided in n bins with time τ and averaged. The Allen variance is calculated similar to the common variance computation:

$$\sigma^2(\tau) = \frac{1}{2(n-1)} \sum_{i=1}^n \left(y(\tau)_i - y(\tau)_{i-1} \right)^2. \quad (3.2.5)$$

Doing this with increasing averaging times, a log-log Allan deviation plot, showing the square root of the Allan variance, can be produced.

Figure 3.2 shows an example plot for a fiber optic gyroscope with different noise terms defined by characteristic slopes. For short averaging times the white sensor noise, often referred to as random walk, dominates. It can be read directly from the log-log plot at $\tau = 1$. With increasing averaging time the variance decreases but increases again at some point. The minimum point on the curve is defined to be the bias instability. A more detailed explanation is found in IEEE (1997, R2008).

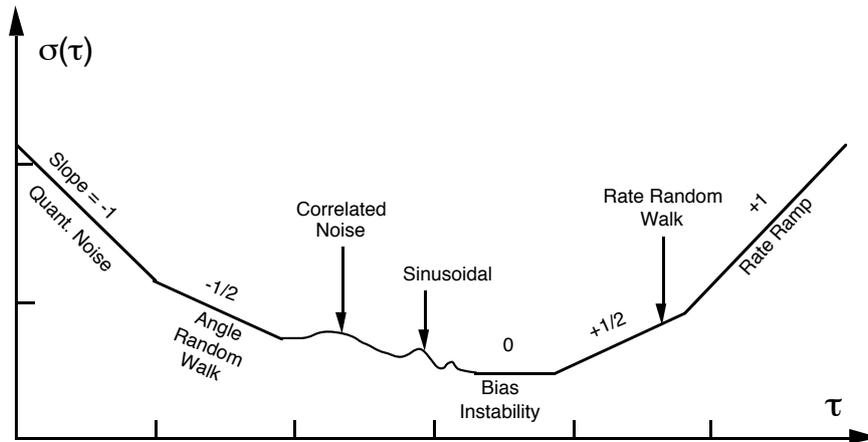


Figure 3.2: Allan deviation sample plot (IEEE, 1997, R2008)

3.2.2 Gyroscope

For the measurement of angular rates w.r.t. an inertial frame, various sensor types with different measurement principles are used. Although MEMS gyroscopes are central for consumer applications, Ring Laser Gyroscopes (RLG) and Interferometric Fiber Optic Gyroscopes (IFOG) are very important technologies, still used for high accuracy applications.

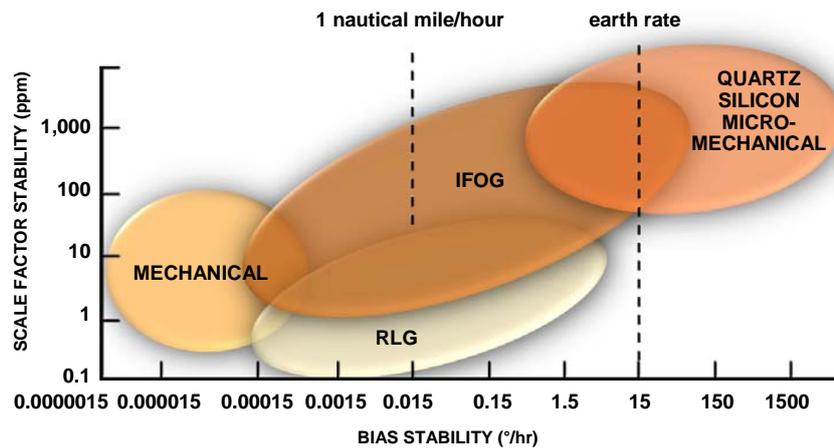


Figure 3.3: Gyroscopes technologies (Schmidt, 2010)

The RLG contains a laser source and three or more mirrors to form an optical path that functions as an optical oscillator. Two independent laser beams travel clockwise and anti-clockwise on this path. An induced rotation perpendicular to

the optical path changes the optical path's length, which is called Sagnac effect. This also changes the frequency of each beam, resulting in a frequency difference, proportional to the angular displacement induced.

IFOGs were developed as a less expensive alternative to RLGs. They also use the Sagnac effect but measure the phase difference of two beams traveling in opposite directions along an optical path. The optical path is defined by a wound coil of optical fiber to increase the length of the path. A more detailed description of both technologies can be found in Titterton and Weston (2004).

MEMS gyroscopes work by detecting the Coriolis force acting on a vibrating proof mass m when rotated with angular rate ω perpendicular to the axis of vibratory motion v .

$$F_c = -2m(\omega \times v) \quad (3.2.6)$$

The mass is pushed out of the oscillation plane, which is detected by capacitor plates in silicon sensors or by piezoelectric sensing element in Quartz Rate Sensors (QRS). Figure 3.4 shows a realization of a balanced oscillator or tuning fork gyroscopes.

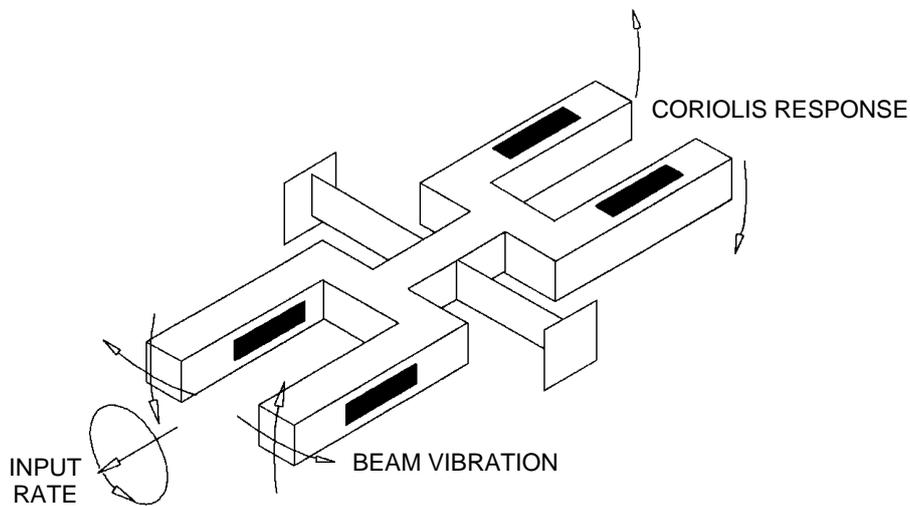


Figure 3.4: Tuning fork principle (IEEE, 2004, R2010)

Although performance has improved continually in recent years (Perlmutter and Robin, 2012), MEMS inertial gyroscopes still show significantly higher amounts of noise as well as a less stable bias and scale factor compared to optical gyroscopes (see fig. 3.3). Zero mean random noise is integrated to angle random walk, expressed in $^{\circ}/\sqrt{h}$. Furthermore, the bias from eq. (3.2.1) causes the angular

error to grow linear with time, if not compensated for. It consists of a constant and a random component that varies from switch-on to switch-on, but then remains constant. Due to flicker noise in the electronics the bias wanders over time. This in-run bias stability, given in °/h is specified by means of the Allan variance. It is a non-stationary process characterized by a $1/f$ power spectral density (IEEE, 2001).

Unlike optical gyroscopes, MEMS devices show a significant error when exerted on by acceleration. This is caused by mechanical design asymmetry and micro-machining inaccuracies (Weinberg, 2011). The g -sensitivity is a linearly dependent error expressed in °/s/g . Equation (3.2.1) has to be extended with $\Delta\omega_b = \mathbf{G}\mathbf{a}_b$, where \mathbf{G} is a 3×3 g -sensitivity matrix and \mathbf{a}_b is the applied acceleration. Some IMUs compensate for that error on-line with their built-in accelerometers. Bancroft and Lachapelle (2012) proposed to estimate the sensitivity matrix dynamically.

The IMU *ADIS-16488* is used for the experimental setup. It consists of silicon gyroscopes with capacitive sensing elements (Zumbahlen, 2005). The measured Allen deviation shown in fig. 3.5 verifies that all three axes lay within the given specification (table 3.1).

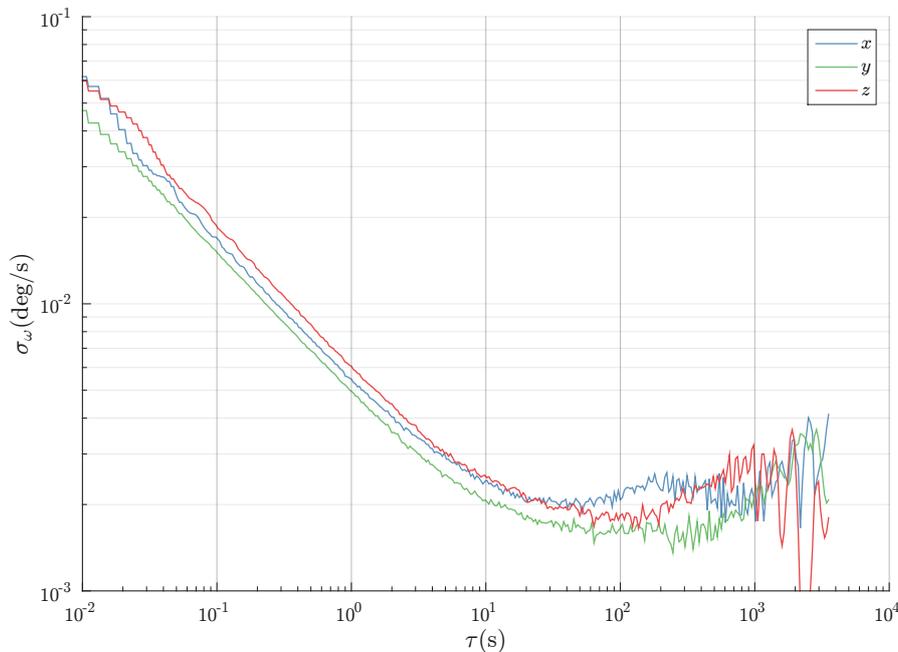


Figure 3.5: *ADIS-16488* gyroscope Allan deviation

	ADIS-16488
Range [°/ s]	450
Bandwidth [Hz]	330
Angle random walk [°/√h]	0.3
Bias stability [°/ h]	6.25
Scale-factor stability [ppm]	10 000
g-Sensitivity [°/s/g]	0.009

Table 3.1: Gyroscope specifications

3.2.3 Accelerometer

Inertial navigation mainly relies upon integrating acceleration measurements to obtain velocity and position changes. An acceleration a is measured indirectly by measuring the force F acting on a proof mass m with

$$F = ma \quad (3.2.7)$$

and

$$a = f + g, \quad (3.2.8)$$

where g is the gravitational acceleration and f a non-gravitational component produced by the specific force applied to the sensor. A good knowledge of the gravitational field is therefore essential for the use of inertial measurement.

Mechanical accelerometers usually work as mass-spring devices. In open-loop systems, the deflection of the spring, when exerted on by acceleration, is a measure of the applied force. Closed-loop or torque-balanced devices, hold the proof mass in null position by torque, which is generated proportional to the displacement. The current for producing that torque is a measure of the applied force. This method is preferred to open-loop devices as it is far more accurate.

Solid state MEMS devices measure acceleration primarily in two ways. Silicon micro-fabricated pendulous-mass accelerometers detect the displacement of a hinged proof mass by a change in capacitive, piezoelectric or piezoresistive readouts. They can be operated in open-loop or closed-loop mode. Vibratory accelerometers make use of the resonant frequency modulation of a vibrating element when put under tension or compression. A common implementation

is the dual beam mechanization in quartz crystal technology as described in Albert (1994). Without external force, both beams vibrate at the same frequency. When acceleration is applied, one beam becomes compressed and its frequency decreased, the other is stretched increasing its frequency. The difference in frequency is proportional to the applied acceleration.

Driven by automotive industry, the quality of MEMS acceleration sensors has reached a level sufficient for lower-grade tactical applications as well as commercial applications (Schmidt, 2010). Mechanical acceleration sensors are still used if high performance is needed, e.g. for military applications (see fig. 3.6).

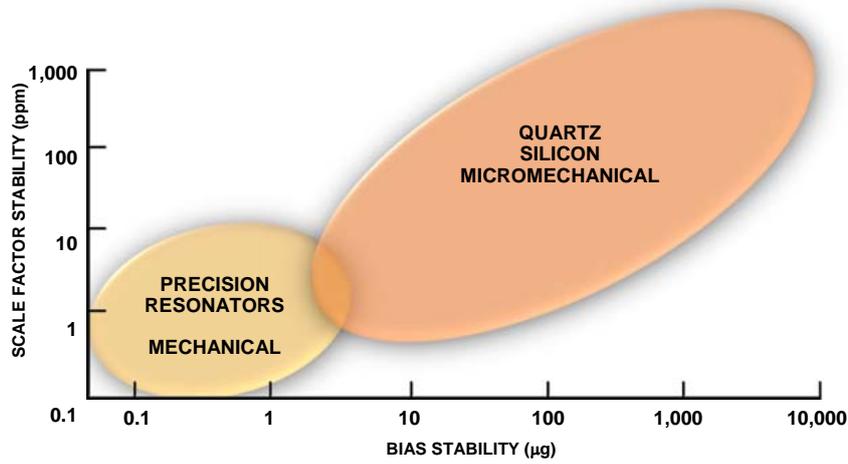


Figure 3.6: Accelerometer technologies (Schmidt, 2010)

Zero mean random noise is specified as velocity random walk, given in $m/s/\sqrt{h}$ or equivalent in mg/\sqrt{Hz} . If integrated twice, it creates a second order position random walk with zero mean and standard deviation

$$\sigma_s(t) \approx \sigma_a t^{3/2} \sqrt{\frac{\Delta t}{3}}, \quad (3.2.9)$$

with sample time Δt (Woodman, 2007). A constant bias causes the position error to grow quadratically with time.

$$\Delta s(t) = a_b \frac{t^2}{2} \quad (3.2.10)$$

Similar to gyroscopic measurement, the bias stability produces a second order velocity random walk resulting in a third order position random walk that grows

proportional to $t^{5/2}$ (Woodman, 2007). An overview of different accelerometers types and their error characterization can be found in IEEE (1998, R2008).

The used *ADIS-16488* provides silicon MEMS capacitive accelerometers, specified in table 3.2. Again, measured Allan deviation confirms the given specification. Figure 3.7 shows that the *y*-axis accelerometer has a significant increased bias stability. Magnetometer and barometer measurements are also available but not used in the current setup.

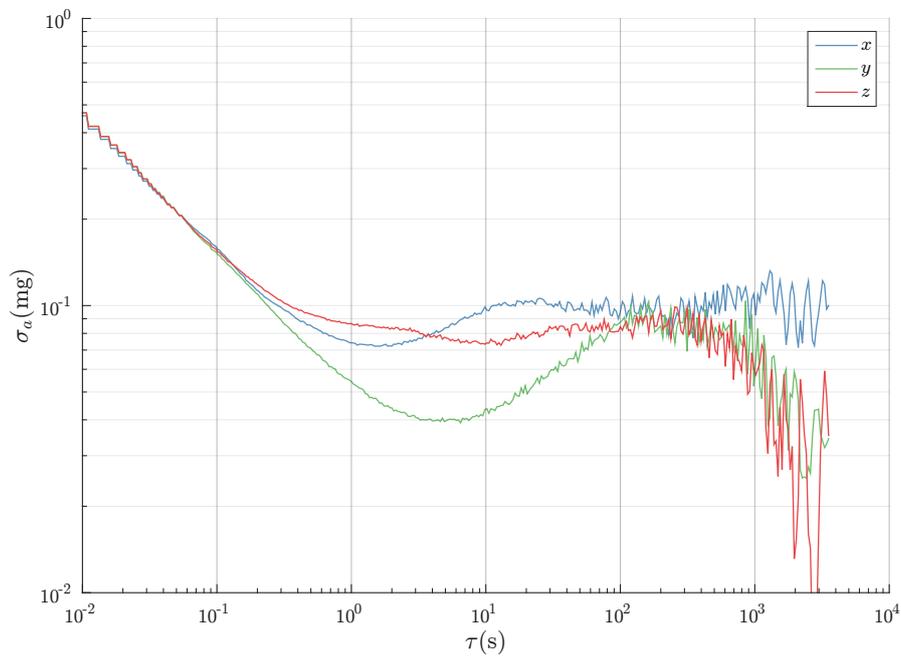


Figure 3.7: *ADIS-16488* accelerometer Allan deviation

	<i>ADIS-16488</i>
Range [g]	± 18
Bandwidth [Hz]	330
Velocity random walk [m/s/ \sqrt{h}]	0.029
Bias stability [mg]	0.1
Scale-factor stability [ppm]	5000

Table 3.2: Accelerometer specifications

3.3 Inclination Sensor

Acceleration sensors used in inclination devices are realized with a force balance (closed-loop) design. A dampened magnetic permeability mass is held in null position with a counter force provided by a magnetic coil. The coil current is a measure for the force acting on the mass. This design provides an excellent bias stability and a high sensitivity with a limited input range. It is also very stable against changing thermal conditions. As it aims to measure static accelerations, the output is filtered to remove any picked up high frequent content.

In case no specific force is applied, only the gravitational acceleration g is measured. An inclinometer, with two perpendicular accelerometer axes, measures the inclination angles $\alpha = \arcsin(a_x/g)$ and $\beta = \arcsin(a_y/g)$ w.r.t the local tangent plane of the earth. This can be expressed as a rotation \mathbf{R}_n^i transforming from the navigation frame to the inclination sensor frame. According to appendix A.2, Euler angles in the representation for body-fixed axes are extracted as follows:

$$\frac{1}{g} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \mathbf{R}_n^i \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \sin(\theta) \\ -\sin(\phi) \cos(\theta) \\ \cos(\phi) \cos(\theta) \end{pmatrix}. \quad (3.3.1)$$

This leads to:

$$\phi = -\arctan\left(\frac{a_y}{a_z}\right) \quad (3.3.2)$$

$$\theta = \alpha = \arcsin\left(\frac{a_x}{g}\right), \quad (3.3.3)$$

with a virtual z-axis component $a_z = \pm\sqrt{g^2 - a_x^2 - a_y^2}$. The sign depends on the mounting orientation of the inclinometer. The dual-axis inclinometer *ADIS-16209* from Analog Devices Inc. was used in the experimental sensor head.

	ADIS-16209
Range [°]	±90
Relative accuracy [°]	±0.1
Sensitivity [°/LSB]	0.025

Table 3.3: Inclinometer specification

Chapter 4

Computer Vision

In this chapter, I describe how to extract motion information from stereo vision. Measuring with images requires having an exact knowledge of the geometrical properties of the camera system. I introduce a novel calibration method using diffractive optical elements for gaining information of the camera model and lens distortion. Furthermore the registration of the stereo camera system is needed as a prerequisite step for the reconstruction of 3D image points used for pose estimation.

Throughout the complete processing chain the uncertainties of calibration, registration, triangulation, and pose estimation are reviewed. This is essential to derive an accuracy measure for the estimated egomotion.

4.1 Single Camera View

Looking at an image, every pixel represents a ray with a distinct direction, passing through the center of projection, eventually meeting some 3D object point. A single camera image therefore shows a 2D projection of the world without the depth information, which can not be retrieved again without additional information. In the following, the world is modeled in projective space \mathbb{P}^n , equal to Euclidean space \mathbb{R}^n along with points at infinity. A Euclidean object point $\mathbf{M} = (X, Y, Z)^T \in \mathbb{R}^3$ is written as homogeneous coordinate $\tilde{\mathbf{M}} = (X, Y, Z, 1)^T \in \mathbb{P}^3$. Because $(X, Y, Z, 1)^T \sim (X \cdot W, Y \cdot W, Z \cdot W, W)^T$, it is in fact defining a ray with an origin at $(0, 0, 0, 1)^T$. Points at infinity are stated as $(X, Y, Z, 0)^T$. An overview regarding projective space is found in Mohr and Triggs (1996).

4.1.1 Camera Model

The pinhole model describes the central projection of an object point in world coordinates \tilde{M}^w to image coordinates $\tilde{m}^i = (u, v, 1)^T \in \mathbb{P}^2$ on a image plane π . Starting at the virtual projection center c , the image plane is defined parallel to the xy -plane of the camera coordinate frame as shown in fig. 4.1.

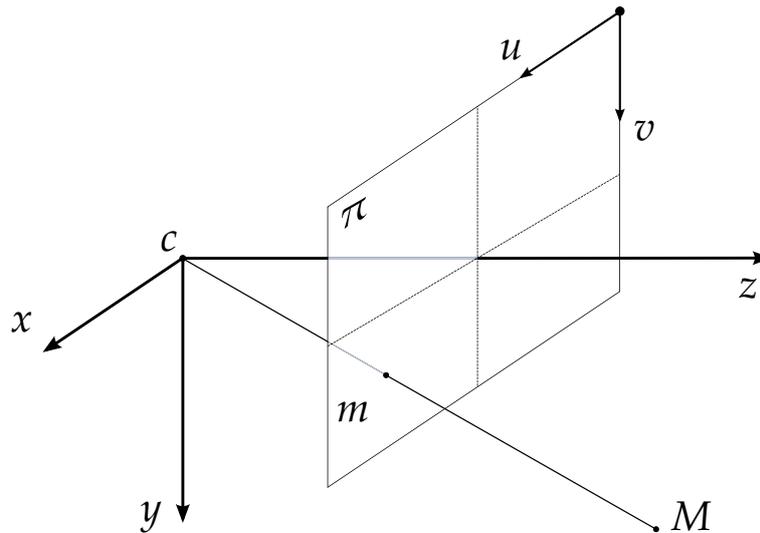


Figure 4.1: Camera coordinate frame

In the following, the successive transformation of a 3D object point to a 2D image point and vice versa is shown. Every transformation step, except the projection, can introduce errors which increases the overall uncertainty.



Figure 4.2: Coordinated frames and transformations

World Frame \leftrightarrow Camera Frame An object point is brought from world frame to camera frame with a homogeneous transformation matrix T_w^c , containing the parameters of exterior orientation. The inverse transformation matrix reverses this operation. See appendix A.1 for a detailed description.

$$\tilde{M}^c = T_w^c \tilde{M}^w \quad (4.1.1)$$

Camera Frame \leftrightarrow Normalized Camera Frame A projection of the object coordinate $\tilde{\mathbf{M}}^c = (X, Y, Z, W)^T$ using the normalized projection matrix $\mathbf{P} = [\mathbf{I}_3 | 0]$ gives normalized camera coordinates $\tilde{\mathbf{m}}^c$.

$$\tilde{\mathbf{m}}^c = \mathbf{P}\tilde{\mathbf{M}}^c \quad (4.1.2)$$

The transformation from homogeneous coordinates to Cartesian coordinates finally projects to a virtual image plane at $z = 1$.

$$\mathbf{m}^c = (x, y)^T = (X/Z, Y/Z)^T \quad (4.1.3)$$

Non-linear error propagation from eq. (B.1.11) maps the object point uncertainty $\Sigma_{\tilde{\mathbf{M}}^c}$ to the uncertainty Σ_{xy} for the normalized camera coordinates. It uses the partial derivation of the image point w.r.t. the homogeneous object point stated in the Jacobian

$$\mathbf{J} = \frac{\partial \mathbf{m}^c}{\partial \tilde{\mathbf{M}}^c} = \frac{1}{Z^2} \begin{bmatrix} Z & 0 & -X & 0 \\ 0 & Z & -Y & 0 \end{bmatrix}. \quad (4.1.4)$$

The reverse transformation requires the triangulation of two normalized camera coordinates, which is described in section 4.2.2.

As can be seen in the following, core algorithms like distortion correction, triangulation, or pose estimation are calculated with normalized coordinates. This is very convenient because they are independent from exterior and interior camera orientation, which greatly simplifies uncertainty calculations.

