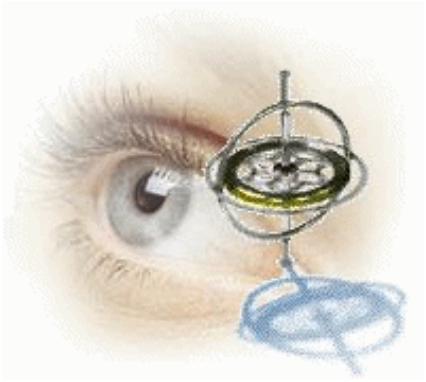
University of Coimbra
Faculty of Science and Tecnology
Electrical and Computer Engineering Department



INTEGRATION OF VISION AND INERTIAL SENSING

Ph.D. Thesis

Jorge Nuno de Almeida e Sousa Almada Lobo

Electrical Engineer

Coimbra
December 2006

Dissertation presented to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

This work was done under the supervision of Doctor Jorge Manuel Miranda Dias, associate professor at the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra.

This thesis is dedicated to my wife, Sílvia, and to our future daughter.

"A thesis cannot be made perfect in a finite amount of time."

citation related to Murphy's Law.

Acknowledgments

I would like to thank Doctor Jorge Dias for his supervision, support and encouragement; ISR-Coimbra[1] for the facilities provided and project financing; DEEC[2] for employment and support for my work; and all my colleagues at ISR, João Filipe, Paulo Menezes, Rui Rocha, Luis Almeida, João Alves, Joërg Rett, Luiz Mirisola, amongst others. I would also like to thank my parents and family for their love and support, and my father for proofreading the thesis and providing useful comments and suggestions. I would like to especially thank my wife Sílvia, for all her love, patience, support and inspiration.

[1]Institute of Systems and Robotics, Coimbra

[2]Department of Electrical and Computer Engineering at Coimbra University

Abstract

Inertial sensors coupled to cameras can provide valuable data about camera ego-motion and how world features are expected to be oriented. Object recognition and tracking benefits from both static and inertial information. Several human vision tasks rely on the inertial data provided by the vestibular system. Artificial systems should also exploit this sensor fusion.

Micromachining enabled the development of low-cost single chip inertial sensors. These can be easily incorporated alongside the camera's imaging sensor, providing an artificial vestibular system.

We will explore some of the benefits of combining the two sensing modalities, and how gravity can be used as a vertical reference. We will also focus on how the two sensors can be cross-calibrated so that they can be used in static and dynamic situations.

The inertial sensed gravity provides a vertical reference for monocular and stereo vision systems, establishing an artificial horizon, enabling segmentation of vertical features and providing restrictions for stereo correspondence of ground plane points and 3D vertical features. This vertical reference can also enable stereo depth map alignment and ground segmentation, reducing the dimensionality of the full registration problem.

To perform independent motion segmentation for a moving robotic, observer we explored the fusion of optical flow and stereo techniques with data from the inertial and magnetic sensors. A depth map registration and independent motion segmentation is presented that explores the cooperation between distinct sensing modalities.

Contents

Chapter 1

Introduction

1.1 Motivation

In our days, machines are no longer expected to be numb and repetitive, but intelligent, autonomous to some extent, and interactive. If autonomous robotic machines are to be integrated in man's environment, they must be capable of perceiving their surroundings. One fundamental component of this perception is vision, but other sensory modalities can play a significant role and enhance artificial vision systems. As with other robotic applications, interesting hints can be gathered by looking at how the human and animal perception systems work.

In humans and in animals the vestibular system in the inner ear gives inertial information essential for navigation, orientation, body posture control and equilibrium. In humans this sensorial system is crucial for several visual tasks and head stabilisation. It is well known that the information provided by the vestibular system is used during the execution of visual movements such as gaze holding and tracking, as described by [Carpenter1988]. Neural interactions of human vision and vestibular system occur at a very early processing stage [Berthoz2000][Gillingham1996]. The inertial information enhances the performance of the vision system, and the visual cues aid the spatial orientation and body equilibrium.