Normalized Camera Frame \leftrightarrow Image Frame By applying the camera matrix \mathbf{K} , containing the parameters of interior orientation, ideal image coordinates $\tilde{\mathbf{m}}^i = (u, v, 1)^T$ are obtained.

$$\tilde{\mathbf{m}}^i = \mathbf{K}\tilde{\mathbf{m}}^c, \quad (4.1.5)$$

with

$$\mathbf{K} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.1.6)$$

where $\alpha = f/d$ is expressed in pixels, with the camera constant f and pixel size d . With the lens focused at infinity, the camera constant is equal to the focal length, otherwise it is slightly less. The principal point $(u_0, v_0)^T$ is defined to be the point

on the image plane that is at the base of the perpendicular of the center of the lens (T. Clarke and Fryer, 1998).

The uncertainty Σ_{uv} of an image point depends on the uncertainty Σ_{xy} of the normalized camera coordinate as well as from errors within the interior orientation $\Sigma_{io} = \text{diag}(\sigma_\alpha^2, \sigma_{u_0}^2, \sigma_{v_0}^2)$. From eq. (B.1.11) follows

$$\Sigma_{uv} = J \begin{bmatrix} \Sigma_{xy} & \\ & \Sigma_{io} \end{bmatrix} J^T, \quad (4.1.7)$$

with partial derivation of eq. (4.1.5)

$$J = \frac{\partial \mathbf{m}^i}{\partial (\mathbf{m}^c, \mathbf{K})} = \begin{bmatrix} \alpha & 0 & x & 1 & 0 \\ 0 & \alpha & y & 0 & 1 \end{bmatrix}. \quad (4.1.8)$$

For the reverse direction $\tilde{\mathbf{m}}^c = \mathbf{K}^{-1} \tilde{\mathbf{m}}^i$ error propagation gives

$$\Sigma_{xy} = J \begin{bmatrix} \Sigma_{uv} & \\ & \Sigma_{io} \end{bmatrix} J^T, \quad (4.1.9)$$

with

$$J = \frac{\partial \mathbf{m}^c}{\partial (\mathbf{m}^i, \mathbf{K})} = \frac{1}{\alpha} \begin{bmatrix} 1 & 0 & -x & -1 & 0 \\ 0 & 1 & -y & 0 & -1 \end{bmatrix}. \quad (4.1.10)$$

If normalized image coordinates are not needed, a direct mapping from world-frame-object points to ideal image points is realized by a 3×4 projection matrix P , containing the parameters of interior and exterior orientation

$$\tilde{\mathbf{m}}^i = P \tilde{\mathbf{M}}^w, \quad (4.1.11)$$

with

$$P = \mathbf{K} \left[\mathbf{R}_m^c | \mathbf{t}_w^c \right]. \quad (4.1.12)$$

4.1.2 Lens Distortion

Real lenses, however, usually do not sufficiently fit into the pinhole camera model. Optical aberrations, such as distortion, chromatic aberration, astigmatism, coma, etc. can occur. Distortion is geometrically the most significant and can be corrected with polynomial model functions by applying normalized camera coordinates.

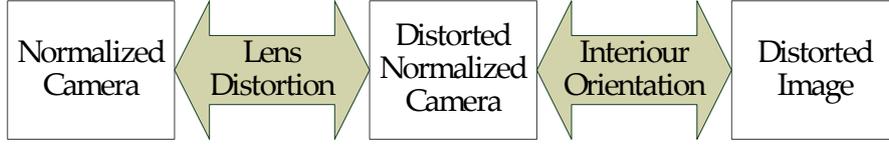


Figure 4.3: Coordinated frames and transformations for real lenses

Normalized Camera Frame \leftrightarrow **Distorted Image Frame** Applying a distortion model to normalized camera coordinates gives distorted normalized camera coordinates $\hat{\mathbf{m}}^c = (\hat{x}, \hat{y}, 1)^T$. Similar to eq. (4.1.5), distorted image coordinates $\hat{\mathbf{m}}^i = (\hat{u}, \hat{v}, 1)^T$ are obtained by a transformation with the camera matrix \mathbf{K} .

$$\hat{\mathbf{m}}^i = \mathbf{K}\hat{\mathbf{m}}^c \quad (4.1.13)$$

Most common is the correction with the distortion model of D. Brown (1971), which models radial symmetric distortion δ_r and decenteric distortion δ_t .

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \delta_r(x, y, \mathbf{k}) + \delta_t(x, y, \mathbf{p}) \quad (4.1.14)$$

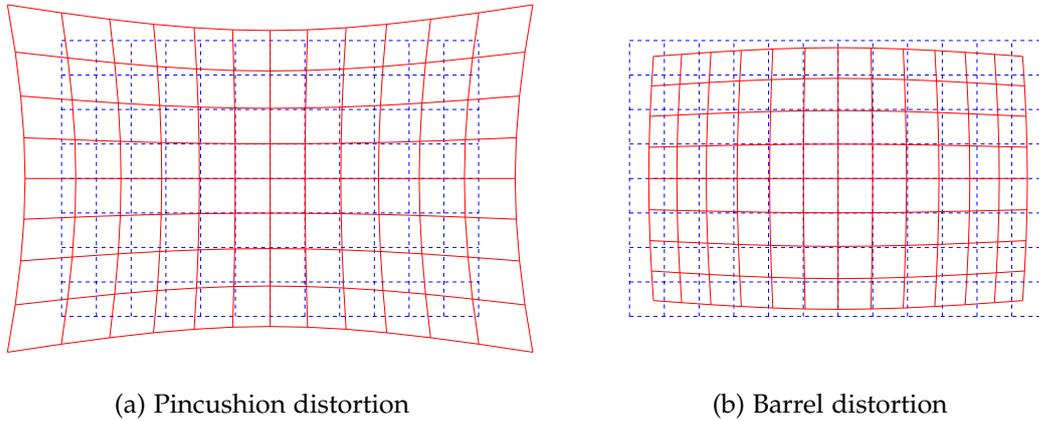


Figure 4.4: Polynomial lens distortion

To date, this model has been and continues to be used in many applications (Weng et al., 1992; Zhang, 1996; Heikkilä and Silven, 1997; T. Clarke, Fryer, and Wang, 1998; Zhang, 2000; Devernay and Faugeras, 2001). The uncertainty of a distorted camera coordinate depends on the covariance of the normalized camera coordinate itself but also on the uncertainty of the model parameters

Σ_k, Σ_p obtained from a calibration process:

$$\Sigma_{\hat{x}\hat{y}} = J_{xy}\Sigma_{xy}J_{xy}^T + J_k\Sigma_kJ_k^T + J_p\Sigma_pJ_p^T, \quad (4.1.15)$$

with

$$J_{xy} = \frac{\partial \hat{m}^c}{\partial m^c} = \begin{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{\partial \delta_r}{\partial x} + \frac{\partial \delta_t}{\partial x} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \frac{\partial \delta_r}{\partial y} + \frac{\partial \delta_t}{\partial y} \end{bmatrix} \quad (4.1.16)$$

$$J_k = \frac{\partial \hat{m}^c}{\partial k} = \begin{bmatrix} \frac{\partial \delta_r}{\partial k_1} & \frac{\partial \delta_r}{\partial k_2} & \frac{\partial \delta_r}{\partial k_3} \end{bmatrix} \quad (4.1.17)$$

$$J_p = \frac{\partial \hat{m}^c}{\partial p} = \begin{bmatrix} \frac{\partial \delta_t}{\partial p_1} & \frac{\partial \delta_t}{\partial p_2} \end{bmatrix}. \quad (4.1.18)$$

Radial Distortion Usually it is sufficient to correct for radial lens distortion, which is caused by thick lenses in connection with an aperture stop. If the magnification decreases with distance from the optical center, it is called negative distortion or barrel distortion. Positive distortion or pincushion distortion show increasing magnification. Equation (4.1.19) describes the symmetric radial distortion as a polynomial with coefficients $k = (k_1, k_2, \dots, k_n)$.

$$\delta_r(x, y, k) = \begin{pmatrix} x \\ y \end{pmatrix} \cdot \sum_{n=1}^{\infty} (k_n r^{2n}) \quad (4.1.19)$$

with

$$r^2 = x^2 + y^2. \quad (4.1.20)$$

The partial derivatives of the radial distortion w.r.t. normalized camera coordinates x, y and the distortion coefficients are given with

$$\frac{\partial \delta_r}{\partial x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \sum_{n=1}^{\infty} (k_n r^{2n}) + \begin{pmatrix} x \\ y \end{pmatrix} \cdot 2x \sum_{n=1}^{\infty} (nk_n r^{2(n-1)}) \quad (4.1.21)$$

$$\frac{\partial \delta_r}{\partial y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \sum_{n=1}^{\infty} (k_n r^{2n}) + \begin{pmatrix} x \\ y \end{pmatrix} \cdot 2y \sum_{n=1}^{\infty} (nk_n r^{2(n-1)}) \quad (4.1.22)$$

$$\frac{\partial \delta_r}{\partial k_n} = \begin{pmatrix} x \\ y \end{pmatrix} k_n r^{2n}. \quad (4.1.23)$$

Decentric Distortion If the optical centers of different lenses are shifted, decentric distortion occurs. It contains radial parts as well as tangential parts and is modeled with coefficients $\mathbf{p} = (p_1, p_2, \dots, p_n]$. This model might be useful for very cheap lenses, like those used for mobile phones or web cams.

$$\delta_t(x, y, \mathbf{p}) = \begin{pmatrix} p_1(3x^2 + y^2) + 2p_2xy \\ p_2(x^2 + 3y^2) + 2p_1xy \end{pmatrix} (1 + p_3r^2 + \dots) \quad (4.1.24)$$

With Jacobians w.r.t. normalized camera coordinates and distortion coefficients.

$$\frac{\partial \delta_t}{\partial x} = 2 \begin{pmatrix} 3p_1x + p_2y \\ p_2x + p_1y \end{pmatrix} \quad (4.1.25)$$

$$\frac{\partial \delta_t}{\partial y} = 2 \begin{pmatrix} p_1y + p_2x \\ 3p_2y + p_1x \end{pmatrix} \quad (4.1.26)$$

$$\frac{\partial \delta_t}{\partial p_1} = \begin{pmatrix} 3x^2 + y^2 \\ 2xy \end{pmatrix} \quad (4.1.27)$$

$$\frac{\partial \delta_t}{\partial p_2} = \begin{pmatrix} 2xy \\ x^2 + 3y^2 \end{pmatrix} \quad (4.1.28)$$

4.2 Stereo Camera View

We have seen that the depth information gets lost in a single camera view. One way to avoid this loss is with stereo vision. An object point can be reconstructed by triangulation of correspondent image points $\mathbf{m} \leftrightarrow \mathbf{m}'$ from two cameras with known exterior orientation. An uncertainty evaluation shows the importance of an accurate camera calibration and registration.

4.2.1 Epipolar Geometry

Epipolar geometry describes the essential geometry between two camera views. As seen in fig. 4.5, the object point $\tilde{\mathbf{M}}^w$, both image points $\mathbf{m}^{i1}, \mathbf{m}^{i2}$, and the centers of projection \mathbf{c}, \mathbf{c}' are coplanar, spanning the epipolar plane. The intersection of the epipolar plane with the image planes are the epipolar lines. Given only the baseline between \mathbf{c}_1 and \mathbf{c}_2 with an image point \mathbf{m}^{i1} in the first image, the correspondent image point \mathbf{m}^{i2} in the second image must lie on the epipolar line $l_2 = \mathbf{F}\mathbf{m}^{i1}$.

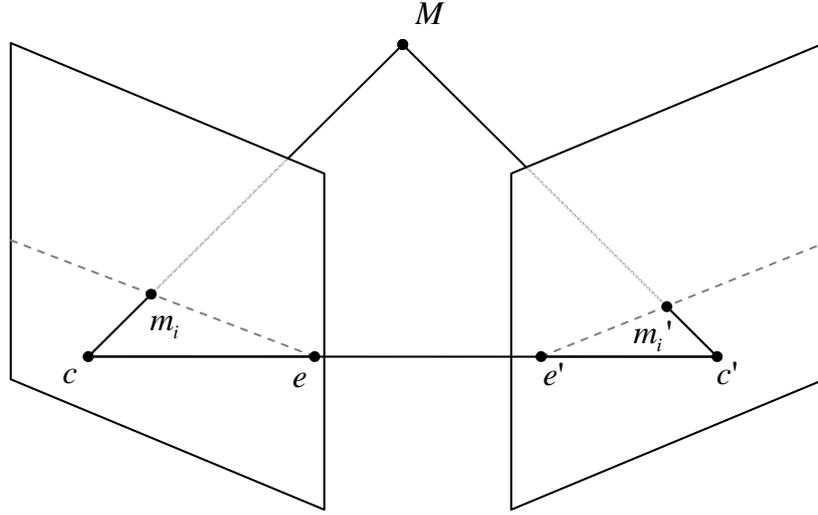


Figure 4.5: Epipolar geometry

The fundamental matrix F , which is dependent on the cameras' interior and exterior orientation, connects correspondent image points with $(\tilde{\mathbf{m}}^{i2})^T F \tilde{\mathbf{m}}^{i1} = 0$. For a stereo setup with orientations $P_1 = K_1[I|0]$ and $P_2 = K_2[R_{c1}^{c2}|t_{c1}^{c2}]$ applies

$$F = K_2^{-T} [t_{c1}^{c2}]_{\times} R_{c1}^{c2} K_1^{-1}. \quad (4.2.1)$$

When using normalized camera coordinates, the interior orientation of the cameras is removed giving normalized projection matrices $P_1 = [I|0]$ and $P_2 = [R_{c1}^{c2}|t_{c1}^{c2}]$. A fundamental matrix for normalized cameras is called essential matrix

$$E = [t_{c1}^{c2}]_{\times} R_{c1}^{c2}, \quad (4.2.2)$$

satisfying the epipolar constraint $(\tilde{\mathbf{m}}^{c2})^T E \tilde{\mathbf{m}}^{c1} = 0$.

4.2.2 Triangulation

The co-planarity of correspondent points and their projection centers, satisfying the epipolar constraint $(\tilde{\mathbf{m}}^{c2})^T E \tilde{\mathbf{m}}^{c1} = 0$, is illustrated in fig. 4.5. Measurement errors, occurring in real scenarios make it necessary to think about optimization methods for the object point reconstruction. Many different methods are proposed (Hartley and Sturm, 1995).

The linear method minimizes the norm $\|\epsilon\| = \|A\mathbf{x}\|$ in projective space \mathbb{P}^3 , which is geometrically less meaningful but very fast and handles points at infinity.

With a normalized projection matrix, an object point is projected to a normalized camera coordinate with $\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}}^w$. Using the cross product $\tilde{\mathbf{m}} \times \mathbf{P}\tilde{\mathbf{M}}^w = 0$ eliminates the scaling factor and gives

$$\begin{aligned} x \left(\mathbf{p}^{3T} \tilde{\mathbf{M}}^w \right) - \left(\mathbf{p}^{1T} \tilde{\mathbf{M}}^w \right) &= 0 \\ y \left(\mathbf{p}^{3T} \tilde{\mathbf{M}}^w \right) - \left(\mathbf{p}^{2T} \tilde{\mathbf{M}}^w \right) &= 0 \\ x \left(\mathbf{p}^{2T} \tilde{\mathbf{M}}^w \right) - y \left(\mathbf{p}^{1T} \tilde{\mathbf{M}}^w \right) &= 0, \end{aligned}$$

with $\tilde{\mathbf{m}} = (x, y, 1)^T$ and \mathbf{p}^{iT} denoting the rows of the matrix \mathbf{P} . Two of the three equations obtained are linearly independent.

The combination of the two measurements $\hat{\mathbf{m}}^{c1} \times \mathbf{P}_1 \tilde{\mathbf{M}}^w$ and $\hat{\mathbf{m}}^{c2} \times \mathbf{P}_2 \tilde{\mathbf{M}}^w$ leads to a homogeneous set of equations of the form $\mathbf{A}\mathbf{x} = 0$, with

$$\mathbf{A} = \begin{bmatrix} \hat{x}\mathbf{p}_1^{3T} - \mathbf{p}_1^{1T} \\ \hat{y}\mathbf{p}_1^{3T} - \mathbf{p}_1^{2T} \\ \hat{x}'\mathbf{p}_2^{3T} - \mathbf{p}_2^{1T} \\ \hat{y}'\mathbf{p}_2^{3T} - \mathbf{p}_2^{2T} \end{bmatrix}, \quad (4.2.3)$$

with $\hat{\mathbf{m}}^{c1} = (\hat{x}, \hat{y}, 1)^T$ and $\hat{\mathbf{m}}^{c2} = (\hat{x}', \hat{y}', 1)^T$. A non-zero solution to this homogeneous equation is given in appendix B.2.3. It also shows the uncertainty calculation giving a 4×4 covariance matrix.

A better way is minimize a geometrical distance in image space \mathbb{R}^2 . The noisy measurements $\hat{\mathbf{m}}^{c1}, \hat{\mathbf{m}}^{c2}$ are corrected to satisfy the epipolar constraint. Besides the complete polynomial solution, which involves finding the root of a polynomial of sixth order an approximation is given with

$$\begin{pmatrix} x \\ y \\ x' \\ y' \end{pmatrix} \approx \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{x}' \\ \hat{y}' \end{pmatrix} - \frac{(\hat{\mathbf{m}}^{c2})^T \mathbf{E} \hat{\mathbf{m}}^{c1}}{(\mathbf{E} \hat{\mathbf{m}}^{c1})_1^2 + (\mathbf{E} \hat{\mathbf{m}}^{c1})_2^2 + (\mathbf{E}^T \hat{\mathbf{m}}^{c2})_1^2 + (\mathbf{E}^T \hat{\mathbf{m}}^{c2})_2^2} \begin{pmatrix} (\mathbf{E}^T \hat{\mathbf{m}}^{c2})_1 \\ (\mathbf{E}^T \hat{\mathbf{m}}^{c2})_2 \\ (\mathbf{E} \hat{\mathbf{m}}^{c1})_1 \\ (\mathbf{E} \hat{\mathbf{m}}^{c1})_2 \end{pmatrix}, \quad (4.2.4)$$

where $(\mathbf{E}\tilde{\mathbf{m}})_i$ is the i -th vector element. The Sampson-approximation (Hartley and Zisserman, 2000) corrects for first order measurement errors. It can also be used for a consistency check as bigger corrections are an indication for mismatched points. After correction, the object point is calculated accordingly to eq. (4.2.3).

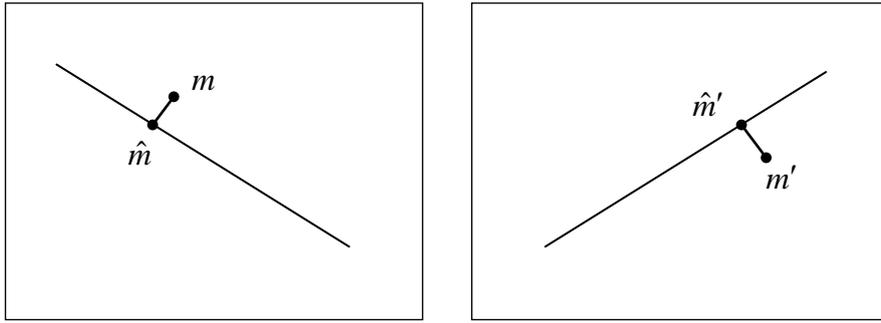


Figure 4.6: Geometric error correction

4.3 Pose Estimation

Homologous image points in two successive stereo frames are used to estimate the egomotion of a stereo system as seen in fig. 4.7. Object points triangulated

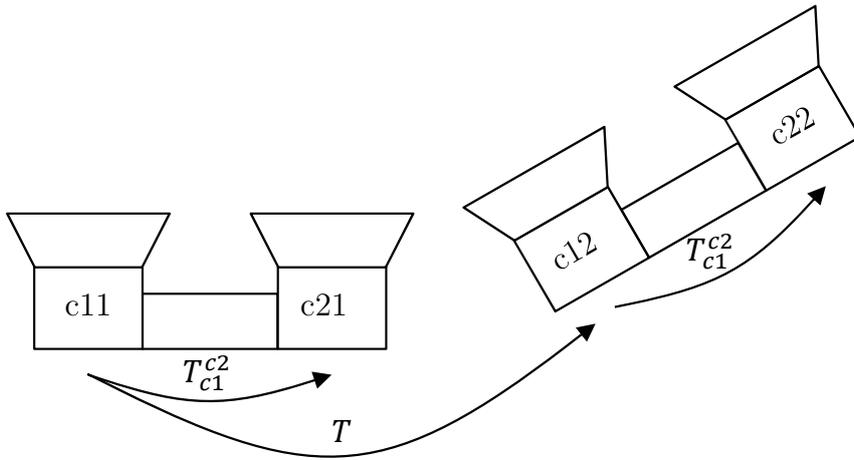


Figure 4.7: Camera frames for pose estimation

from both stereo frames are connected by the transformation T .

$$\tilde{M}^{c12} = T\tilde{M}^{c11} \quad (4.3.1)$$

Minimizing for T in projective space \mathbb{P}^3 does not provide a meaningful error measure. Therefore, it is better to minimize distances in \mathbb{R}^2 , e.g. in normalized camera coordinates. The transformed object coordinates have to be projected to get normalized camera coordinates

$$\mathbf{m}^{c12} = \left(\tilde{M}_x^{c12} / \tilde{M}_z^{c12}, \tilde{M}_y^{c12} / \tilde{M}_z^{c12} \right)^T. \quad (4.3.2)$$

To find the relative transformation $T = f(\omega, \varphi, \kappa, t_x, t_y, t_z)$ between both frames, these coordinates are compared to measured normalized camera coordinates \hat{m}^{c12} from the reference camera of the second stereo frame.

$$\min_T \|\hat{m}^{c12} - m^{c12}\|^2 \quad (4.3.3)$$

This non-linear least square optimization problem is solved accordingly to appendix B.2.2 with the Jacobian

$$J_T = \frac{\partial m^{c12}}{\partial T} = \frac{\partial m^{c12}}{\partial \tilde{M}^{c12}} \frac{\partial \tilde{M}^{c12}}{\partial T}, \quad (4.3.4)$$

with partial derivatives of the normalized camera coordinate w.r.t. the transformed object point from eq. (4.1.4) and the object point w.r.t. to T .

The quality of the pose estimation depends not only on the quality of the calibration and registration of the cameras; the number of homologous feature points, their distance, and their distribution within the observed scene also have a strong influence. Therefore, the uncertainty of the solution can vary widely, which has to be taken into account for the successive data processing. As stated in appendix B.2.2 the Jacobians of the shift vector w.r.t. the error sources have to be determined. A major sources of error will be the triangulated object point of the first stereo frame, with the Jacobian

$$J_{M^{c11}} = \frac{\partial m^{c12}}{\partial \tilde{M}^{c11}} = \frac{\partial m^{c12}}{\partial \tilde{M}^{c12}} \frac{\partial \tilde{M}^{c12}}{\partial \tilde{M}^{c11}}. \quad (4.3.5)$$

In general, it is sufficient to use only image points of the reference camera of the second stereo frame for the estimation process. By using feature points from the second camera as well, the number of measurements and therefore the accuracy of the solution is increased. The procedure is similar to that of the reference camera, only the relative transformation of both cameras T_{c1}^{c2} has to be added.

$$\tilde{M}^{c22} = T_{c1}^{c2} T \tilde{M}^{c11} \quad (4.3.6)$$

It has to be noted that the triangulated object point \tilde{M}^{c11} is already dependent on T_{c1}^{c2} . Instead of using the covariance of the object point directly, it is therefore necessary to use the Jacobians of the object point w.r.t. the sources of error within the triangulation.

$$J_{T_{c1}^{c2}} = \frac{\partial m^{c22}}{\partial T_{c1}^{c2}} = \frac{\partial m^{c22}}{\partial \tilde{M}^{c22}} \left(\frac{\partial T_{c1}^{c2}}{\partial T_{c1}^{c2}} T \tilde{M}^{c11} + T_{c1}^{c2} T \frac{\partial \tilde{M}^{c11}}{\partial T_{c1}^{c2}} \right) \quad (4.3.7)$$

Besides the error of the relative orientation, errors of the normalized camera coordinates m^{c11}, m^{c21} in the left, respective right image of the first stereo frame are considered. Jacobians are as follows:

$$J_{m^{c11}} = \frac{\partial m^{c22}}{\partial m^{c11}} = \frac{\partial m^{c22}}{\partial \tilde{M}^{c22}} \frac{\partial \tilde{M}^{c22}}{\partial \tilde{M}^{c11}} \frac{\partial \tilde{M}^{c11}}{\partial m^{c11}} \quad (4.3.8)$$

$$J_{m^{c21}} = \frac{\partial m^{c22}}{\partial m^{c21}} = \frac{\partial m^{c22}}{\partial \tilde{M}^{c22}} \frac{\partial \tilde{M}^{c22}}{\partial \tilde{M}^{c11}} \frac{\partial \tilde{M}^{c11}}{\partial m^{c21}} \quad (4.3.9)$$

To solve eq. (4.3.3), a perfect data set of homologous image points is needed. This can not be guaranteed due to mismatching at repetitive structures or low textured areas. Single outliers can be crucial as they introduce significant errors to the solution or can even cause the minimization to fail. Even more, the proportion of mismatched points can be quite high.

Therefore, the Random Sample Consensus (RANSAC) introduced by Fischler and Bolles (1980) is used to estimate model parameters in the presents of outliers. Instead of using all points of the data set, the model parameters are estimated with a randomly chosen minimum data set. The solution is then applied to the complete data set, which allows for the counting data points that are compatible to said solution. The number of iterations needed to find the optimal data set with a probability of p is given with

$$k = \frac{\log(1 - p)}{\log(1 - w^n)} \quad (4.3.10)$$

where w is the proportion of in-liners with respect to the complete data set and n being the minimum number of data points to find the model parameters.

4.3.1 Feature Extraction

To perform the aforementioned steps, it is necessary to extract a selection of feature points that are reliably detectable over consecutive frames. Natural landmarks, such as corners, isolated points or line endings, meet these requirements. Harris and Stephens (1988) proposed to locate these feature points by analyzing the autocorrelation matrix

$$A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (4.3.11)$$

with the partial derivatives of the image intensity $I_x = \partial I / \partial x$ and $I_y = \partial I / \partial y$. Convolving the image with the Sobel operator approximates the image intensity gradients with a slightly higher weight in the direction of the gradients.

$$\partial I / \partial x \approx I \star \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4.3.12)$$

$$\partial I / \partial y \approx I \star \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.3.13)$$

The eigenvalues λ_1, λ_2 of the matrix A serve as a rotational invariant measure of the curvature of the autocorrelation matrix. For both eigenvalues being small the region looked at is of constant intensity. If one eigenvalue is higher than the other, this indicates an edge; whereas, both eigenvalues being high specifies a corner. Shi and Tomasi (1994) proposed that a corner is found if

$$\min(\lambda_1, \lambda_2) > \lambda, \quad (4.3.14)$$

with λ as a threshold value. A non-maximum suppression follows, which keeps only local maximums to guarantee a minimum distance between each feature. In practical applications, a constant threshold turns out to be a problem for differently textured scenes. Therefore, an automatic adaptation similar to an automatic exposure is needed. This can be done by defining the desired number of features to be found. Starting with some initial threshold, the current number of features is determined. If too many features are found, the threshold is increased for the next image, respectively decreased if too few features are found.

4.3.2 Feature Matching

A prerequisite for triangulation or pose estimation is the identification of homologous feature points in multiple images. Template-based approaches, describing a feature according to its local neighborhood, can be used for matching. The best match can be determined through convolution of the template with the region of interest within the search image. Although different correlation methods are used, Normalized Cross Correlation (NCC) will be applied here because of its

robustness against changes in lighting and exposure. The matching coefficient is calculated as follows:

$$\gamma(u, v) = \frac{1}{n} \sum_{i,j} \frac{(S(u+i, v+j) - \bar{S})(T(i, j) - \bar{T})}{\sigma_S \sigma_T}, \quad (4.3.15)$$

with the search image S , the template T with $n = i \cdot j$ pixels and the mean \bar{S}, \bar{T} , respective to the standard deviation, σ_S, σ_T of the templates in both images.

Having real-time applications in mind, the restricting the region of interest within the search image is a main objective. This not only speeds up the matching process, it also avoids mismatching, often encountered in indoor environments, e.g. at repetitive structures. Dependent on the available information, two types of regions of interests can be distinguished (see fig. 4.8).

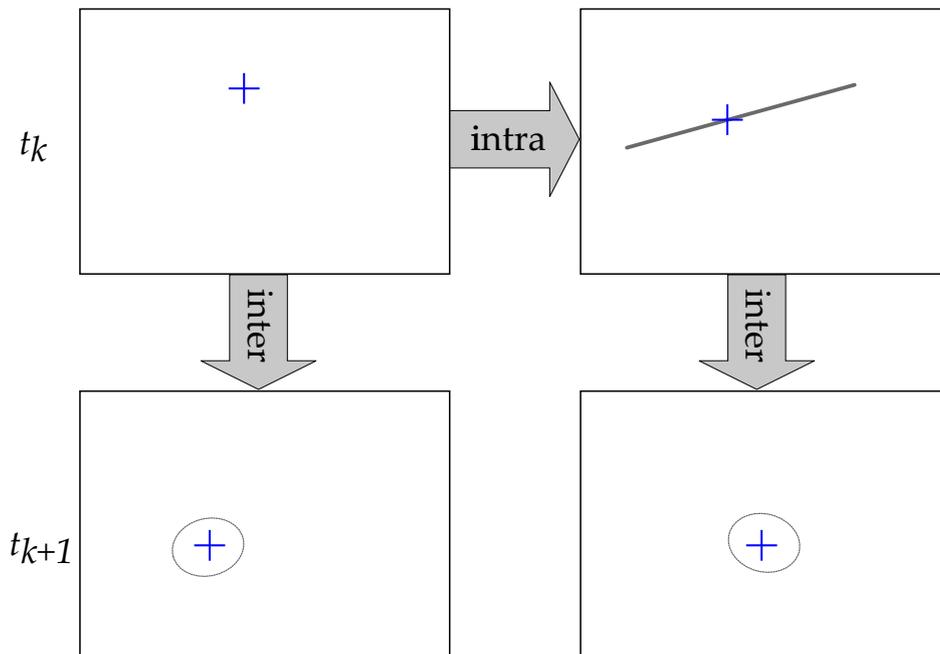


Figure 4.8: Paths for intra-frame and inter-frame matching with search areas

Intra-frame matching For known orientation between both views but unknown object points, the epipolar geometry is used to confine the search area to the epipolar line (see section 4.2.1). The search for corresponding features within a stereo frame with a precisely known calibration can be confined to a one-pixel-thick epipolar line. For dedicated applications, the minimum and maximum

distance of considered object points can be limited. This also limits the length of the epipolar line, resulting in a further reduction of the search area.

Inter-frame matching With known object points, e.g. from triangulation and known relative orientation, the image point in the second frame could be calculated directly with eqs. (4.1.1) and (4.1.3), which would render the matching obsolete. Still, large uncertainties on these input parameters cause a significant uncertainty on the calculated image point. Matching is needed again. The resulting covariance is now used to determine the size of the search area.

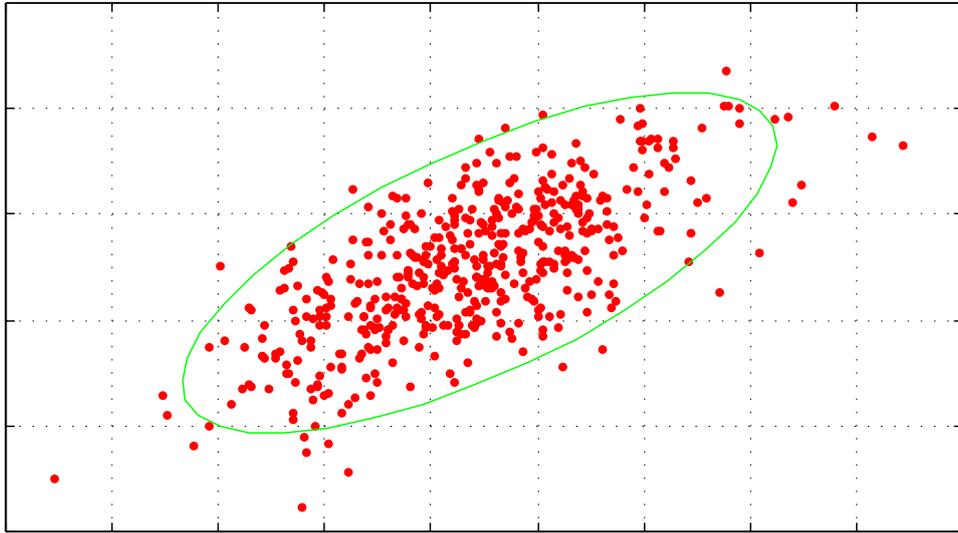


Figure 4.9: Elliptical 95 % confidence region

J. Clarke (1998) stated that the elliptical confidence region for a point $m = (x, y)^T$ with covariance Σ_m is given by points x satisfying

$$\chi^2 = (x - m)^T \Sigma_m (x - m). \quad (4.3.16)$$

The size of the region is determined by the χ^2 -distribution, which is used to calculate the confidence interval for multiple independent standard normal distributed variables. For two degrees of freedom, it is given with

$$\chi^2 = -2 \log(1 - p), \quad (4.3.17)$$

where p is the probability of a value to be within the confidence interval. The points x from eq. (4.3.16) are calculated as follows:

$$x_i = m + \chi \sqrt{d_1} \sin(\theta_i) v_1 + \chi \sqrt{d_2} \cos(\theta_i) v_2, \quad (4.3.18)$$

with $\theta_i = 0 \dots 2\pi$ and v, d being the eigenvalues and eigenvectors of the eigen-decomposition $\Sigma_m = VDV^{-1}$. As covariance matrices are symmetric, eigenvalues and eigenvectors can be calculated more efficiently with the singular value decomposition $\Sigma_m = VDV^T$.

4.4 Calibration and Registration

Measuring in images requires exact knowledge of the camera model and an additional non-linear distortion model. Several methods for calibrating camera systems have been proposed. Many of them use predefined calibration grids to extract camera parameters with a complex bundle adjustment (Tsai, 1987; Zhang, 2000; Strobl et al., 2005). To separate all parameters safely, several observations with different viewing angles and distances are needed. Although this method is very common, it is prone to errors if not conducted properly. Because the size of the target is limited, it is also not suitable for telephoto lens calibration.

A way to eliminate those difficulties is the use of collimated light to illuminate single pixels. Rotating either the light source or the camera by known angles produces a well defined grid of points at infinity (Schuster and Braunecker, 2000). These points are invariant against translation, an important advantage as it makes multiple pose observations obsolete, resulting in a more stable estimation process. As this method is rather time consuming and mechanically challenging, a holographic pattern produced by an Diffractive Optical Element (DOE) is used to achieve a fast and reliable camera calibration (Grießbach et al., 2008; Grießbach, Bauer, et al., 2010).

4.4.1 Diffractive Optical Element

The advantages of the classical calibration grid approach and the single pixel illumination method are combined by using a DOE. It works as a beam splitter with precisely known diffraction angles. The virtual sources of the diffracted beams are points at infinity, which yields an image that is invariant against translation.

A collimated laser beam with wavelength λ is split in well known propagation

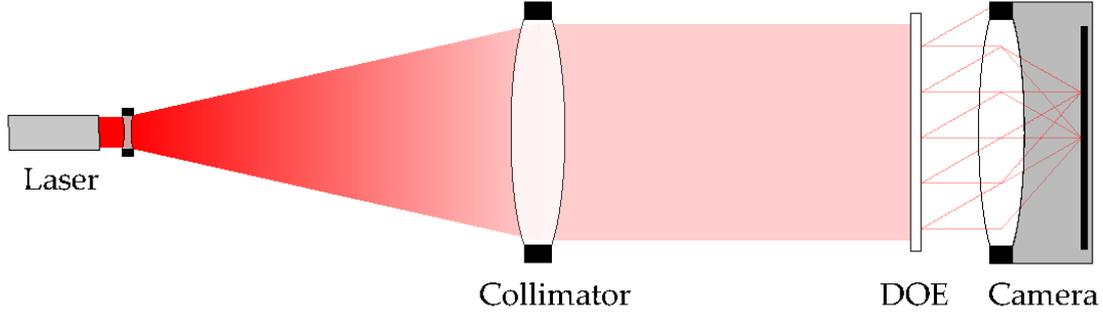


Figure 4.10: Schematic DOE calibration setup

directions denoted with

$$\tilde{M}^d = \begin{pmatrix} \lambda f_x \\ \lambda f_y \\ \sqrt{1 - \lambda^2 (f_x^2 + f_y^2)} \\ 0 \end{pmatrix}, \quad (4.4.1)$$

where f_x, f_y are spatial frequencies encoded in the DOE, which is given by $f_x = n_x/g$ and $f_y = n_y/g$, with n denoting the particular diffraction orders and the grating period g . The grating vectors define the x -axis and y -axis of the DOE coordinate frame. Table 4.1 shows the specification of the DOE used.

Diameter [mm]	75
Grating period [μm]	41.1
Angular spacing [$^\circ$] ¹	0.82–1.17

Table 4.1: DOE specification

Equation (4.4.1) holds if the incident light wave is a plane wave with uniform intensity distribution, perfectly perpendicular to the DOE surface. In a real setup, the beam is finite in extension and often has a non-uniform intensity profile, which is typically Gaussian. Deviations from the beam profile result in a certain spot size, which is not critical in terms of geometry. A slight tilt of the DOE with regard to the incident beam is technically almost unavoidable. The incident beam is therefore given by $\mathbf{r} = (\sin(\beta), -\sin(\alpha) \cos(\beta), \cos(\alpha) \cos(\beta))^T$ with α and β representing a rotation of the x -axis and y -axis of the DOE coordinate frame w.r.t.

¹ for $\lambda = 632.8 \text{ nm}$ and diffraction orders 0–50

the collimator coordinate frame. The directions of the diffracted beams are now obtained as follows (McPhedran et al., 1980).

$$\tilde{\mathbf{M}}^d = \begin{pmatrix} \lambda f_x + r_x \\ \lambda f_y + r_y \\ \sqrt{1 - (\lambda f_x + r_x)^2 - (\lambda f_y + r_y)^2} \\ 0 \end{pmatrix} \quad (4.4.2)$$

Diffraction pattern uncertainty A main advantage of the DOE method is the high accuracy of the diffraction pattern that is produced without involving a mechanical device, e.g. a goniometer. But still, any error induced by a varying wavelength has a direct impact on the accuracy of the pattern. From eq. (4.4.1) follows the resulting diffraction angle:

$$\Theta = \arccos \left(\sqrt{1 - \lambda^2 (f_x^2 + f_y^2)} \right). \quad (4.4.3)$$

Error propagation for the angular error w.r.t. the wavelength leads to

$$\sigma_{\Theta} = \frac{\partial \Theta}{\partial \lambda} \sigma_{\lambda}, \quad (4.4.4)$$

with

$$\frac{\partial \Theta}{\partial \lambda} = \sqrt{\frac{f_x^2 + f_y^2}{1 - \lambda^2 (f_x^2 + f_y^2)}}. \quad (4.4.5)$$

A helium-neon laser with a wavelength of 632.8 nm and a frequency instability of 10 MHz is used. With $\lambda = c/f_{\lambda}$ and

$$\sigma_{\lambda} = \frac{\partial \lambda}{\partial f_{\lambda}} \sigma_{f_{\lambda}} = \frac{\lambda^2}{c} \sigma_{f_{\lambda}}, \quad (4.4.6)$$

where c denotes the speed of light, the wavelength uncertainty σ_{λ} is calculated to 0.0133 nm. In combination with the DOE given in table 4.1, this leads to a angle uncertainty of 0.0012° at the maximum seen diffraction order ($n = 50$).

4.4.2 Single View Calibration

Because the projection of the diffraction pattern onto the image plane is invariant against translation, the complete camera calibration can be done with a single image as seen in fig. 4.11. Complex bundle adjustments with multiple poses are

avoided. The exterior orientation of the DOE frame w.r.t. the camera frame only consists of the rotation matrix \mathbf{R}_d^c . From eq. (4.1.11) ideal camera coordinates of a diffraction point are given with

$$(x, y, 1)^T = [\mathbf{R}_d^c | 0] \cdot \tilde{\mathbf{M}}^d. \quad (4.4.7)$$

By using eq. (4.1.14), the mapping to distorted image coordinates $(\hat{u}, \hat{v})^T$ is subsumed to

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + f \begin{pmatrix} x \\ y \end{pmatrix} \left(1 + \delta_r(x, y, \mathbf{k}) + \delta_t(x, y, \mathbf{p}) \right). \quad (4.4.8)$$

Given a set of correspondent points $(x, y)^T \leftrightarrow (\hat{u}, \hat{v})^T$, we seek to minimize the non-linear cost function

$$\min_m \left\| \begin{pmatrix} \hat{u} - u_0 \\ \hat{v} - v_0 \end{pmatrix} - f \begin{pmatrix} x \\ y \end{pmatrix} \left(1 + \delta_r(x, y, \mathbf{k}) + \delta_t(x, y, \mathbf{p}) \right) \right\|^2, \quad (4.4.9)$$

where $m = (f, u_0, v_0, \mathbf{k}, \mathbf{p}, \omega, \varphi, \kappa, \alpha, \beta)^T$ describes the interior and exterior orientation of the camera $\mathbf{R}_d^c(\omega, \varphi, \kappa)$, a possible rotation $\mathbf{R}_{col}^d(\alpha, \beta)$ of the DOE frame w.r.t. the collimation coordinate frame as well as the coefficients of the distortion model. The cost function is minimized using the Jacobians from eqs. (4.1.16) to (4.1.18), (4.1.21) to (4.1.23) and (4.1.25) to (4.1.27).

4.4.3 Stereo Registration

When estimating the parameters of interior orientation, the exterior orientation of the camera w.r.t. the DOE frame \mathbf{R}_d^c is also estimated as a by-product. Suppose that the stereo rig movement between separate camera calibrations is a pure translation than the relative rotation is calculated with

$$\mathbf{R}_{c1}^{c2} = \mathbf{R}_d^{c2} (\mathbf{R}_d^{c1})^T. \quad (4.4.10)$$

It is not possible to obtain the relative translation because the target with points at infinity is invariant against translation. For this reason, a single view of a chessboard pattern is used to complete the registration (Grießbach, Bauer, et al., 2010). The chessboard rows and columns define the x - y -plane of the chessboard coordinate frame. All corners of the pattern can therefore be given

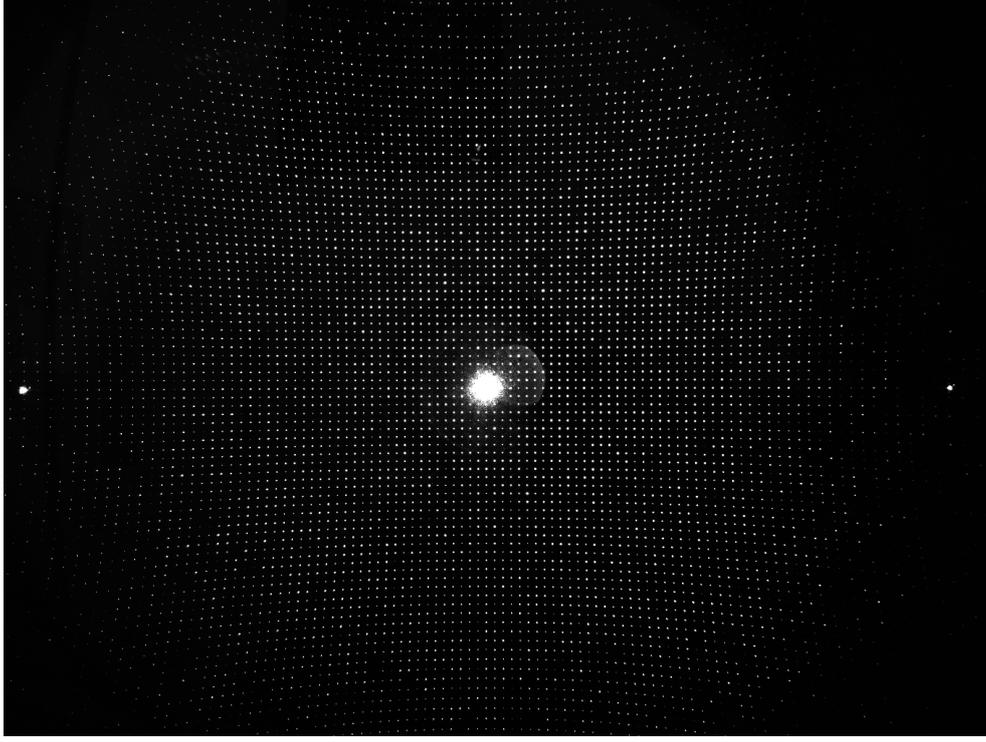


Figure 4.11: DOE calibration image for a wide angle lens

as homogeneous coordinates $\tilde{M}^{ch} = (X, Y, 0, 1)^T$ w.r.t. an arbitrary origin. By applying eq. (4.1.11) they are projected to image points of the left and right camera as follows:

$$\tilde{m}^{c1} = K_{c1} \begin{bmatrix} R_{ch}^{c1} & t_{ch}^{c1} \end{bmatrix} \tilde{M}^{ch} \quad (4.4.11)$$

$$\tilde{m}^{c2} = K_{c2} \begin{bmatrix} R_{c1}^{c2} R_{ch}^{c1} & R_{c1}^{c2} t_{ch}^{c1} + t_{c1}^{c2} \end{bmatrix} \tilde{M}^{ch} \quad (4.4.12)$$

A bundle block adjustment minimizes the squared difference to the measured image points \hat{m}^{c1} and \hat{m}^{c2} of both cameras by estimating the orientation of the left camera w.r.t. the chessboard and the unknown translation from left to right camera.

$$\min_{R_{ch}^{c1}, t_{ch}^{c1}, t_{c1}^{c2}} \left\| \hat{m}^{c1} - \tilde{m}^{c1} \right\|^2 + \left\| \hat{m}^{c2} - \tilde{m}^{c2} \right\|^2 \quad (4.4.13)$$

It is also possible to estimate the rotation between the cameras in the same process. In practice this should be avoided because of strong correlations between rotational and translational parameters. For a good parameter separation, multiple poses are necessary.

4.5 Experimental Setup

The stereo system consists of two panchromatic CCD-cameras with 4.8 mm Schneider-Kreuznach lenses as shown in table 4.2. A highly sensitive CCD allows for low noise images even in difficult lighting conditions. To support high data rates for full resolution images, a GigaBit Ethernet interface is provided.

	Prosilica GC1380H
Sensor	2/3" CCD
Resolution [pixel]	1360×1024
Frame rate [Hz]	≤ 30
Pixel size [μm]	6.45
Focal length [mm]	4.8
Instantaneous field of view [°]	0.077
Field of view [°]	85×69, 98 (diagonal)

Table 4.2: Camera specifications

In combination with the DOE shown in table 4.1, a very dense pattern with about 9000 diffraction points can be used for calibration (see fig. 4.11). Correspondent points are found with an iterative approach constantly refining the model parameters. The residual error after optimization is about 0.15 pixel for both cameras. Due to the strong pincushion distortion, the real field of view is expanded to $95^\circ \times 76^\circ$ and 110° in diagonal.

	Camera 1	Camera 2
f [pixel]	776.07 ± 0.05	772.87 ± 0.05
u_0 [pixel]	711.58 ± 0.04	680.24 ± 0.04
v_0 [pixel]	547.89 ± 0.03	546.95 ± 0.04
k_0	$-259.39 \pm 0.15 \times 10^{-3}$	$-254.95 \pm 0.25 \times 10^{-3}$
k_1	$113.79 \pm 0.16 \times 10^{-3}$	$107.48 \pm 0.38 \times 10^{-3}$
k_2	$-25.91 \pm 0.04 \times 10^{-3}$	$-23.26 \pm 0.16 \times 10^{-3}$
points	6893	7204

Table 4.3: Camera calibration results

Both cameras are mounted on an optical bench to provide a stable setup. The registration of the cameras is a two step process. First, the rotation is calculated from the DOE calibration results according to eq. (4.4.10). A single checker board image is finally needed to determine the translation between both cameras using eq. (4.4.13).

ω [°]	0.335 ± 0.002
ϕ [°]	0.464 ± 0.003
κ [°]	0.384 ± 0.002
t_x [m]	$-200.89 \pm 0.06 \times 10^{-3}$
t_y [m]	$-0.04 \pm 0.06 \times 10^{-3}$
t_z [m]	$0.38 \pm 0.03 \times 10^{-3}$

Table 4.4: Stereo registration results

The chosen configuration with a stereo base line of 0.2 m is optimized for indoor environments. It gives an usable stereo range from 0.55 m with an image overlap of 80 % to 10 m with reasonable uncertainties from triangulation of about 0.6 m.

Chapter 5

Sensor Fusion

This chapter shows how inertial data and visual measurements can be integrated to achieve an optimal estimate of the sensor system motion. A very important prerequisite for data fusion is an exact model of the sensor system. This includes single sensor calibration, registration, and synchronization. Difficulties and possible solutions regarding these issues are pointed out. I have chosen a Kalman filter approach for optimal fusion of different sensor measurements in the presents of noise.

5.1 Sensor Registration

The term registration refers to the determination of the spatial relation between sensor systems. All sensors are registered w.r.t. the IMU because it is the main sensor in the system. What the following methods have in common is that they use static poses to estimate the rotational alignment of the IMU acceleration vector w.r.t. the local tangent plane (navigation frame). For the translational part, it would be necessary to look at the fictitious forces that are seen in a rotating reference frame with angular rate ω and displacement t .

$$a' = a - \underbrace{\omega \times \omega \times t}_{\text{centrifugal}} - \underbrace{\dot{\omega} \times t}_{\text{Euler}} - \underbrace{2\omega \times v}_{\text{Coriolis}} \quad (5.1.1)$$

As the expected accuracy with the low-cost IMUs used is rather low, the sensor displacements are measured manually or taken from a mechanical drawing.

5.1.1 IMU to Inclination Sensor

According to section 3.3 the inclination or tilt measurement can be rewritten as an acceleration $\hat{\mathbf{a}}^t$. Since the construction of the inclination sensor guaranties an absolute accuracy, no relevant scaling or bias errors will be assumed. Therefore, $\hat{\mathbf{a}}^t \approx \mathbf{a}^t$ applies. This measurement is connected to the IMU acceleration $\hat{\mathbf{a}}^i$ by the rotation $\mathbf{R}_i^t(\phi, \theta, \psi)$ of the IMU coordinate frame w.r.t. the tilt sensor frame. The translational displacement can be omitted since only rotations are measured. With the sensor error model from eq. (3.2.1) follows:

$$\mathbf{a}^t = \mathbf{R}_i^t \left(\mathbf{S}^{-1} (\hat{\mathbf{a}}^i - \mathbf{b}_a) \right). \quad (5.1.2)$$

The unknown Euler angles ϕ, θ, ψ , scaling factors \mathbf{S} , and acceleration bias \mathbf{b}_a are estimated in a non-linear minimization process. Again, independent measurements have to be provided to achieve uncorrelated results.

$$\min_{\phi, \theta, \psi, \mathbf{S}, \mathbf{b}_a} \left\| \mathbf{a}^t - \mathbf{R}_i^t \left(\mathbf{S}^{-1} (\hat{\mathbf{a}}^i - \mathbf{b}_a) \right) \right\|^2 \quad (5.1.3)$$

5.1.2 IMU to Camera

The rotation \mathbf{R}_i^c of the IMU coordinate frame w.r.t. the camera frame can be registered by means of acceleration measurements $\hat{\mathbf{a}}^i$ from static poses. When placed in front of a chessboard, the rotation \mathbf{R}_c^{ch} of the camera frame w.r.t. the chessboard frame is determined as shown in section 4.4.3. With the sensor error model from eq. (3.2.1) follows:

$$\mathbf{g}^n = \mathbf{R}_{ch}^n \mathbf{R}_c^{ch} \mathbf{R}_i^c \left(\mathbf{S}^{-1} (\hat{\mathbf{a}}^i - \mathbf{b}_a) \right), \quad (5.1.4)$$

where $\mathbf{g}^n = (0, 0, g)^T$ is the gravitational acceleration. The rotational alignment, along with the inclination \mathbf{R}_{ch}^n of the checker board w.r.t. the navigation frame, and the acceleration bias \mathbf{b}_a are determined in a non-linear estimation process.

$$\min_{\mathbf{R}_{ch}^n, \mathbf{R}_i^c, \mathbf{b}_a, \mathbf{S}} \left\| \mathbf{g}^n - \mathbf{R}_{ch}^n \mathbf{R}_c^{ch} \mathbf{R}_i^c \left(\mathbf{S}^{-1} (\hat{\mathbf{a}}^i - \mathbf{b}_a) \right) \right\|^2 \quad (5.1.5)$$

Optionally, the scaling factors \mathbf{S} of the acceleration sensors may also be estimated if unknown. It is of great importance to choose the appropriate poses to guarantee the observability of the unknown parameters.

5.1.3 Time Line Registration

Beside the spatial registration, it might be necessary to take latency times of different sensor systems into account. The time shift or phase τ between two signals f, g is determined by calculating their cross-correlation with

$$R_{fg}(\tau) = (f \star g)[\tau] \quad (5.1.6)$$

$$= \int_{-\infty}^{\infty} f^*(t)g(t + \tau)dt, \quad (5.1.7)$$

where f^* is the conjugate complex function which is f for real functions. Equation (5.1.7) is given in the time discrete form with

$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f^*[m]g[m + n]. \quad (5.1.8)$$

The time shift is given at the maximum value of the correlation function where both signals are best aligned.

$$\tau_{latency} = \arg \max R_{fg}(\tau) \quad (5.1.9)$$

To determine the time delay of the IMU – caused by their internal signal processing – w.r.t. the camera, both sensors have to provide a similar signal. This is found with the magnitude of the IMU angular velocity $\omega_i = |\omega_i|$. The camera signal is derived from the quaternion of the egomotion $T_c(t, q)$ by using eq. (3.1.7).

$$\omega_c = 2 \arccos(q_0)/dt \quad (5.1.10)$$

This formulation has the advantage to be invariant to a possible spatial transformation between both systems.

5.2 Navigation Filter

To fuse inertial sensor measurements with vision data, an inertial navigation filter has to be designed. This means incorporating the strapdown equations from section 3.1 and the sensor error model given in section 3.2. The objective is to aid the inertial navigation with relative egomotion data from the stereo camera system. While doing this, the inertial sensor biases – part of the sensor model – are estimated.

5.2.1 Error State Filter

One possibility for the optimal fusing of sensor data is the Kalman filter (see appendix B.3). Instead of filtering the total states, it is a common approach to filter error states. Wendel (2003) showed that the application of an error state filter has no disadvantages compared to a total state filter. On the contrary, the dynamic of the errors can often be described in a linear model rather than highly non-linear total state equations.

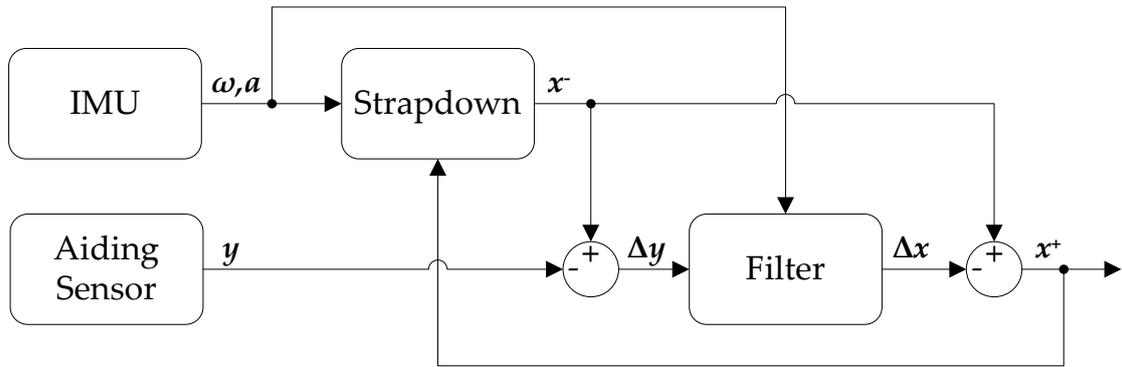


Figure 5.1: Error state spatial navigation filter design

Figure 5.1 shows the principal design of an error state spatial navigation filter. The inertial measurements are integrated in a strapdown manner and – although they are measurements – treated as known input values. This means they are only processed within the Kalman filter time update, with the error state predicted as zero. The error state covariance is propagated by adding the sensor noise through the system noise term. This is very efficient as the rather time-consuming measurement update is only applied for aiding measurements. Using the measurement \mathbf{y} and the a priori state \mathbf{x}^- an error measurement is calculated with $\Delta \mathbf{y} = \mathbf{x}^- - \mathbf{y}$. The estimated error state is used to correct the a priori state and reset to zero. According to Wendel (2003) the following error state vector is estimated:

$$\Delta \mathbf{x} = \left(\Delta \mathbf{a}_b^n, \Delta \mathbf{p}_b^n, \Delta \mathbf{v}^n, \Delta \mathbf{b}_\omega^b, \Delta \mathbf{b}_a^b \right)^T. \quad (5.2.1)$$

It includes the attitude error $\Delta \mathbf{a}$, the position error $\Delta \mathbf{p}$, and the velocity error $\Delta \mathbf{v}$. The angular rate bias error $\Delta \mathbf{b}_\omega$ and the acceleration bias error $\Delta \mathbf{b}_a$ are needed for a dynamic estimation of the inertial sensor bias terms.

5.2.2 State Transition Model

To begin, the continuous-time error states are derived, using the inertial navigation equations seen in section 3.1.

Attitude Error The change of the attitude error $\Delta\alpha$ is expressed according to Wendel (2003) with

$$\Delta\dot{\alpha}^n = -\omega_{in}^n \times \Delta\alpha - \Delta\omega_{in}^n + \hat{R}_b^n \Delta\omega^b, \quad (5.2.2)$$

where ω_{in}^n is the sum of the angular velocity of the earth and the transport rate. Again, this term is neglected as described in section 3.1.3, leaving

$$\Delta\dot{\alpha}^n \approx \hat{R}_b^n \Delta\omega^b. \quad (5.2.3)$$

According to the error model from eq. (3.2.1): $\omega^b = \hat{\omega}^b - \mathbf{b}_\omega + \mathbf{n}_\omega$, the angular velocity bias error depends from the measured angular velocity error with

$$\Delta\omega^b = \frac{\partial\omega^b}{\partial\mathbf{b}_\omega} \Delta\mathbf{b}_\omega + \mathbf{n}_\omega \quad (5.2.4)$$

$$= -\Delta\mathbf{b}_\omega + \mathbf{n}_\omega, \quad (5.2.5)$$

leading to

$$\Delta\dot{\alpha}^n \approx \hat{R}_b^n (-\Delta\mathbf{b}_\omega + \mathbf{n}_\omega). \quad (5.2.6)$$

Position Error The change of the position error is given directly by the velocity error with

$$\Delta\dot{\mathbf{p}}_b^n = \Delta\mathbf{v}^n. \quad (5.2.7)$$

Velocity Error With the simplified velocity differential equation from eq. (3.1.9)

$$\dot{\mathbf{v}}^n = R_b^n \mathbf{a}^b + \mathbf{g}^n, \quad (5.2.8)$$

the velocity error differential is given with

$$\Delta\dot{\mathbf{v}}^n = \frac{\partial\dot{\mathbf{v}}^n}{\partial\alpha} \Delta\alpha + \frac{\partial\dot{\mathbf{v}}^n}{\partial\mathbf{a}^b} \Delta\mathbf{a}^b. \quad (5.2.9)$$

Using the small angle approximation given in eq. (A.2.1) to relate the estimated attitude \hat{R}_b^n to the real attitude R_b^n with

$$R_b^n \approx \hat{R}_b^n (\mathbf{I} + [\Delta\alpha \times]), \quad (5.2.10)$$

leads to an approximation for the attitude error dependency with

$$\frac{\partial \dot{v}^n}{\partial \alpha} \Delta \alpha \approx -[\mathbf{a}^n \times] \Delta \alpha. \quad (5.2.11)$$

The dependency to the acceleration bias error is given with the sensor error model from eq. (3.2.1): $\mathbf{a}^b = \hat{\mathbf{a}}^b - \mathbf{b}_a + \mathbf{n}_a$ with

$$\frac{\partial \dot{v}^n}{\partial \mathbf{a}_b} \Delta \mathbf{a}^b \approx \mathbf{R}_b^n (-\Delta \mathbf{b}_a + \mathbf{n}_a). \quad (5.2.12)$$

Inertial Bias Error For the modeling of the inertial sensor bias, a random walk process which use the bias stabilities \mathbf{n}_{b_ω} and \mathbf{n}_{b_a} was chosen.

$$\dot{\mathbf{b}}_\omega = \mathbf{n}_{b_\omega} \quad (5.2.13)$$

$$\dot{\mathbf{b}}_a = \mathbf{n}_{b_a} \quad (5.2.14)$$

Using eq. (B.3.1), the above continuous-time error states are summarized as

$$\Delta \dot{\mathbf{x}} = \mathbf{F} \Delta \mathbf{x} + \mathbf{G} \mathbf{w}, \quad (5.2.15)$$

with the continuous-time system matrix \mathbf{F} , the noise transition matrix \mathbf{G} , and the inertial sensor noise $\mathbf{w} = (\mathbf{n}_\omega, \mathbf{n}_a, \mathbf{n}_{b_\omega}, \mathbf{n}_{b_a})^T$.

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & -\hat{\mathbf{R}}_b^n & 0 \\ 0 & 0 & \mathbf{I} & 0 & 0 \\ -[\mathbf{a}^n \times] & 0 & 0 & 0 & -\hat{\mathbf{R}}_b^n \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.2.16)$$

$$\mathbf{G} = \begin{bmatrix} \hat{\mathbf{R}}_b^n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \hat{\mathbf{R}}_b^n & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \quad (5.2.17)$$

According to Woodman (2007), the noise terms are given with $\mathbf{n} = RW/\sqrt{\delta t}$ and $\mathbf{n}_b = BS\sqrt{\delta t/\tau}$, with the device sampling period δt , random walk RW , and bias stability BS , derived from the Allan Variance as shown in section 3.2.1.

Using eq. (B.3.5), the discrete-time system is given with

$$\Delta \mathbf{x}_{k+1} = (\mathbf{I} + \mathbf{F} \Delta t) \Delta \mathbf{x}_k + \mathbf{G} \Delta t \mathbf{w}_k \quad (5.2.18)$$

$$= \Phi_k \Delta \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k. \quad (5.2.19)$$

Stochastic Cloning

A Kalman filter assumes that the measurement is only related to the current system state at time t_{k+1} . Since the camera egomotion is a relative measurement, which also depends on a previous state at time t_k , this turns out to be a problem. Roumeliotis and Burdick (2002) therefore propose a method called *stochastic cloning* to take correlations between the previous and current state into account. Cloning means augmenting the filter state at time t_k with

$$\Delta\check{\mathbf{x}}_k = \begin{pmatrix} \Delta\mathbf{x}_k \\ \Delta\mathbf{x}_k \end{pmatrix}. \quad (5.2.20)$$

As both states contain the same information, the covariance of the augmented system is

$$\check{\mathbf{P}}_k = \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_k \\ \mathbf{P}_k & \mathbf{P}_k \end{bmatrix}. \quad (5.2.21)$$

The discrete-time state propagation has to be rewritten to keep the cloned state $\Delta\mathbf{x}_k$ stationary.

$$\Delta\check{\mathbf{x}}_{k+1} = \check{\Phi}_k \Delta\check{\mathbf{x}}_k + \check{\mathbf{G}}_k \mathbf{w}_k \quad (5.2.22)$$

$$= \begin{bmatrix} \Phi_k & 0 \\ 0 & \mathbf{I} \end{bmatrix} \Delta\check{\mathbf{x}}_k + \begin{bmatrix} \mathbf{G}_k \\ 0 \end{bmatrix} \mathbf{w}_k \quad (5.2.23)$$

From eq. (B.3.14) the propagated covariance is given by

$$\check{\mathbf{P}}_{k+m} = \begin{bmatrix} \mathbf{P}_{k+m} & \mathbf{P}_k \mathcal{F}^T \\ \mathcal{F} \mathbf{P}_k & \mathbf{P}_k \end{bmatrix}, \quad (5.2.24)$$

with $\mathcal{F} = \prod_{i=0}^m \Phi_{k+i}$. The augmented state vector is kept small by only cloning the states showing a dependency to the relative measurement (Schmid et al., 2012).

5.2.3 Measurement Model

Aiding sensors are needed to correct the inertial strapdown solution, which degrades over time due to noise and uncompensated bias drifts. The measurement matrix \mathbf{H} projects the filter error-state $\Delta\mathbf{x}$ to an error-measurement $\Delta\mathbf{y}$.

Vision Measurement As given in section 4.3, the camera egomotion is a relative measurement $\hat{\mathbf{T}}_{cam}$. It relates the current pose \mathbf{T}_{k+1} to a previous pose \mathbf{T}_k .

$$\mathbf{T}_{cam} = \mathbf{T}_k^{-1} \mathbf{T}_{k+1} \quad (5.2.25)$$

The error measurement $\Delta \mathbf{y}_{cam} = \left(\Delta \hat{\boldsymbol{\alpha}}, \Delta \hat{\mathbf{p}} \right)^T$ is extracted from the relative pose error $\Delta \hat{\mathbf{T}}_{cam}$, found by

$$\Delta \hat{\mathbf{T}}_{cam} = \hat{\mathbf{T}}_{cam}^{-1} \mathbf{T}_{cam}. \quad (5.2.26)$$

Because the vision measurement only depends on attitude and position, the state vector is augmented at time t_k with $(\Delta \boldsymbol{\alpha}_k, \Delta \mathbf{p}_k)^T$. The error measurement is derived from eq. (5.2.25) with the partial derivatives of \mathbf{T}_{cam} w.r.t to the pose error for time $k+1$ and the augmented pose error at time k .

$$\Delta \mathbf{y}_{cam} = \frac{\partial \mathbf{T}_{cam}}{\partial (\boldsymbol{\alpha}, \mathbf{p})_k} \begin{pmatrix} \Delta \boldsymbol{\alpha} \\ \Delta \mathbf{p} \end{pmatrix}_k + \frac{\partial \mathbf{T}_{cam}}{\partial (\boldsymbol{\alpha}, \mathbf{p})_{k+1}} \begin{pmatrix} \Delta \boldsymbol{\alpha} \\ \Delta \mathbf{p} \end{pmatrix}_{k+1}. \quad (5.2.27)$$

This leads to the augmented measurement matrix $\check{\mathbf{H}}_{cam}$, which projects the augmented state to the error measurement.

$$\check{\mathbf{H}}_{cam} = \begin{bmatrix} \frac{\partial \mathbf{T}_{cam}}{\partial (\boldsymbol{\alpha}, \mathbf{p})_{k+1}} & \mathbf{0}_{6 \times 9} & \frac{\partial \mathbf{T}_{cam}}{\partial (\boldsymbol{\alpha}, \mathbf{p})_k} \end{bmatrix} \quad (5.2.28)$$

Inclination Measurement An inclination sensor is designed to measure $\hat{\mathbf{R}}_{xy}$, the absolute attitude w.r.t. the earth tangential plane without relevant bias terms (see section 3.3). With the attitude \mathbf{R}_b^n derived from the strapdown solution, the attitude error is calculated with

$$\Delta \mathbf{R} = \hat{\mathbf{R}}_{xy}^{-1} \mathbf{R}_b^n. \quad (5.2.29)$$

To derive the inclination measurement error $\Delta \mathbf{y}_{inc} = [\Delta \phi, \Delta \theta]^T$, the Euler angle decomposition for body fixed axes (see appendix A.2.1) is applied.

$$\Delta \phi = -\arctan \left(\frac{\Delta \mathbf{R}_{(1,2)}}{\Delta \mathbf{R}_{(2,2)}} \right) \quad (5.2.30)$$

$$\Delta \theta = -\arcsin \left(\Delta \mathbf{R}_{(0,2)} \right) \quad (5.2.31)$$

The augmented observation matrix $\check{\mathbf{H}}_{inc}$ projects the augmented filter state to the measurement error.

$$\check{\mathbf{H}}_{inc} = \begin{bmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 19} \end{bmatrix} \quad (5.2.32)$$

Chapter 6

System Integration and Application

In this chapter, I introduce a design for a heterogeneous multi-sensor system with a hardware concept as well as a software framework for real-time data handling and processing. The minimal configuration for this VINS consists of a stereo camera, an IMU, and an inclination sensor.

A solution based on a Field Programmable Gate Array (FPGA) is used to reference the asynchronous sensor measurements. Successive processing steps can now be applied on standard computing systems. A software framework allows for particular tasks to be encapsulated and provides standardized interfaces. It is appropriate for dealing with complex, hierarchic data flows to produce high level information from low level data. Altogether the system is designed in a way that facilitates both hardware and software extension.

I set up a prototype for the system to demonstrate the application of the proposed VINS with real data. Various pedestrian navigation tasks were used to test its capabilities, which are discussed below.

6.1 Hardware Concept

The heterogeneous character of the multi-sensor system delivers an optimal solution regarding space and energy constraints on the one hand and the computational demands on the other (Grießbach et al., 2012). For flexibility in various application requirements, different types of off-the-shelf FPGAs can be used. They range from small low-cost devices to more expensive and more powerful ones. An important requirement is system's flexibility for changes and extension.

During the design process, a hardware operation system (Krutz, 2006) has been used. It increases the potential re-usability of hardware modules by standardizing their interface descriptions. This makes the system design more dependable while easing development. Major interface standards are provided by custom-made add-on cards. These include low bandwidth interfaces, like SPI¹, CAN², RS-232, and digital inputs/outputs as well as high data-rate standards like CameraLink, Gigabit Ethernet, PCI Express, and USB for more sophisticated applications.

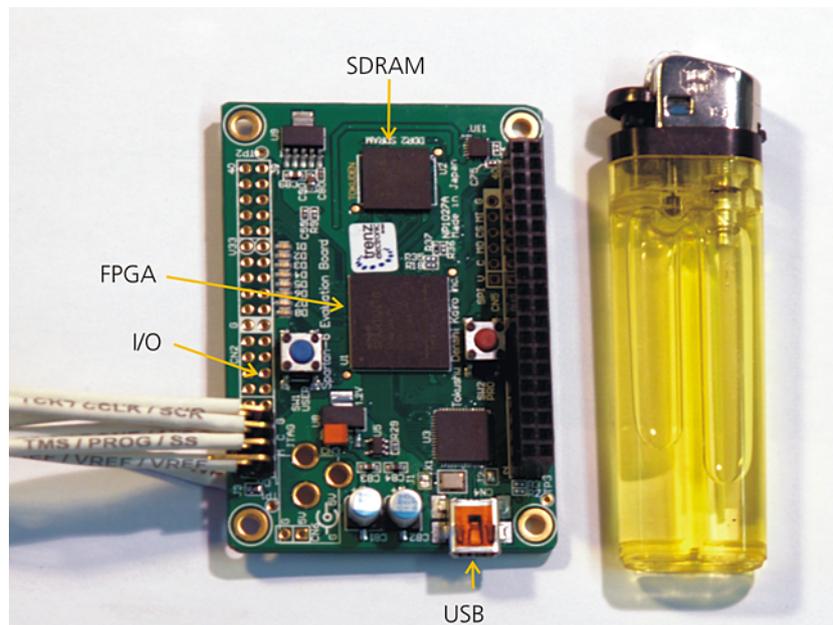


Figure 6.1: Spartan 6 FPGA board

Generally, each device in a multi-sensor system delivers its data to the capturing device asynchronously. To guarantee referenced sensor data, measurements have to be aligned along a common timeline, a necessary step for any further processing. In many cases, CPU-based or MPU³-based systems with standard or real-time operating systems are used, for which the timing behavior is either unknown or only defined by upper bounds.

The FPGA-based solution allows for a precise and deterministic referencing of all sensor data to a local timeline. A high-precision clock generates time stamps that are attached to all of the incoming measurements. Using external trigger signals is common practice for aligning external measurements that are not captured by the FPGA. The signals in this system are realized as either inputs

¹ Serial Peripheral Interface ² Controller Area Network ³ Micro Processor Unit

or outputs and each receives a time stamp. If a GPS signal is available, the clock signal of the built-in GPS receiver is used to refer the internal FPGA time to a global time scale. This allows for synchronization to many external systems also equipped with a GPS receiver.

The actual data processing is conducted on a CPU-based non-real-time operating system. The data handling is supported by a software framework, which allows to set up a data processing chain.

6.2 Software Framework

A hierarchical data processing chain is required for processing of the high level information from low level data, e.g. to get a navigation solution from raw sensor measurements. Ideally, particular tasks are encapsulated within a container with defined inputs and outputs. Connecting those containers, called *feeders* in the following, enables flexible and efficient data handling.

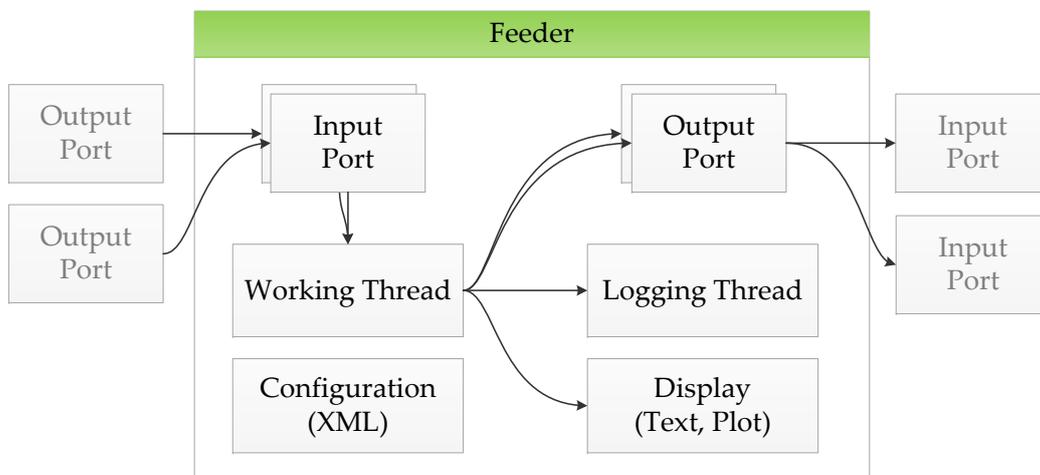


Figure 6.2: Feeder concept

A feeder, as shown in fig. 6.2, provides the base structure for data exchange, processing, and recording. It is also possible to monitor the processed data via text display and graphical output. Each feeder is configured by a standardized XML-node, which also contains individual settings. Because a feeder provides its own working thread, the workload of the processing chain is automatically distributed across a multi-core architecture. An additional logging thread is used to save the processed data to a binary file.

As stated above, synchronized data is a prerequisite for further data processing steps. Therefore, a structure containing a time stamp and an arbitrary data type is defined and used for data exchange between feeders. By using Runtime Type Information (RTTI), any type of data can be exchanged, e.g. an array of double values, images, or more complex types.

Each feeder can have multiple *output ports* for different data types. Succeeding feeders instantiate an *input port* that is connected to the output port of a preceding feeder. Multiple input ports per single output port provide for the sharing of data between different feeders. An input port is an adjustable FIFO⁴ buffer for holding the data. Buffer capabilities are important, because a working thread may not be ready to process new data immediately upon arrival.

The separation of different tasks to dedicated feeders aids in handling the complexity of a workflow. Complex tasks can be divided into subtasks and distributed within a feeder network. Loops provide even more possibilities to split the process. Figure 6.3 shows the realization of a feeder network for a VINS. It is divided up within a hardware part, including sensors, grabbing devices and a software part, containing middleware feeders and application feeders. The output may be used again for consecutive processing steps.

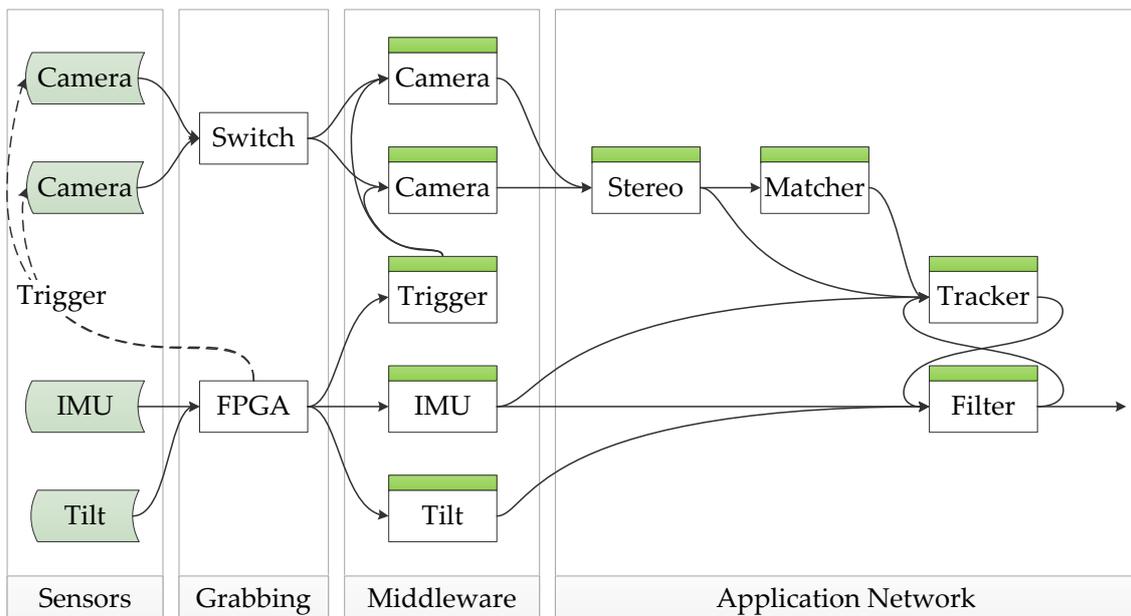


Figure 6.3: Feeder chain for a VINS

⁴ First In, First Out

6.2.1 Middleware

Middleware feeders are responsible for the communication with a hardware device. They grab data, preprocess it, feed the processing chain and save the received data as a binary file. For offline data processing, each middleware feeder is replaced by its file feeder counterpart. They load previously saved sensor data from the hard disk and feed it to the feeder network. Succeeding feeders are not affected.

Each sensor connected to the FPGA is fed into the processing chain by a FPGA feeder, which sets up communication channels for monitoring the connected sensor and receiving sensor data. The particular feeder knows the sensor protocol, allocates memory and fills the output data structure.

The camera feeder handles the communication with a particular camera. It configures the camera, e.g. controls the exposure settings, starts the image acquisition and grabs the images from the camera. Since image grabbing may not be delayed, a logging thread with additional buffer capacities is used for image saving. The camera is triggered by a FPGA-generated trigger event, which is counted and attached to the corresponding image. A trigger feeder controls the hardware trigger and also provides a counter with a corresponding time stamp. The time stamp is associated to the camera image by comparing the different frame counters, including the detection of lost frames.

6.2.2 Application Network

Application feeders are responsible for the actual data processing. Algorithm implementation is greatly simplified due to the fact that data grabbing and synchronization issues have already been dealt with.

The stereo feeder is connected to two camera feeders. Both image streams' individual time stamps are compared, which allows for the streams to be fused. Lost frames are even detected in the process. The feeder also contains the stereo registration information, which may be needed for successive feeders.

In accordance with section 4.3.1, distinctive feature points are extracted in the stereo reference camera. These points are matched to the second camera by using the epipolar constraint (see section 4.3.2). Preceding feeders are provided with a list of matched points for further processing steps.

In order to apply the pose estimation algorithm, corresponded feature points are found by inter-frame matching (see section 4.3.2). This process is supported by inertial measurements which are integrated using the dead reckoning algorithm shown in section 3.1. An additional input is provided to set the strapdown state from an external source, e.g. the filter feeder.

The filter feeder implements the final sensor data fusion as described in section 5.2. It receives measurements from IMU, inclination sensor, and stereo tracker to calculate an optimal 6-DOF motion estimation. After each filter cycle, the updated state is fed back into the tracker feeder to ensure the best possible feature prediction.

6.3 Experimental Results

A prototype sensor head as shown in fig. 6.4 was set up to evaluate the proposed VINS for accuracy, robustness, and repeatability. The head includes the FPGA with an interface board to connect the sensor system, a power supply unit, and a computing unit. The sensor system itself consists of the stereo cameras (section 4.5), the MEMS-IMU (tables 3.1 and 3.2), and an inclination sensor (table 3.3) mounted on an optical bench for stability. Additional near infra-red illumination is available for very dark environments.



Figure 6.4: Sensor head prototype

The system's capabilities were demonstrated on various pedestrian navigation tasks and are discussed here. A smoothed post-processed trajectory is used to assess the real-time navigation solution. To illustrate the benefits of fusing the input measurements, both inputs are also processed separately, giving the following trajectories:

- unaided inertial navigation (INS)
- visual odometry (VO)
- real-time navigation solution (VINS)
- post-processed ground truth (RTS)

6.3.1 Ground Truth Data

Lacking a real ground truth, each run is started and finished at a fixed reference point. This closed loop scenario is used to determine the accumulated position and rotation error for each run.

Furthermore, a verification dataset is created by including the closed loop information as an absolute position and heading angle measurement at the end of the run. To make this measurements known to the past filter states as well, a backward pass of the Kalman filter is needed. An efficient solution to this problem is the Rauch-Tung-Striebel (RTS) smoother, presented by Rauch et al., 1965. It uses the saved a priori and a posteriori filter estimates $\hat{\mathbf{x}}_k^-$, $\hat{\mathbf{x}}_k^+$ and its covariances \mathbf{P}_k^- , \mathbf{P}_k^+ from the Kalman filter forward pass to calculate smoothed state estimates:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^+ - \mathbf{K}_k \left(\hat{\mathbf{x}}_{k+1}^- - \hat{\mathbf{x}}_{k+1}^+ \right) \quad (6.3.1)$$

$$\mathbf{P}_k = \mathbf{P}_k^+ - \mathbf{K}_k \left(\mathbf{P}_{k+1}^- - \mathbf{P}_{k+1}^+ \right) \mathbf{K}_k^T, \quad (6.3.2)$$

with

$$\mathbf{K}_k = \mathbf{P}_k^+ \boldsymbol{\Phi}_k^T \left(\mathbf{P}_{k+1}^- \right)^{-1}. \quad (6.3.3)$$

6.3.2 System Initialization

The strapdown states as well as the Kalman filter states need to be initialized prior to each run. Since the navigation frame has an arbitrary origin, the position can be set to zero. The navigation frame is aligned to the LTP, which allows direct pitch and roll angle measurement by the inclination sensor. As it is rather difficult

and time consuming to determine north with the MEMS-IMU used, the heading direction is assumed to be zero. With the device at rest, the initial velocity is zero as well. Omitting the rotation rate of the earth, the angular rate biases are the direct measures of the gyroscopes. Leaving the acceleration biases, which are determined using eqs. (3.1.9) and (3.2.1) with

$$\mathbf{b}_a = \hat{\mathbf{a}}^b + \mathbf{R}_n^b \mathbf{g}^n. \quad (6.3.4)$$

Applying the small angle approximation, it can be seen that the limited accuracy of the inclination sensor directly effects the accuracy of the acceleration biases.

$$\Delta \mathbf{b}_a = [\Delta \boldsymbol{\alpha} \times] \mathbf{g}^n \quad (6.3.5)$$

If leveled, an inclination angle error of 0.1° leads to a bias error of about 1.7 mg for both horizontal acceleration axes.

Having finished this pre-initialization step, the Kalman filter is started. With the IMU roughly leveled to the LTP, the accuracy of the horizontal acceleration biases is only slightly improved, as can be seen in fig. 6.5a. However, the rotation of the device around its pitch and roll axes increases their observability, which results in greater acceleration bias accuracy. As a consequence, the precision of the measured gravitation vector – acting as an absolute vertical reference – increases. Figure 6.5b shows that this supports the horizontal angles quite well. The heading angle uncertainty is growing with time as it is not affected by the vertical reference.

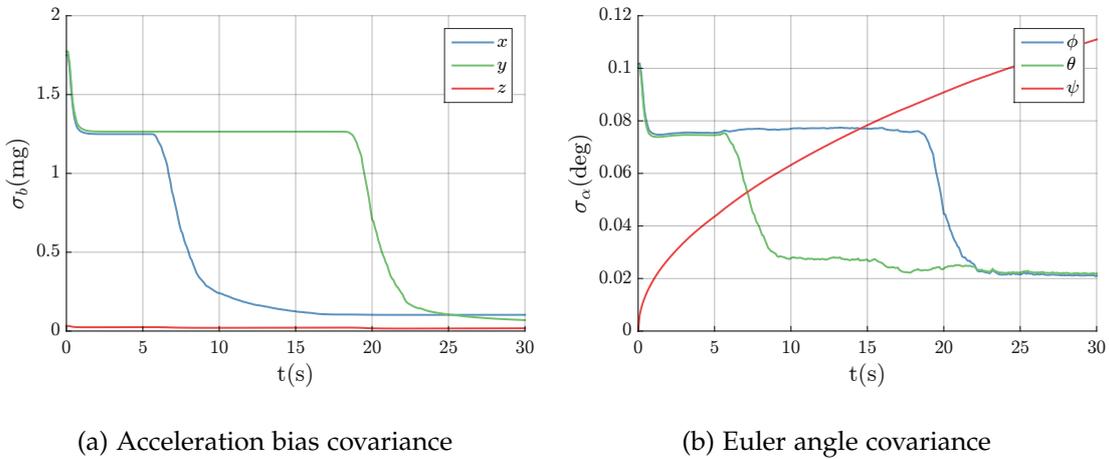


Figure 6.5: Initialization phase using the inclination sensor

Initializing the system without the inclination sensor is also feasible. In that case, the specified acceleration bias repeatability of 16 mg is used to initialize the bias covariance, as can be seen in fig. 6.6a. This results in a horizontal Euler angle accuracy of about 1°. By rotating the sensor head, the acceleration bias and Euler angle covariances are refined to the same level of accuracy as in the latter case. Using the inclination sensor will therefore only be advantageous if, for some reason, it is not possible to apply the initial roll and pitch rotations. It also turns out that its absolute accuracy of 0.1° is above the 0.02° reached by the initialization procedure and therefore not sufficient to serve as a reference for the filtering. After the filter states have run in, the measurement can be started.

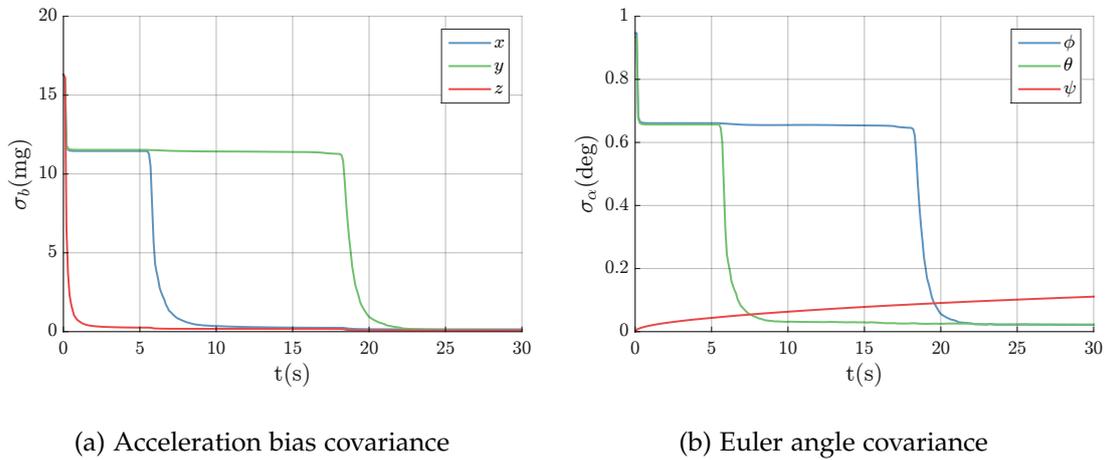


Figure 6.6: Initialization phase without inclination sensor

6.3.3 Experimental Data Sets

I chose three scenarios to demonstrate the capability of the proposed system, each challenging under different aspects: The first is a pedestrian indoor navigation task. Second, a combined indoor/outdoor run shows that the method is not restricted to indoor scenarios. The third task provides a data set to demonstrate the system's robustness against inconsistent aiding measurements.

All three runs began at the same starting point, determined manually on the floor plan overlay plots. A slight offset rotation was applied for the heading angle. This is to be expected as the produced trajectory is in reference how the device was initially positioned, which is not necessarily aligned with the map.

Indoor Environment

A typical office building was chosen for the indoor scenario. The run shown in figs. 6.7 and 6.8 covers four floors with an absolute path length of 310.8 m. This includes staircases, narrow corridors, and a wider entrance area with challenging light conditions; thus, the features varied greatly throughout the run. It was repeated ten times under real-life conditions, at normal walking speed of about 1.2 m/s, during working hours with people interfering with the visual measurement. To avoid systematic errors, the system was carried by several different people and initialized for each trial.



Figure 6.7: Indoor run floor plan overview

The unaided INS solution diverges quickly, resulting in an error of several hundred meters. Although the inertial sensor biases are determined in the initialization phase, an error in the bias estimation would cause the position error to grow quadratically with time, as can be seen from eq. (3.2.10). Even if it could have been estimated perfectly, white noise and bias instability causes random walk processes, making longer periods of unaided navigation prone to error.

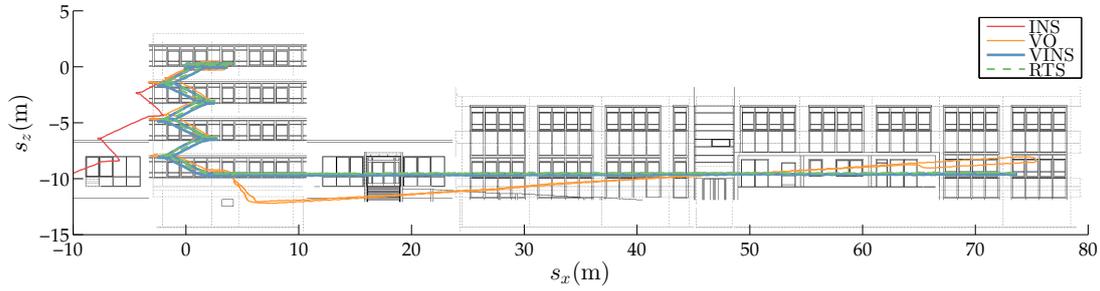


Figure 6.8: Indoor run building side view

The visual odometry performs better but suffers from unaided horizontal rotation axes as seen in fig. 6.9. Their increased uncertainty is propagated to the position, leading to a less accurate position. Dependent on the distance from the starting point, a small angular error may result in a large position error, reflected in the position uncertainty as shown in fig. 6.10.

Moreover, the integration only works for gapless visual data, which can usually not be guaranteed. Indeed, the VO for the majority of trials fails due to such gaps, caused by difficult light conditions or sections with very low texture. The absolute closed loop position Root Mean Square (RMS) error for the VO accumulates to 4.31 m. This corresponds to an absolute distance error of about 1.4% related to the total path length.

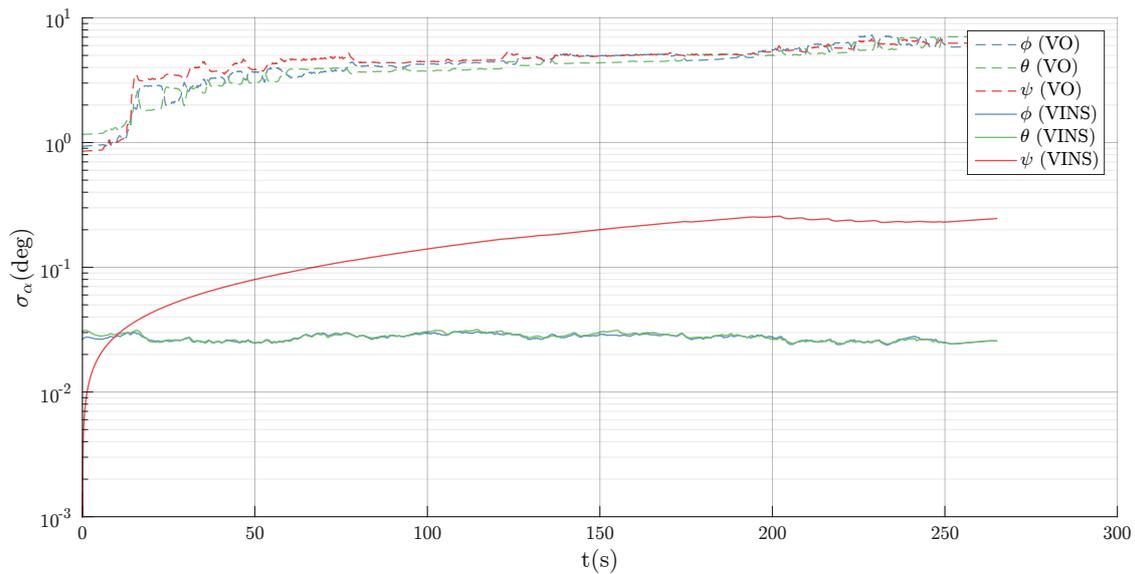


Figure 6.9: Indoor run Euler angle uncertainties

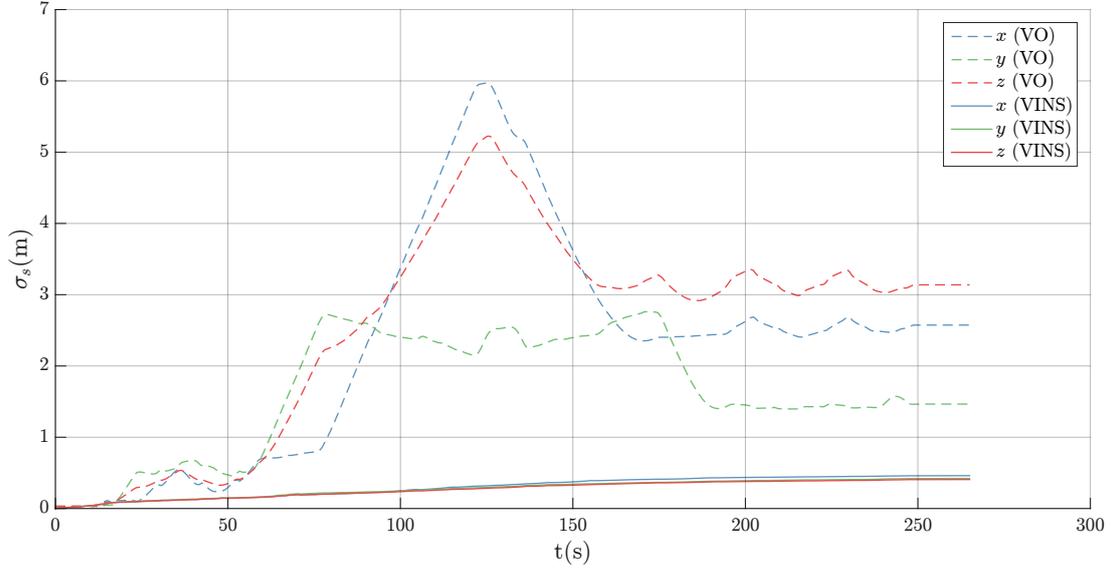


Figure 6.10: Indoor run position uncertainties

Altogether, the filtered navigation data shows the best performance. It uses the visual data to aid the inertial measurement, constantly adjusting the inertial biases. At the same time, the gravity measured acts as a vertical reference, stabilizing the horizontal rotation axes. In case the visual measurement fails due to bad light conditions or occlusions, the inertial navigation runs unaided, bridging the gaps.

The results of the ten independent trials are evaluated to get a statistical measure of the error distribution. Table 6.1 summarizes the closed loop errors of the averaged VINS with a 1σ uncertainty measure as well as the results of the VO and filtered VINS for the shown run.

	VO	VINS	avg. VINS
ϕ [°]	-2.49 ± 5.85	-0.03 ± 0.02	-0.03 ± 0.05
θ [°]	4.91 ± 7.07	0.01 ± 0.02	0.07 ± 0.06
ψ [°]	5.84 ± 6.28	-0.17 ± 0.27	-0.21 ± 0.86
s_x [m]	-0.66 ± 2.57	-0.35 ± 0.45	-0.32 ± 0.39
s_y [m]	-0.22 ± 1.47	-0.07 ± 0.43	0.08 ± 0.32
s_z [m]	-0.30 ± 3.14	-0.24 ± 0.41	0.36 ± 0.22

Table 6.1: Indoor run closed loop errors

While directly supported by the vertical reference, the error of the horizontal rotation axes of the VINS remains at the low level, reached by the initialization process. In comparison, the yaw angle accumulates a higher error with time when aided merely by relative visual measurements. The same is true for the absolute position RMS error, which reaches 0.74 m or 2.3 ‰ of the total path length for the given run. This equals a position random walk of about $2.7 \text{ m}/\sqrt{\text{h}}$. The filtered uncertainties are very close to the averaged results of the experimental data, showing the good consistency of the Kalman filter.

Combined Indoor/Outdoor Environment

The run shown in figs. 6.11 and 6.12 has a total path length of 410.9 m and includes two outdoor sections. Although the stereo system is optimized for a measuring range of 0.5 m to 10 m, the performance attained for the indoor run could be confirmed.

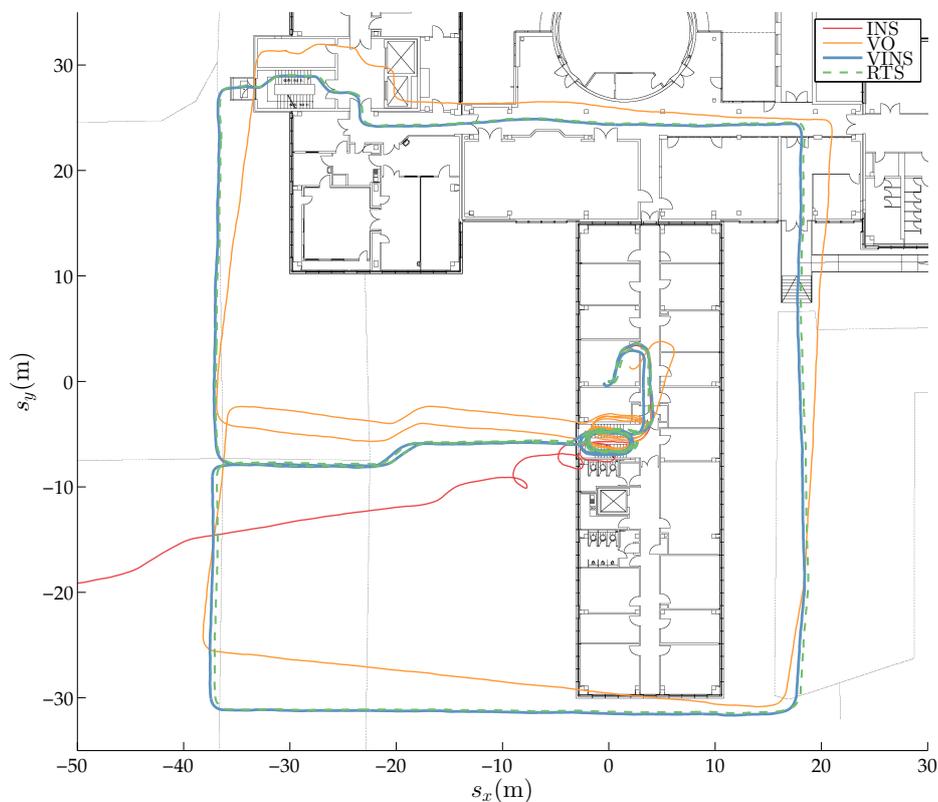


Figure 6.11: Indoor/outdoor run floor plan overview

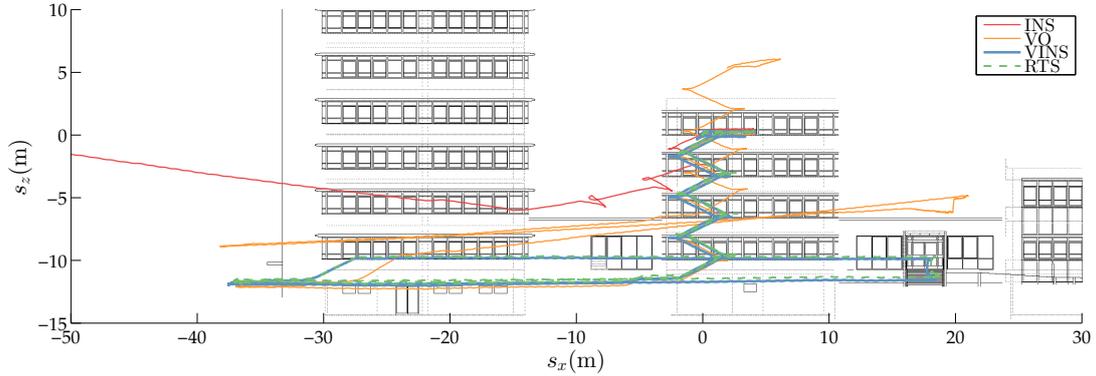


Figure 6.12: Indoor/outdoor run building side view

Even with a changed feature distribution, providing less features in the dedicated measuring range but an overall increased number of features, the visual measurement shows a consistent quality. The key for this rather unexpected behavior is the well textured floor in the outdoor sections, which delivers a sufficient number of close by features. Without those, the uncertainties of the relative position measurements from the stereo system would increase. In the long run, this leads to a decreased overall positioning accuracy. The uncertainty of the estimated relative rotations are not affected by the features distances. However, the possibility of integrating GNSS in outdoor environments should compensate for the potential limited accuracy of the visual measurements.

	VO	VINS	avg. VINS
ϕ [°]	-2.71 ± 4.44	-0.06 ± 0.02	0.00 ± 0.05
θ [°]	-5.52 ± 5.58	0.02 ± 0.02	0.03 ± 0.03
ψ [°]	12.22 ± 4.80	0.00 ± 0.27	-2.26 ± 1.52
s_x [m]	-1.44 ± 3.02	0.25 ± 0.37	0.21 ± 0.22
s_y [m]	1.97 ± 4.48	-0.45 ± 0.36	-0.13 ± 0.26
s_z [m]	5.32 ± 5.04	-0.34 ± 0.36	-0.44 ± 0.25

Table 6.2: Indoor/outdoor run closed loop errors

Table 6.2 shows the results for the VINS, averaged over ten trials, together with the closed loop errors of the VO and the filtered VINS for the shown run. An absolute position RMS error of 0.65 m or 1.6 ‰ of the total path length could be achieved. This equals a position random walk of about $2.1 \text{ m}/\sqrt{\text{h}}$.

Except from the heading angle, the predicted uncertainties from the Kalman filter are very close to the experimental data. The heading angle error reflects the systematic error introduced by omitting the Earth rotation of $15^\circ/\text{h}$ to simplify the state transition model of the Kalman filter. Further work will address this issue as shown in the next chapter.

Again, the VO solution could not be calculated in most cases due to occasional gaps in the visual data. For the shown run, a closed loop distance RMS error of 7.39 m, corresponding to an error of 1.8% of the total path length was achieved. Although it is not fully comparable, this is in accordance with some results shown for the *KITTI Vision Benchmark Suite* (Geiger et al., 2012). Unfortunately, their data sets do not provide inertial measurements. Thus, the slightly different setup with a increased stereo base could not be tested with the current approach.

Indoor Environment with Elevator

All of the incoming aiding measurements are checked for consistency with the inertial data. This is particularly important for the visual measurement as it may deliver misleading data. The effect is illustrated nicely in an elevator trial, as can be seen in fig. 6.13. While the inertial sensor measures the moving elevator, the visual data signals a resting observer.

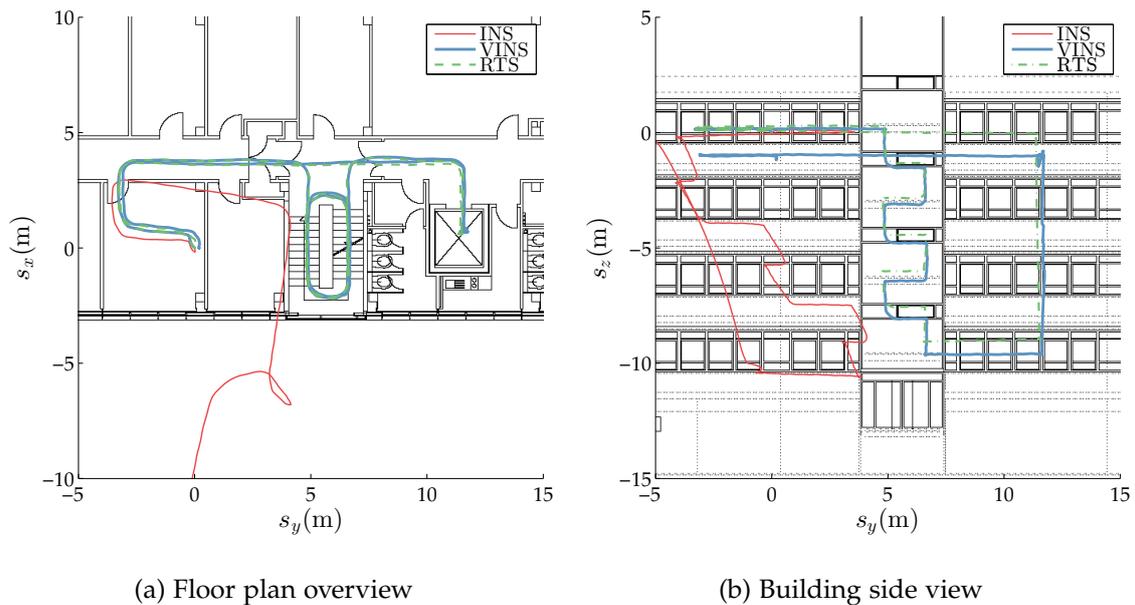


Figure 6.13: Elevator run trajectories

The inconsistent visual data is discarded within the Kalman filter, causing the velocity uncertainty to grow in a period of unaided inertial navigation (figs. 6.14 and 6.15). This is corrected with the next valid visual measurement when leaving the elevator. The accumulated height error shows the limits of the unaided inertial navigation.

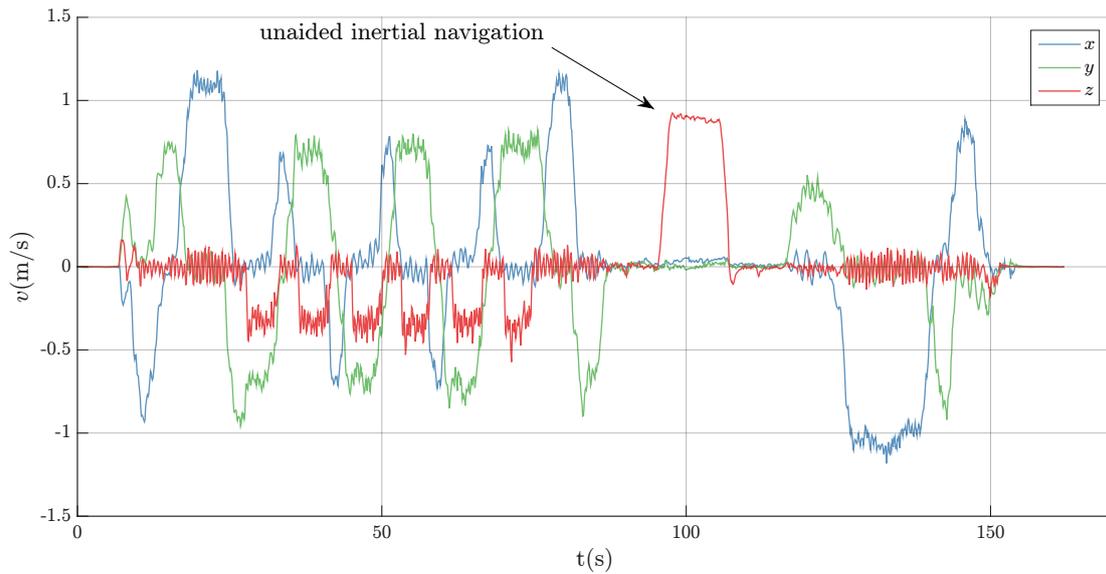


Figure 6.14: Elevator run velocity

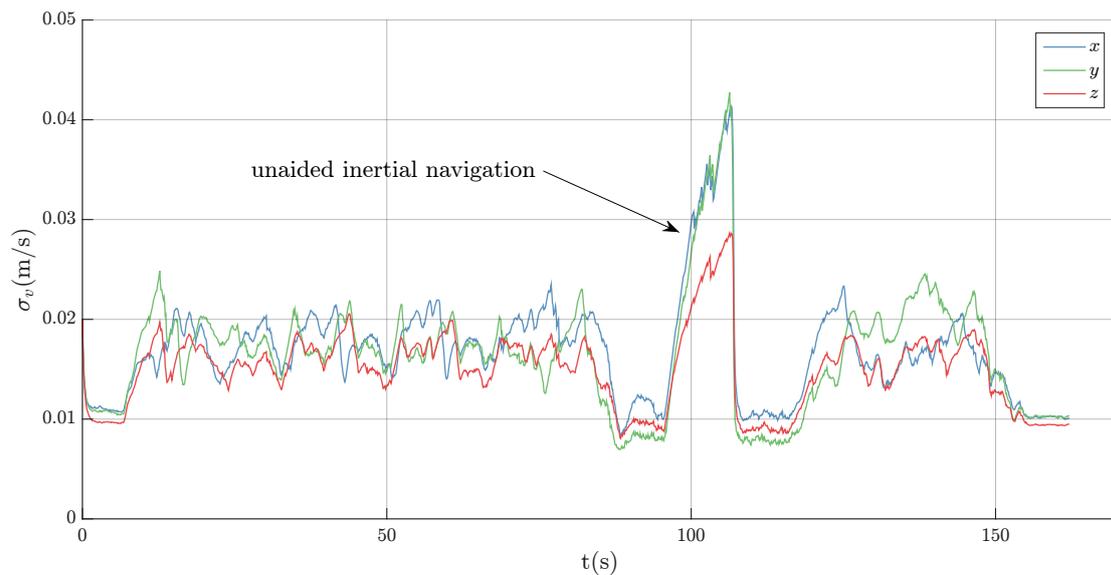


Figure 6.15: Elevator run velocity uncertainties

Chapter 7

Conclusion and Future Work

7.1 Summary

In this publication, I have investigated the use of low-cost inertial navigation aided by stereo vision to provide local navigation in unknown environments. The system created uses neither active measurements nor external references or a priori assumptions about the environment.

The proposed method takes advantage of the complementary properties of inertial sensors and stereo-vision measurements. Visual data aids the inertial navigation solution, while the inertial measurements are used to constrain the correspondence problem and track feature points simultaneously. Possible issues and solutions with respect to synchronization and registration of the sensor systems have been discussed above. I lay out a hardware concept as well as a software framework that allows for real-time data processing.

I give special emphasis to modeling of the stereo-vision system and present a novel calibration method for optical systems, which uses a holographic pattern. This method allows for accurate, fast, and reliable camera calibration. It is an important prerequisite for the successive overall error modeling, which derives uncertainties throughout the whole data processing chain.

This local navigation solution is obtained by fusing inertial measurements and vision data within an error state Kalman filter, incorporating an error model for the inertial sensor. It has been shown that the complementary character of both measurements support one another quite well. The stability and accuracy was significantly increased compared to the original sensor data.

To demonstrate the capability of the proposed method, a prototype sensor head has been set up. The complete data processing chain was evaluated with an indoor navigation task as well as for a combined indoor and outdoor environment. Although the quality of the visual measurement strongly depends on the given environment, an overall accuracy of about $2\text{ m}/\sqrt{h}$ could be achieved. This equals 1-2 ‰ of the accumulated path length for the given trial runs.

The presented results are accepted for publication at the 5th International Conference on Indoor Positioning and Indoor Navigation (Grießbach et al., 2014).

Meanwhile, the presented method already plays an important role for various applications, e.g. for the assistance of industrial inspection tasks. The state of the art practice provides printed maps for the inspection team, which are used to take notes on damages, positions of measurements, or photographs. These informations have to be transferred to a data base in an expensive and error prone post-processing step. With the indoor navigation solution, the trajectory of the inspector is always known. Each photograph of a camera, which is rigidly attached to the navigation device, is spatially referenced and may later be easily related to a digital model. Another application under way is the support for large scale indoor surveyings with multiple laser scans. This actually is completing the circle as a similar task has once been the starting point for this work.

7.2 Future Work

The design of a vision-aided inertial navigation system involves many different areas of expertise, including computer vision, inertial navigation, and multi sensor fusion. My work outlined here, provides a baseline in each area and will be subject to further investigations and improvements.

- Currently, the visual odometry is derived from immediately adjacent images combined with the cadence of the camera update rate. If the observer is not moving or only slowly rotating, the observed scene will not significantly change. Thus, additional odometry could be achieved with a *key frame* that is updated at a lower rate. This method would correct for short-term accumulated noise.

- A more advanced approach would be to include real-time *loop closing* to correct for long-term accumulated drift. Using SfM for offline post-processing is also a conceivable solution.
- Potential for further work lies in feature extraction; computational costs may be avoided with different types of extractors. Moreover, the quality of the VO depends on the distribution of features within the image. Artificial environments often exhibit poorly distributed features, a situation that could be countered by partitioning the image. This would open up the possibility of handling low texture areas differently than areas with a lot of texture.
- The current implementation provides local navigation without true north direction. Static methods to find true north hardly work with the MEMS-IMU employed here. As a consequence, the earth's rotation has to be neglected, which is reasonable for short-term navigation. To avoid systematic errors for long-term observations, dynamic north finding methods that use visual measurements have to be investigated.
- Additional effort should be put into the modeling and calibration of the MEMS inertial sensors. This would increase the ability to navigate independently from the visual measurements, which may fail in difficult light conditions. It would also enable the use of even lower quality sensors as those used in many consumer grade devices.
- Finally, the filter design has to be extended to include GNSS measurements for global navigation. This would open the door for outdoor applications where GNSS drop-outs have to be bridged, e.g. in urban or forested areas.
- Additional work could be done regarding system applications. Dense depth maps that are created from the stereo images can be used in conjunction with the navigation solution as an input for 3D modeling of indoor environments. Furthermore, guidance can be provided based on positioning and environmental analysis.

Appendix A

Coordinates Transformation

A Cartesian coordinate $v \in \mathbb{R}^3$, given in the reference frame, can be transformed to another frame by applying

$$v' = Rv + t.$$

The orientation of the reference frame w.r.t. the destination coordinate frame is described by a translation t and a rotation matrix R which can be parametrized, e.g. by a unit quaternion or Euler angles (see appendix A.2).

In accordance with appendix B.1, partial derivatives of v' to map uncertainties in v, t are as follows:

$$\begin{aligned}\frac{\partial v'}{\partial v} &= R \\ \frac{\partial v'}{\partial t} &= I_3 \\ \frac{\partial v'}{\partial x} &= \frac{\partial R}{\partial x} v.\end{aligned}$$

The partial derivative of $R = f(x)$ depends on the parametrization of the rotation and is shown in detail in appendices A.2.1 and A.2.3.

A.1 Homogeneous Coordinates Transformation

For homogeneous coordinates $v \in \mathbb{P}^3$, the transformation is given with

$$v' = Tv,$$

with

$$T = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}.$$

The inverse transformation is defined to be

$$T^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{t} \\ 0 & 1 \end{bmatrix}.$$

A.2 Attitude Representations

A matrix is called a rotation matrix if it preserves the length of a vector and the angles between vectors. They are orthogonal with determinant 1, yielding

$$\mathbf{R}^{-1} = \mathbf{R}^T.$$

A.2.1 Euler Angle

Euler angles in the $x - y - z$ convention, also referred to as Tait-Bryan angles or nautical angles (roll, pitch, yaw), are used to parametrize a counter-clockwise rotation (mathematically positive) about the particular axis with the rotation matrices

$$\begin{aligned} \mathbf{R}_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \\ \mathbf{R}_y &= \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \\ \mathbf{R}_z &= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Euler Angle to Rotation Matrix

The combined rotation $\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$ describes a rotation about body-fixed principal axes.

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) \cos(\psi) & -\cos(\theta) \sin(\psi) & \sin(\theta) \\ \cos(\phi) \sin(\psi) + \sin(\phi) \sin(\theta) \cos(\psi) & \cos(\phi) \cos(\psi) - \sin(\phi) \sin(\theta) \sin(\psi) & -\sin(\phi) \cos(\theta) \\ \sin(\phi) \sin(\psi) - \cos(\phi) \sin(\theta) \cos(\psi) & \sin(\phi) \cos(\psi) + \cos(\phi) \sin(\theta) \sin(\psi) & \cos(\phi) \cos(\theta) \end{bmatrix}$$

whereas $\mathbf{R} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x$ represents a rotation about space-fixed axes.

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) \cos(\psi) & -\cos(\phi) \sin(\psi) + \sin(\phi) \sin(\theta) \cos(\psi) & \sin(\phi) \sin(\psi) + \cos(\phi) \sin(\theta) \cos(\psi) \\ \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) + \sin(\phi) \sin(\theta) \sin(\psi) & -\sin(\phi) \cos(\psi) + \cos(\phi) \sin(\theta) \sin(\psi) \\ -\sin(\theta) & \sin(\phi) \cos(\theta) & \cos(\phi) \cos(\theta) \end{bmatrix}$$

Partial derivatives w.r.t. Euler angles are

$$\begin{aligned} \frac{\partial \mathbf{R}_x}{\partial \phi} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\sin(\phi) & -\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \end{bmatrix}, \\ \frac{\partial \mathbf{R}_y}{\partial \theta} &= \begin{bmatrix} -\sin(\theta) & 0 & \cos(\theta) \\ 0 & 1 & 0 \\ -\cos(\theta) & 0 & -\sin(\theta) \end{bmatrix}, \\ \frac{\partial \mathbf{R}_z}{\partial \psi} &= \begin{bmatrix} -\sin(\psi) & -\cos(\psi) & 0 \\ \cos(\psi) & -\sin(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Combined rotations, e.g. for body-fixed axes w.r.t. ϕ , are derived as follows:

$$\frac{\partial \mathbf{R}}{\partial \phi} = \frac{\partial \mathbf{R}_x}{\partial \phi} \mathbf{R}_y \mathbf{R}_z.$$

The rotation matrix can be approximation for small angles $\boldsymbol{\alpha} = [\phi, \theta, \psi]$ with

$$\mathbf{R} \approx \mathbf{I} + [\boldsymbol{\alpha} \times] = \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix}. \quad (\text{A.2.1})$$

Rotation Matrix to Euler Angle

With a rotation matrix

$$\mathbf{R} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix},$$

the factorization of a body-fixed rotation is given with

$$\phi = -\arctan\left(\frac{a_{12}}{a_{22}}\right),$$

$$\theta = \arcsin(a_{02}),$$

$$\psi = \arctan\left(\frac{a_{01}}{a_{00}}\right).$$

Space-fixed rotations are factorized as follows:

$$\phi = \arctan\left(\frac{a_{21}}{a_{22}}\right)$$

$$\theta = -\arcsin(a_{20})$$

$$\psi = \arctan\left(\frac{a_{10}}{a_{00}}\right)$$

A.2.2 Quaternion

Quaternions were first introduced in Hamilton (1844-1850, 1853). Good reads for their practical use can be found in Shoemaker (1994), Dam et al. (1998), Vicci (2001), and Eberly (2002). The general quaternion is defined as follows:

$$\mathbf{q} \equiv (s, \mathbf{v})^T, \quad s \in \mathbb{R}, \mathbf{v} \in \mathbb{R}^3$$

Multiplication Let $\mathbf{q}' = (s', \mathbf{v}')^T$, then

$$\begin{aligned} \mathbf{q} \circ \mathbf{q}' &= (ss' - \mathbf{v} \cdot \mathbf{v}', \mathbf{v} \times \mathbf{v}' + s\mathbf{v}' + s'\mathbf{v})^T \\ &= \begin{bmatrix} s & -v_x & -v_y & -v_z \\ v_x & s & -v_z & v_y \\ v_y & v_z & s & -v_x \\ v_z & -v_y & v_x & s \end{bmatrix} \begin{pmatrix} s' \\ v'_x \\ v'_y \\ v'_z \end{pmatrix}. \end{aligned}$$

Conjugate

$$\mathbf{q}^* \equiv (s, -\mathbf{v})^T$$

Norm

$$\|\mathbf{q}\| = \sqrt{s^2 + \mathbf{v} \cdot \mathbf{v}}$$

Inverse

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2}$$

A.2.3 Unit Quaternion

Unit quaternions are a subset of quaternions satisfying the condition $\|\mathbf{q}\| = 1$ and thus $\mathbf{q}^{-1} = \mathbf{q}^*$. A rotation is represented by an angle 2Θ about an axis $\mathbf{u} \in \mathbb{R}^3$ with

$$\mathbf{q} = (\cos(\Theta), \mathbf{u} \sin(\Theta))^T.$$

If $\mathbf{q} = (1, \mathbf{0})^T$, then $\Theta = 0$ and \mathbf{u} can be an arbitrary chosen vector. For general rotations in \mathbb{R}^3 , a vector quaternion $\mathbf{q}_v = (0, \mathbf{v})^T$ is defined and rotated as follows:

$$\mathbf{q}'_v = (0, \mathbf{v}')^T = \mathbf{q} \circ \mathbf{q}_v \circ \mathbf{q}^*$$

Successive rotations about \mathbf{q}_1 and \mathbf{q}_2 are done by

$$\begin{aligned} \mathbf{q}'_v &= \mathbf{q}_2 \circ (\mathbf{q}_1 \circ \mathbf{q}_v \circ \mathbf{q}_1^*) \circ \mathbf{q}_2^* \\ &= (\mathbf{q}_2 \circ \mathbf{q}_1) \circ \mathbf{q}_v \circ (\mathbf{q}_1^* \circ \mathbf{q}_2^*) \\ &= \mathbf{q}_{21} \circ \mathbf{q}_v \circ \mathbf{q}_{21}^*. \end{aligned}$$

Unit Quaternion to Rotation Matrix

$$\mathbf{R} = \begin{bmatrix} s^2 + v_x^2 - v_y^2 - v_z^2 & 2(v_x v_y - s v_z) & 2(v_x v_z + s v_y) \\ 2(s v_z + v_x v_y) & s^2 - v_x^2 + v_y^2 - v_z^2 & 2(v_y v_z - s v_x) \\ 2(v_x v_z - s v_y) & 2(s v_x + v_y v_z) & s^2 - v_x^2 - v_y^2 + v_z^2 \end{bmatrix}$$

Partial derivatives w.r.t. the unit quaternion are

$$\begin{aligned}\frac{\partial \mathbf{R}}{\partial s} &= 2 \begin{bmatrix} s & -v_z & v_y \\ v_z & s & -v_x \\ -v_y & v_x & s \end{bmatrix}, \\ \frac{\partial \mathbf{R}}{\partial v_x} &= 2 \begin{bmatrix} v_x & v_y & v_z \\ v_y & -v_x & -s \\ v_z & s & -v_x \end{bmatrix}, \\ \frac{\partial \mathbf{R}}{\partial v_y} &= 2 \begin{bmatrix} -v_y & v_x & s \\ v_x & v_y & v_z \\ -s & v_z & -v_y \end{bmatrix}, \\ \frac{\partial \mathbf{R}}{\partial v_z} &= 2 \begin{bmatrix} -v_z & -s & v_x \\ s & -v_z & v_y \\ v_x & v_y & v_z \end{bmatrix}.\end{aligned}$$

Rotation Matrix to Unit Quaternion

To extract a unit quaternion, the trace of the rotation matrix shown in appendix A.2.3 is used. If the trace is greater than zero, it can be seen that $4v_x^2 = 1 + a_{00} + a_{11} + a_{22}$ and therefore

$$\begin{aligned}s &= \sqrt{1 + a_{00} + a_{11} + a_{22}}/2, \\ v_x &= (a_{21} - a_{12})/4s, \\ v_y &= (a_{02} - a_{20})/4s, \\ v_z &= (a_{10} - a_{01})/4s.\end{aligned}$$

For a trace less or equal to zero, the largest component of the quaternion vector part v is extracted first. If a_{00} is the maximum diagonal value, then v_x is the largest component and $4v_x^2 = 1 + a_{00} - a_{11} - a_{22}$, therefore

$$\begin{aligned}v_x &= \sqrt{1 + a_{00} - a_{11} - a_{22}}/2, \\ s &= (a_{21} - a_{12})/4v_x, \\ v_y &= (a_{01} + a_{10})/4v_x, \\ v_z &= (a_{02} + a_{20})/4v_x.\end{aligned}$$

With a_{11} being the maximum diagonal value, $4v_y^2 = 1 + a_{11} - a_{00} - a_{22}$ and therefore

$$\begin{aligned} v_y &= \sqrt{1 + a_{11} - a_{00} - a_{22}}/2, \\ s &= (a_{02} - a_{20})/4v_y, \\ v_x &= (a_{01} + a_{10})/4v_y, \\ v_z &= (a_{12} + a_{21})/4v_y. \end{aligned}$$

For a_{22} being the maximum diagonal value, $4v_z^2 = 1 + a_{22} - a_{00} - a_{11}$ and therefore

$$\begin{aligned} v_z &= \sqrt{1 + a_{22} - a_{00} - a_{11}}/2, \\ s &= (a_{10} - a_{01})/4v_z, \\ v_x &= (a_{02} + a_{20})/4v_z, \\ v_y &= (a_{12} + a_{21})/4v_z. \end{aligned}$$

Unit Quaternion to Euler Angle

Euler angles for body-fixed rotations are extracted with

$$\begin{aligned} \phi &= \arctan \left(\frac{2(v_y v_z - s v_x)}{s^2 - v_x^2 - v_y^2 + v_z^2} \right), \\ \theta &= \arcsin (2(v_x v_z + s v_y)), \\ \psi &= \arctan \left(\frac{2(v_x v_y - s v_z)}{s^2 + v_x^2 - v_y^2 - v_z^2} \right). \end{aligned}$$

Space-fixed rotations are factorized as follows:

$$\begin{aligned} \phi &= \arctan \left(\frac{2(s v_x + v_y v_z)}{s^2 - v_x^2 - v_y^2 + v_z^2} \right), \\ \theta &= \arcsin (2(v_x v_z - s v_y)), \\ \psi &= \arctan \left(\frac{2(s v_z + v_x v_y)}{s^2 + v_x^2 - v_y^2 - v_z^2} \right). \end{aligned}$$

Euler Angle to Unit Quaternion

Euler angles for body-fixed rotations are transformed as follows:

$$q = \begin{pmatrix} \cos(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\psi}{2}) - \sin(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\psi}{2}) + \sin(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\psi}{2}) - \sin(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\psi}{2}) + \sin(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\psi}{2}) \end{pmatrix},$$

whereas Euler angles for space-fixed rotations are given as

$$q = \begin{pmatrix} \cos(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\psi}{2}) + \sin(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\psi}{2}) \\ \sin(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\psi}{2}) - \cos(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\psi}{2}) + \sin(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\psi}{2}) - \sin(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\psi}{2}) \end{pmatrix}.$$

Appendix B

Uncertainty Calculus

B.1 Uncertainty Mapping

B.1.1 Linear Equations

Having a linear equation system with $y = Ax$ and uncertainties in x denoted by

$$\Sigma_x = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots \\ \sigma_{21} & \sigma_2^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad (\text{B.1.1})$$

uncertainties in y are given by

$$\Sigma_y = A\Sigma_x A^T. \quad (\text{B.1.2})$$

B.1.2 Non-Linear Single Valued Equations

A non-linear equation $y = f(x)$ with $x \in \mathbb{R}^1, y \in \mathbb{R}^1$ is linearized with a Taylor series about the mean x_0 .

$$y = f(x_0) + \left. \frac{\partial f}{\partial x} \right|_{x_0} (x - x_0) + \frac{1}{2} \left. \frac{\partial^2 f}{\partial x^2} \right|_{x_0} (x - x_0)^2 + \dots \quad (\text{B.1.3})$$

$$y = a_0 + a_1(x - x_0) + \frac{1}{2}a_2(x - x_0)^2 + \dots \quad (\text{B.1.4})$$

The mean x_0 is defined by the expectation value $E[X]$ with

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx, \quad (\text{B.1.5})$$

where $f_X(x)$ is the Probability Density function (PDF) of X . The second central moment is the variance, which is given with

$$\begin{aligned}\text{Var}(X) &= E[(x - x_0)^2] \\ &= E[x^2] - 2x_0E[x] + E[x]^2 \\ &= E[x^2] - E[x]^2.\end{aligned}\tag{B.1.6}$$

Its square root is the standard deviation σ .

First Order Approximation For a Taylor series (see eq. (B.1.4)) truncated after the first term the following approximations for mean and variation hold (Mekid and Vaja, 2008).

$$\begin{aligned}y_0 &= E[y] \\ &= a_0 + a_1E[(x - x_0)] \\ &= a_0 + a_1E[x] - a_1x_0 \\ &= a_0\end{aligned}\tag{B.1.7}$$

$$\begin{aligned}\sigma_y^2 &= E[(y - y_0)^2] \\ &= E\left[(a_0 + a_1(x - x_0) - a_0)^2\right] \\ &= a_1^2E\left[(x - x_0)^2\right] \\ &= a_1^2\sigma_x^2\end{aligned}\tag{B.1.8}$$

B.1.3 Non-Linear Vector Valued Equations

A more general Taylor series formulation for the non-linear equation $\mathbf{y} = f(X)$ with $X \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$ is found with

$$\begin{aligned}\mathbf{y} &= f(\mathbf{x}_0) + \sum_{i=1}^n \left. \frac{\partial f}{\partial x_i} \right|_{x_{0,i}} (\mathbf{x} - x_{0,i}) + \\ &\quad \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left. \frac{\partial^2 f}{\partial x_i \partial x_j} \right|_{x_{0,i}, x_{0,j}} (\mathbf{x} - x_{0,i})(\mathbf{x} - x_{0,j}) + \dots\end{aligned}\tag{B.1.9}$$

First Order Approximation Arras (1998) showed the derivation of mean and variance for first order Taylor series truncation.

$$\mathbf{y}_0 = f(\mathbf{x}_0) \quad (\text{B.1.10})$$

$$\Sigma_{\mathbf{y}} = \mathbf{J}\Sigma_{\mathbf{x}}\mathbf{J}^T, \quad (\text{B.1.11})$$

with covariance matrices Σ (B.1.1) and Jacobian matrix \mathbf{J} with $J_{ij} = \partial f_i / \partial x_j$ defining the partial derivatives of $f(\mathbf{x})$ w.r.t. \mathbf{x} .

B.2 Least Square Solutions

B.2.1 Linear Equations

Linear equations of the form $\mathbf{M}\mathbf{y} = \mathbf{b}$ are solved by minimizing the sum of squared residual errors.

$$\min_{\mathbf{y}} = \|\mathbf{b} - \mathbf{M}\mathbf{y}\|^2 \quad (\text{B.2.1})$$

A solution is given with

$$\mathbf{y} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{b}. \quad (\text{B.2.2})$$

With covariances Σ_b for errors in \mathbf{b} and Σ_x for errors in \mathbf{M} , the covariance of the solution is given with

$$\Sigma_{\mathbf{y}} = \mathbf{J}_b\Sigma_b\mathbf{J}_b^T + \mathbf{J}_x\Sigma_x\mathbf{J}_x^T, \quad (\text{B.2.3})$$

with the Jacobian of f w.r.t. $\mathbf{b} = f(\mathbf{z})$

$$\mathbf{J}_b = \frac{\partial \mathbf{y}}{\partial \mathbf{z}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T \frac{\partial \mathbf{b}}{\partial \mathbf{z}} \quad (\text{B.2.4})$$

and the Jacobian of f w.r.t. $\mathbf{M} = f(\mathbf{x})$.

$$\mathbf{J}_x = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial (\mathbf{M}^T\mathbf{M})^{-1}}{\partial \mathbf{x}} \mathbf{M}^T\mathbf{b} + \mathbf{A}^{-1} \frac{\partial \mathbf{M}^T}{\partial \mathbf{x}} \mathbf{b} \quad (\text{B.2.5})$$

This is simplified using the differentiation rule

$$\frac{\partial \mathbf{A}^{-1}}{\partial \mathbf{x}} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \mathbf{x}} \mathbf{A}^{-1} \quad (\text{B.2.6})$$

and the property of idempotent matrices that $\mathbf{B} = \mathbf{B}^2$ with

$$\mathbf{B} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \quad (\text{B.2.7})$$

yielding

$$J_x = -(M^T M)^{-1} M^T \frac{\partial M}{\partial x} \mathbf{y}. \quad (\text{B.2.8})$$

B.2.2 Non-Linear Equations

A set of m non-linear equations

$$b_i = f(x_i, \mathbf{y}) \quad i = 1, \dots, m \quad (\text{B.2.9})$$

with unknowns in \mathbf{y} is solved by minimizing the sum of squared residual errors.

$$\min_{\mathbf{y}} = \|\mathbf{b} - f(\mathbf{x}, \mathbf{y})\|^2 \quad (\text{B.2.10})$$

Starting with an initial value \mathbf{y}_0 , the solution is iteratively refined with

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \Delta \mathbf{y}_k. \quad (\text{B.2.11})$$

The shift vector $\Delta \mathbf{y}$ is found similar to equation (B.2.2) with

$$\Delta \mathbf{y} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \Delta \mathbf{b}, \quad (\text{B.2.12})$$

with $\Delta \mathbf{b} = \mathbf{b} - f(\mathbf{x}, \mathbf{y})$ and the Jacobian

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f(x_1, \mathbf{y})}{\partial y_1} & \frac{\partial f(x_1, \mathbf{y})}{\partial y_2} & \dots & \frac{\partial f(x_1, \mathbf{y})}{\partial y_n} \\ \frac{\partial f(x_2, \mathbf{y})}{\partial y_1} & \frac{\partial f(x_2, \mathbf{y})}{\partial y_2} & \dots & \frac{\partial f(x_2, \mathbf{y})}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(x_m, \mathbf{y})}{\partial y_1} & \frac{\partial f(x_m, \mathbf{y})}{\partial y_2} & \dots & \frac{\partial f(x_m, \mathbf{y})}{\partial y_n} \end{bmatrix}, \quad (\text{B.2.13})$$

which is the linearization of $f(\mathbf{x}, \mathbf{y})$ w.r.t. the n unknowns in \mathbf{y} . As this method is a gradient-based local optimization, the choice of a good initial value, which is close to the true value is important. Otherwise the method can diverge or end up in a secondary minimum.

After the iteration has finished, the covariance of the solution $\Sigma_{\mathbf{y}}$ is calculated analogous to eq. (B.2.3).

$$\mathbf{J}_b = \frac{\partial \Delta \mathbf{y}}{\partial \mathbf{b}} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \frac{\partial \Delta \mathbf{b}}{\partial \mathbf{b}} \quad (\text{B.2.14})$$

$$\mathbf{J}_x = \frac{\partial \Delta \mathbf{y}}{\partial \mathbf{x}} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \left(-\frac{\partial \mathbf{J}}{\partial \mathbf{x}} \Delta \mathbf{y} + \frac{\partial \Delta \mathbf{b}}{\partial \mathbf{x}} \right) \quad (\text{B.2.15})$$

As for the last iteration step $\Delta \mathbf{y}$ is close to zero, \mathbf{J}_x can be simplified, leaving

$$\mathbf{J}_x \approx (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \frac{\partial \Delta \mathbf{b}}{\partial \mathbf{x}}. \quad (\text{B.2.16})$$

B.2.3 Homogeneous Equation

A non-zero solution to the homogeneous equation $M\mathbf{y} = 0$ is found by a constraint minimization

$$\min_y C = \|\mathbf{M}\mathbf{y}\|^2, \quad (\text{B.2.17})$$

with

$$\|\mathbf{y}\| = 1, \quad (\text{B.2.18})$$

which is also used to find the uncertainty of the solution (J. Clarke, 1998). C is minimized by the smallest eigenvalue of $A = M^T M$, obtained by a singular value decomposition.

The covariance of the solution is gained from $\Phi = A\mathbf{y}$ with

$$\Sigma_y = J\Sigma_x J^T \quad (\text{B.2.19})$$

where the Jacobian is given by

$$J = - \left(\frac{\partial \Phi}{\partial \mathbf{y}} \right)^{-1} \frac{\partial \Phi}{\partial \mathbf{x}}, \quad (\text{B.2.20})$$

using the pseudo-inverse with

$$\left(\frac{\partial \Phi}{\partial \mathbf{y}} \right)^{-1} \approx A^+ \quad (\text{B.2.21})$$

and

$$\mathbf{U} = \frac{\partial \Phi}{\partial A} = \begin{bmatrix} \frac{\partial \Phi_1}{\partial a_{11}} & \cdots & \frac{\partial \Phi_1}{\partial a_{1n}} & \frac{\partial \Phi_1}{\partial a_{21}} & \cdots & \frac{\partial \Phi_1}{\partial a_{nn}} \\ \vdots & & & & & \vdots \\ \frac{\partial \Phi_n}{\partial a_{11}} & \cdots & \frac{\partial \Phi_n}{\partial a_{1n}} & \frac{\partial \Phi_n}{\partial a_{21}} & \cdots & \frac{\partial \Phi_n}{\partial a_{nn}} \end{bmatrix} \quad (\text{B.2.22})$$

$$\mathbf{V} = \frac{\partial A}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial a_{11}}{\partial x} & \cdots & \frac{\partial a_{1n}}{\partial x} & \frac{\partial a_{21}}{\partial x} & \cdots & \frac{\partial a_{nn}}{\partial x} \end{bmatrix}^T \quad (\text{B.2.23})$$

yielding

$$J = -A^+ \mathbf{U} \mathbf{V}. \quad (\text{B.2.24})$$

B.3 Kalman Filter

The Kalman filter gives an optimal state estimate of a linear dependent system with measurements in the presents of noise. A good introduction to Kalman filters is given by Welch and Bishop (1995). The derivation of the Kalman filter equations from the Bayesian point of view can be found in Ribeiro (2004).

B.3.1 Continuous System Model

A continuous time, linear system is described by the differential equation

$$\dot{x} = Fx + Bu + Gw, \quad (\text{B.3.1})$$

with the system matrix F , state vector x , input matrix B , input vector u , noise transition matrix G and system noise w . According to Simon (2006), the solution for eq. (B.3.1) is given with

$$x_{k+1} = e^{F(t_{k+1}-t_k)} x_k + \int_{t_k}^{t_{k+1}} e^{F(t_{k+1}-\tau)} B u_k d\tau + \int_{t_k}^{t_{k+1}} e^{F(t_{k+1}-\tau)} G w_k d\tau. \quad (\text{B.3.2})$$

Using the substitution $\alpha = \tau - t_k$ and the discretization step size $\Delta t = t_{k+1} - t_k$ this can be rewritten to

$$x_{k+1} = e^{F\Delta t} x_k + e^{F\Delta t} \int_0^{\Delta t} e^{-F\alpha} d\alpha B u_k + e^{F\Delta t} \int_0^{\Delta t} e^{-F\alpha} d\alpha G w_k. \quad (\text{B.3.3})$$

If the matrices F, B, G and vectors u, w are assumed to be constant within the integration interval, a discrete-time, linear approximation can be obtained by

$$x_{k+1} \approx \Phi_k x_k + \Phi_k B \Delta t u_k + \Phi_k G \Delta t w_k \quad (\text{B.3.4})$$

$$\approx \Phi_k x_k + B_k u_k + G_k w_k, \quad (\text{B.3.5})$$

with the time-discrete system matrix

$$\Phi_k = e^{F\Delta t} = \sum_{j=0}^{\infty} \frac{(F\Delta t)^j}{j!} \quad (\text{B.3.6})$$

$$= (F\Delta t)^0 + (F\Delta t)^1 + \frac{(F\Delta t)^2}{2!} + \frac{(F\Delta t)^3}{3!} + \dots \quad (\text{B.3.7})$$

$$\approx I + F\Delta t. \quad (\text{B.3.8})$$

A complete derivation of the above equations is found in Simon (2006).

B.3.2 Linear Kalman Filter

The Kalman filter approach addresses the recursive estimation of the state $x \in \mathbb{R}^n$ from the stochastic difference equation given in eq. (B.3.8)

$$x_{k+1} = \Phi_k x_k + B_k u_{k+1} + G_k w_k \quad (\text{B.3.9})$$

and a measurement $\mathbf{y} \in \mathbb{R}^m$, which is

$$\mathbf{y}_{k+1} = \mathbf{H}\mathbf{x}_{k+1} + \mathbf{v}_{k+1}. \quad (\text{B.3.10})$$

The propagation of the state from the previous time step k to the current time step $k + 1$ is described by $n \times n$ state transition model Φ . An optional control input $\mathbf{u} \in \mathbb{R}^l$ can be included with the $n \times l$ control-input model \mathbf{B} . The $n \times m$ observation model \mathbf{H} relates the measurement \mathbf{y} to the state \mathbf{x} . The process noise \mathbf{w} and the measurement noise \mathbf{v} are assumed to be zero mean, independent, white, and normal distributed:

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}) \quad (\text{B.3.11})$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}), \quad (\text{B.3.12})$$

with the process noise covariance \mathbf{Q} and the measurement noise covariance \mathbf{R} . The Kalman filter equations consists of a time update and a measurement update. First, the *time update* propagates an a priori state estimate $\hat{\mathbf{x}}_{k+1}^-$ with the a priori covariance $\hat{\mathbf{P}}_{k+1}^-$.

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_{k+1} \quad (\text{B.3.13})$$

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k \Phi_k^T + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^T \quad (\text{B.3.14})$$

In the second step, called *measurement update*, a measurement is included by means of the *innovation*: updating the a priori state estimate to an a posteriori state $\hat{\mathbf{x}}_{k+1}$ with covariance \mathbf{P}_{k+1} .

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} \overbrace{(\mathbf{y}_{k+1} - \mathbf{H}\hat{\mathbf{x}}_{k+1}^-)}^{\text{innovation}} \quad (\text{B.3.15})$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{P}_{k+1}^- \quad (\text{B.3.16})$$

The $n \times m$ Kalman gain \mathbf{K} weights the innovation to the a priori state estimate and minimizes the a posteriori error covariance.

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (\text{B.3.17})$$

Bibliography

- Albert, W. (1994). "Monolithic quartz structure vibrating beam accelerometer (VBA)". In: *IEEE International Frequency Control Symposium*, pp. 415–420.
- Arras, K. (1998). *An introduction to error propagation: derivation, meaning and examples of $C_y = F_x C_x F_x^T$* . Tech. rep. Autonomous Systems Lab, EPFL.
- Atiya, S. and G. Hager (1993). "Real-time vision-based robot localization". In: *IEEE Transactions on Robotics and Automation* 9.6, pp. 785–800.
- Aulinas, J., Y. Petillot, J. Salvi, and X. Lladó (2008). "The SLAM problem: A survey". In: *Conference on Artificial Intelligence Research and Development*, pp. 363–371.
- Bailey, T. and H. Durrant-Whyte (2006). "Simultaneous localization and mapping (SLAM): part II". In: *IEEE Robotics and Automation Magazine* 13.3, pp. 108–117.
- Bancroft, J. and G. Lachapelle (2012). "Estimating MEMS gyroscope g-sensitivity errors in foot mounted navigation". In: *Ubiquitous Positioning, Indoor Navigation, and Location Based Service*.
- Bestaven, E., E. Guillaud, and J.-R. Cazalets (2012). "Is 'circling' behavior in humans related to postural asymmetry?" In: *PloS one* 7.9, e43861.
- Biswas, J. and M. Veloso (2012). "Depth camera based indoor mobile robot localization and navigation". In: *IEEE International Conference on Robotics and Automation*, pp. 1697–1702.
- Bonin-Font, F., A. Ortiz, and G. Oliver (2008). "Visual navigation for mobile robots: a survey". In: *Journal of Intelligent and Robotic Systems* 53, pp. 263–296.
- Börner, A., D. Griesßbach, and I. Ernst (2010). "Real-time dense stereo mapping". In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII*, pp. 256–261. URL: <http://www.isprs.org/proceedings/XXXVIII/part5/papers/218.pdf>.

- Bowditch, N. (1802). *The new american practical navigator*. U.S. Government Printing Office.
- Brown, A. and D. Sullivan (2002). *Inertial navigation electro-optical aiding during GPS dropouts*. Tech. rep. NAVSYS Corporation.
- Brown, D. (1971). "Close-range camera calibration". In: *Photogrammetric Engineering* 37, pp. 855–866.
- Burschka, D. and G. Hager (2004). "V-GPS (SLAM): vision-based inertial system for mobile robots". In: *IEEE International Conference on Robotics and Automation* 1, pp. 409–415.
- Chao, H., Y. Gu, and M. Napolitano (2014). "A survey of optical flow techniques for robotics navigation applications". In: *Journal of Intelligent & Robotic Systems* 73.1-4, pp. 361–372.
- Chao, S. (2009). *Indoorpositioning: technologies, characteristics and limitations*. Tech. rep. Albert-Ludwigs-Universität Freiburg.
- Clark, S. and G. Dissanayake (1999). "Simultaneous localisation and map building using millimetre wave radar to extract natural features". In: *IEEE International Conference on Robotics and Automation* 2.May, pp. 1316–1321.
- Clarke, J. (1998). *Modelling uncertainty: a primer*. Tech. rep. Department of Engineering Science, Oxford University.
- Clarke, T. and J. Fryer (1998). "The development of camera calibration methods and models". In: *Photogrammetric Record* 16.91, pp. 51–66.
- Clarke, T., J. Fryer, and X. Wang (1998). "The principal point and ccd cameras". In: *Photogrammetric Record* 16, pp. 293–312.
- Coombs, D. and K. Roberts (1993). "Centering behavior using peripheral vision". In: *Computer Vision and Pattern Recognition*, pp. 440–445.
- Corke, P., J. Lobo, and J. Dias (2007). "An introduction to inertial and visual sensing". In: *International Journal of Robotics Research* 26.6, pp. 519–535.
- Cox, I. (1991). "Blanche – an experiment in guidance and navigation of an autonomous robot vehicle". In: *IEEE Transactions on Robotics and Automation* 7.2, pp. 193–204.
- Cummins, M. and P. Newman (2007). "Probabilistic appearance based navigation and loop closing". In: *IEEE International Conference on Robotics and Automation*, pp. 10–14.

- Dam, E., M. Koch, and M. Lillholm (1998). *Quaternions, interpolation and animation*. Tech. rep. DIKU-TR-98/5. University of Copenhagen.
- Davison, A. J., I. D. Reid, N. D. Molton, and O. Stasse (2007). "MonoSLAM: real-time single camera SLAM". In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 29, p. 2007.
- Dellaert, F., W. Burgard, D. Fox, and S. Thrun (1999). "Using the condensation algorithm for robust, vision-based mobile robot localization". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2, pp. 588–594.
- DeSouza, G. N. and A. C. Kak (2002). "Vision for mobile robot navigation: a survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2, pp. 237–267.
- Devernay, F. and O. D. Faugeras (2001). "Straight lines have to be straight". In: *Machine Vision and Applications* 13.1, pp. 14–24.
- Diel, D. D. (2005). "Stochastic constraints for vision-aided inertial navigation". PhD thesis. Massachusetts Institute of Technology.
- Durrant-Whyte, H. and T. Bailey (2006). "Simultaneous localisation and mapping (SLAM): part I the essential algorithms". In: *IEEE Robotics and Automation Magazine* 2, p. 2006.
- Eberly, D. (2002). *Rotation representations and performance issues*. <http://www.geometrictools.com>.
- Elfes, A. (1987). "Sonar-based real-world mapping and navigation". In: *IEEE Journal on Robotics and Automation* 3.3, pp. 249–265.
- Filliat, D. and J.-A. Meyer (2003). "Map-based navigation in mobile robots: I. a review of localization strategies". In: *Cognitive Systems Research* 4.4, pp. 243–282.
- Fischler, M. A. and R. C. Bolles (1980). *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Tech. rep. 213. 333 Ravenswood Ave., Menlo Park, CA 94025: AI Center, SRI International.
- Fox, D., W. Burgard, F. Dellaert, and S. Thrun (1999). "Monte Carlo localization: efficient position estimation for mobile robots". In: *National Conference on Artificial Intelligence*.
- Geiger, A., P. Lenz, and R. Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

- George, M. and S. Sukkarieh (2007). "Inertial navigation aided by monocular camera observations of unknown features". In: *IEEE International Conference on Robotics and Automation*, pp. 3558–3564.
- Gillingham, K. K. and F. H. Previc (1993). *Spatial orientation in flight*. Tech. rep. Armstrong Laboratory.
- Google (2014). *Project Tango*. www.google.com/atap/projecttango.
- Grewal, M., V. Henderson, and R. Miyasako (1990). "Application of Kalman filtering to the calibration and alignment of inertial navigation systems". In: *IEEE Conference on Decision and Control*, 3325–3334 vol.6.
- Grießbach, D., M. Bauer, A. Hermerschmidt, S. Krüger, and M. Scheele (2010). "Geometric stereo camera calibration with diffractive optical elements". In: *ISPRS International Calibration and Orientation Workshop*. Vol. XXXVIII. Castelldefels, Spain. URL: www.isprs.org/proceedings/XXXVIII/Eurocow2010/euroCOW2010_files/papers/12.pdf.
- Grießbach, D., M. Bauer, A. Hermerschmidt, S. Krüger, M. Scheele, and A. Schischmanow (2008). "Geometrical camera calibration with diffractive optical elements". In: *Optics Express* 16.25, pp. 20241–20248. URL: <http://dx.doi.org/10.1364/OE.16.020241>.
- Grießbach, D., D. Baumbach, A. Börner, M. Buder, I. Ernst, E. Funk, J. Wohlfeil, and S. Zuev (2012). "IPS – A system for real-time navigation and 3D-modeling". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIX-B5*, pp. 21–26. URL: www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B5/21/2012.
- Grießbach, D., D. Baumbach, and S. Zuev (2010). "Vision aided inertial navigation". In: *ISPRS International Calibration and Orientation Workshop*. Vol. XXXVIII. Castelldefels, Spain. URL: www.isprs.org/proceedings/XXXVIII/Eurocow2010/euroCOW2010_files/papers/13.pdf.
- Grießbach, D., D. Baumbach, and S. Zuev (2014). "Stereo vision aided inertial navigation for unknown indoor and outdoor environments". In: *The Fifth International Conference on Indoor Positioning and Indoor Navigation 2014 (IPIN 2014)*. Busan, Korea.
- Guivant, J., E. Nebot, and S. Baiker (2000). "Autonomous navigation and map building using laser range sensors in outdoor applications". In: *Journal of Robotic Systems* 17, pp. 3817–3822.

- Hamilton, W. (1844-1850). *On quaternions, or on a new system of imaginaries in algebra*. Philosophical Magazine.
- Hamilton, W. (1853). *Lectures on quaternions*. Dublin: Hodges Smith & Co.
- Harris, C. and M. Stephens (1988). "A combined corner and edge detector". In: *ALVEY Vision Conference*, pp. 147–151.
- Hartley, R. and P. Sturm (1995). "Triangulation". In: *Computer Analysis of Images and Patterns*. Vol. 970. Lecture Notes in Computer Science. Springer Berlin, Heidelberg, pp. 190–197.
- Hartley, R. and A. Zisserman (2000). *Multiple view geometry in computer vision*. Cambridge University Press.
- Heikkilä, J. and O. Silven (1997). "A four-step camera calibration procedure with implicit image correction". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112.
- Hirschmüller, H. (2003). "Stereo vision based mapping and immediate virtual walkthroughs". PhD thesis. De Montfort University, Leicester, UK.
- Hofmann-Wellenhof, B., K. Legat, and M. Wieser (2003). *Navigation - principles of positioning and guidances*. Springer.
- Horn, B. K. and B. G. Schunck (1981). "Determining optical flow". In: *Artificial Intelligence* 17.1-3, pp. 185–203.
- Howard, A. (2008). "Real-time stereo visual odometry for autonomous ground vehicles". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3946–3952.
- Huster, A. (2003). "Relative position sensing by fusing monocular vision and inertial rate sensors". PhD thesis. Stanford University.
- IEEE (1997, R2008). "Standard specification format guide and test procedure for single axis interferometric fiber optic gyros". In: *IEEE Standard 952*.
- IEEE (1998, R2008). "Standard specification format guide and test procedure for linear single-axis, nongyroscopic accelerometers". In: *IEEE Standard 1293*.
- IEEE (2001). "Standard for inertial sensor terminology". In: *IEEE Standard 528*.
- IEEE (2004, R2010). "Standard specification format guide and test procedure for coriolis vibratory gyros". In: *IEEE Standard 1431*.
- Johnson, E. and H. Mathies (1999). "Precise image-based motion estimation for autonomous small body exploration". In: *Artificial Intelligence, Robotics and Automation in Space*, pp. 627–634.

- Karlsson, N., E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich (2005). "The vSLAM algorithm for robust localization and mapping". In: *IEEE International Conference on Robotics and Automation*, pp. 24–29.
- Kayton, M. and W. R. Fried (1969). *Avionics navigation systems*. John Wiley & Sons.
- Kitt, B., A. Geiger, and H. Lategahn (2010). "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme". In: *IEEE Intelligent Vehicles Symposium*. San Diego, USA, pp. 486–492.
- Kjærsgaard, M. B. (2010). "Indoor positioning with radio location fingerprinting". PhD thesis. University of Aarhus.
- Konolige, K., J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua (2010). "View-based maps". In: *International Journal of Robotics Research* 29.8, pp. 941–957.
- Krutz, D. (2006). "Ein Betriebssystem für konfigurierbare Hardware". PhD thesis. Berlin: Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II.
- Lobo, J. (2002). "Inertial sensor data integration in computer vision systems". PhD thesis. University of Coimbra.
- Lobo, J. and J. Dias (2003). "Vision and inertial sensor cooperation using gravity as a vertical reference". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.12, pp. 1597–1608.
- Lobo, J. and J. Dias (2004). "Inertial sensed ego-motion for 3D vision". In: *Journal of Robotic Systems* 21, pp. 3–12.
- Longuet-Higgins, H. (1981). "A computer algorithm for reconstructing a scene from two projections". In: *Nature* 293.September, pp. 133–135.
- Matsumoto, Y., M. Inaba, and H. Inoue (1996). "Visual navigation using view-sequenced route representation". In: *Robotics and Automation*, pp. 83–88.
- Mathies, L. and S. Shafer (1987). "Error modelling in stereo navigation". In: *IEEE Journal of Robotics and Automation [legacy, pre - 1988]* 3.3, pp. 239–248.
- Mautz, R. (2012). "Indoor positioning technologies". PhD thesis. ETH Zurich.
- McPhedran, R., G. Derrick, and L. Brown (1980). "Electromagnetic theory of gratings". In: *Theory of crossed gratings*. Springer Verlag Berlin. Chap. R. Petit (ed.) Pp. 227–275.

- Mekid, S. and D. Vaja (2008). "Propagation of uncertainty: expressions of second and third order uncertainty with third and fourth moments". In: *Measurement* 41.6, pp. 600–609.
- Microsoft (2011). *Kinect for Windows*. www.microsoft.com/en-us/kinectforwindows.
- Miller, M. M., A. Soloviev, M. U. de Haag, M. Veth, J. Raquet, T. J. Klausutis, and J. E. Touma (2010). "Navigation in GPS denied environments: feature-aided inertial systems". In: *RTO-EN-SET-116, Low-Cost Navigation Sensors and Integration Technology*.
- Mohr, R. and B. Triggs (1996). "Projective geometry for image analysis". In: *ISPRS International Symposium on Photogrammetry and Remote Sensing*.
- Moravec, H. P. (1980). "Obstacle avoidance and navigation in the real world by a seeing robot rover." PhD thesis. Carnegie-Mellon University.
- Moravec, H. P. and A. Elfes (1985). "High resolution maps from wide angle sonar". In: *IEEE International Conference on Robotics and Automation 2*, pp. 116–121.
- Nistér, D. (2004). "An efficient solution to the five-point relative pose problem." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.6, pp. 756–77.
- Nistér, D., O. Naroditsky, and J. Bergen (2004). "Visual odometry". In: *Computer Vision and Pattern Recognition*, pp. 652–659.
- Olson, C. F., L. H. Matthies, M. Schoppers, and M. W. Maimone (2003). "Rover navigation using stereo ego-motion". In: *Robotics and Autonomous Systems* 43.4, pp. 215–229.
- Oskiper, T., Z. Zhu, S. Samarasekera, and R. Kumar (2007). "Visual odometry system using multiple stereo cameras and inertial measurement unit". In: *IEEE Conference on Computer Vision and Pattern Recognition*.
- Perlmutter, M. and L. Robin (2012). "High-performance, low cost inertial MEMS: a market in motion!" In: *IEEE/ION Position Location and Navigation Symposium*, pp. 225–229.
- PrimeSense (2010). *3D Sensing Technology Solutions*. www.primesense.com.
- Randeniya, D., S. Sarkar, and M. Gunaratne (2010). "Vision-IMU integration using a slow-frame-rate monocular vision system in an actual roadway setting". In: *IEEE Transactions on Intelligent Transportation Systems* 11.2, pp. 256–266.
- Rauch, H. E., C. Striebel, and F. Tung (1965). "Maximum likelihood estimates of linear dynamic systems". In: *AIAA journal* 3.8, pp. 1445–1450.

- Ribeiro, M. I. (2004). *Kalman and extended Kalman filters: concept, derivation and properties*. Tech. rep. Institute for Systems and Robotics, Lisboa.
- Roumeliotis, S. and J. Burdick (2002). "Stochastic cloning: A generalized framework for processing relative state measurements". In: *IEEE International Conference on Robotics and Automation*. Vol. 2, pp. 1788–1795.
- Roumeliotis, S., A. Johnson, and J. Montgomery (2002). "Augmenting inertial navigation with image-based motion estimation". In: *IEEE International Conference on Robotics and Automation*. Vol. 4, pp. 4326–4333.
- Schmid, K., F. Ruess, M. Suppa, and D. Burschka (2012). "State estimation for highly dynamic flying systems using key frame odometry with varying time delays". In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2997–3004.
- Schmidt, G. T. (2010). "INS/GPS technology trends". In: *RTO-EN-SET-116, Low-Cost Navigation Sensors and Integration Technology*.
- Schubert, M. and L. Minor (2004). "Vestibulo-ocular physiology underlying vestibular hypofunction". In: *Physical Therapy* 84, pp. 373–385.
- Schuster, R. and B. Braunecker (2000). "The calibration of the ADC (airborne digital camera) -system". In: *ISPRS International Archive of Photogrammetry and Remote Sensing*, pp. 288–294.
- Shi, J. and C. Tomasi (1994). "Good features to track". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Seattle.
- Shoemake, K. (1994). *Quaternions*. Tech. rep. Department of Computer and Information Science University of Pennsylvania.
- Sim, R. and J. Little (2006). "Autonomous vision-based exploration and mapping using hybrid maps and Rao-Blackwellised particle filters". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2082–2089.
- Simon, D. (2006). *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. Wiley-Interscience.
- Souman, J. L., I. Frissen, M. N. Sreenivasa, and M. O. Ernst (2009). "Walking straight into circles". In: *Current Biology* 19.18, pp. 1538–1542.
- Srinivasan, M. V., S. W. Zhang, J. S. Chahl, G. Stange, and M. Garratt (2004). "An overview of insect-inspired guidance for application in ground and airborne platforms". In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 218.6, pp. 375–388.

- Stovall, S. H. (1997). *Basic inertial navigation*. Tech. rep. NAWCWPNS TM 8128.
- Stratmann, I. and E. Solda (2004). "Omnidirectional vision and inertial clues for robot navigation". In: *Journal of Robotic Systems* 21, pp. 33–39.
- Strelow, D. (2004). "Motion estimation from image and inertial measurements". PhD thesis. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.
- Strobl, K., W. Sepp, S. Fuchs, C. Paredes, and K. Arbter (2005). *DLR Callab and DLR CalDe*. Oberpfaffenhofen, Germany: Institute of Robotics and Mechatronics, German Aerospace Center (DLR). URL: <http://www.robotic.dlr.de/callab/>.
- Thrun, S., D. Fox, W. Burgard, and F. Dellaert (2000). "Robust monte carlo localization for mobile robots". In: *Artificial Intelligence* 128.1-2, pp. 99–141.
- Thrun, S. (2002). "Robotic Mapping: A Survey". In: *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.
- Titterton, D. and J. Weston (2004). *Strapdown inertial navigation technology*. Second. MIT Lincoln Laboratory.
- Torres-Solis, J., T. Falk, and T. Chau (2010). "A review of indoor localization technologies: towards navigational assistance for topographical disorientation". In: *Ambient Intelligence*.
- Tsai, R. (1987). "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses". In: *IEEE Journal of Robotics and Automation* 3.4, pp. 323–344.
- Veth, M. and J. Raquet (2007). *Fusion of low-cost imaging and inertial sensors for navigation*. Tech. rep. Air Force Institute of Technology.
- Vicci, L. (2001). *Quaternions and rotations in 3-space: the algebra and its geometric interpretation*. Tech. rep. Chapel Hill, NC, USA: University of North Carolina.
- Viéville, T. and O. D. Faugeras (1989). "Computation of inertial information on a robot". In: *International Symposium on Robotics Research*, pp. 57–65.
- Viéville, T. and O. D. Faugeras (1990). "Cooperation of the inertial and visual systems". In: *Traditional and Non-Traditional Robotic Sensors*. Ed. by T. C. Henderson. Vol. 63. NATO ASI Series. Springer Berlin Heidelberg, pp. 339–350.
- Walchko, K. (2002). "Low cost inertial navigation: learning to integrate noise and find your way". PhD thesis. University of Florida.
- Weinberg, H. (2011). *Gyro mechanical performance: the most important parameter*. Tech. rep. Analog Device, Inc.

- Welch, G. and G. Bishop (1995). *An introduction to the Kalman filter*. Tech. rep. Chapel Hill, NC, USA: University of North Carolina at Chapel Hill.
- Wendel, J. (2003). "Entwurfs- und Analysemethoden Integrierter Navigationssysteme". PhD thesis. University Fridericiana Karlsruhe.
- Weng, J., P. Cohen, and M. Herniou (1992). "Camera calibration with distortion models and accuracy evaluation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.10, pp. 965–980.
- Woodman, O. J. (2007). *An introduction to inertial navigation*. Tech. rep. UCAM-CL-TR-696. University of Cambridge, Computer Laboratory. URL: <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>.
- You, S., U. Neumann, and R. Azuma (1999). "Hybrid inertial and vision tracking for augmented reality registration". In: *IEEE Virtual Reality*, pp. 260–267.
- Zhang, Z. (1996). "The epipolar geometry between two images with lens distortion". In: *IEEE International Conference on Pattern Recognition*. Vol. I, pp. 407–411.
- Zhang, Z. (2000). "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11, pp. 1330–1334.
- Zhu, Z., T. Oskiper, O. Naroditsky, S. Samarasekera, H. Sawhney, and R. Kumar (2006). "An improved stereo-based visual odometry system". In: *Performance Metrics for Intelligent Systems Workshop*.
- Zumbahlen, H. (2005). *Basic linear design*. Analog Devices, Incorporated.

Acronyms

DOE Diffractive Optical Element

FPGA Field Programmable Gate Array

GNSS Global Navigation Satellite System

GPS Global Positioning System

IMU Inertial Measurement Unit

INS Inertial Navigation System

LTP Local Tangent Plane

MEMS Microelectromechanical System

NCC Normalized Cross Correlation

RANSAC Random Sample Consensus

RMS Root Mean Square

SfM Structure from Motion

VINS Vision-Aided Inertial Navigation System

VO Visual Odometry

Declaration

I hereby declare that this thesis is a record of work undertaken by myself and is not being submitted concurrently in candidature for any other degree.

Denis Grießbach