Inertial sensors explore intrinsic properties of body motion. From the principle of

generalised relativity of Einstein we known that only the specific force on one point and the angular instantaneous velocity, but no other quantity concerning motion and orientation with respect to the rest of the universe, can be measured from physical experiments inside an isolated closed system. Therefore from inertial measurements one can only determine an estimate for linear acceleration and angular velocity. Linear velocity and position, and angular position, can be obtained by integration. Inertial navigation systems (INS) implement this process of obtaining velocity and position information from inertial sensor measurements.

Internal sensing using inertial sensors is very useful in mobile robotic systems since it is not dependent on any external references, except for the gravity field which does provide an external reference. Artificial vision systems can provide better perception of the robot's environment by using the inertial sensors measurements of camera pose (rotation and translation). As in human vision, low level image processing should take into account the ego motion of the observer.

Micromachining enabled the development of low-cost single chip inertial sensors. These can be easily incorporated alongside the camera's imaging sensor, providing an artificial vestibular system. The noise level of these sensors is not suitable for inertial navigation systems, but their performance is similar to biological inertial sensors and can play a key role in artificial vision systems.

In our work we explore some aspects of inertial and vision sensing integration. Fig. 1.1 shows some of the data available from the two sensors and processing systems, setting a framework for possible combination of inertial and vision sensing

The 3D structured world is observed by the visual sensor, and its pose and motion parameters directly measured by the inertial sensors. These motion parameters can also be inferred from the image flow and known scene features [Eason1992]. Combining the two sensing modalities simplifies the 3D reconstruction of the observed world. The inertial sensors also provide important cues about the observed scene structure, such as vertical and horizontal references.

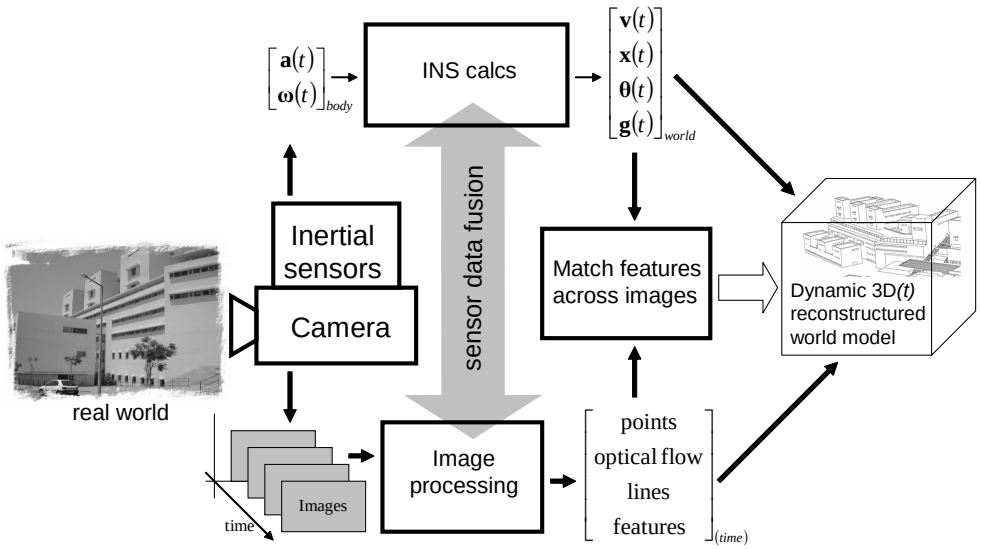


Figure 1.1: Combining inertial and vision sensing

1.2 Human Vision and Inertial Sensing

In the next sections we will briefly describe the human vestibular system, its main functions and interactions with human vision. For a more detailed and medical description see [Gillingham1996], and for more on human senses and perception see [Coren1994] and [Carpenter1988]. [Berthoz2000] gives insight into neural interactions of human vision and vestibular system, and action-perception behaviours.

1.2.1 Human Vestibular System

Within the vestibule of inner ear we find the human inertial sensor, the vestibular system (see figure 1.2). It measures both tilt and angular acceleration. The vestibular end-organs measure just 1.5 *cm* across, and reside well protected within the bony labyrinth of the temporal bone.

It has three main parts: the cochlea, the vestibule, and the semicircular canals. They are all filled with a fluid, the endolymph. The cochlea, the snail-like part seen in figure 1.2, converts acoustic energy into neural information. In the vestibule lie the two otolith organs, the utricle and the saccule. They translate gravitational and inertial forces into

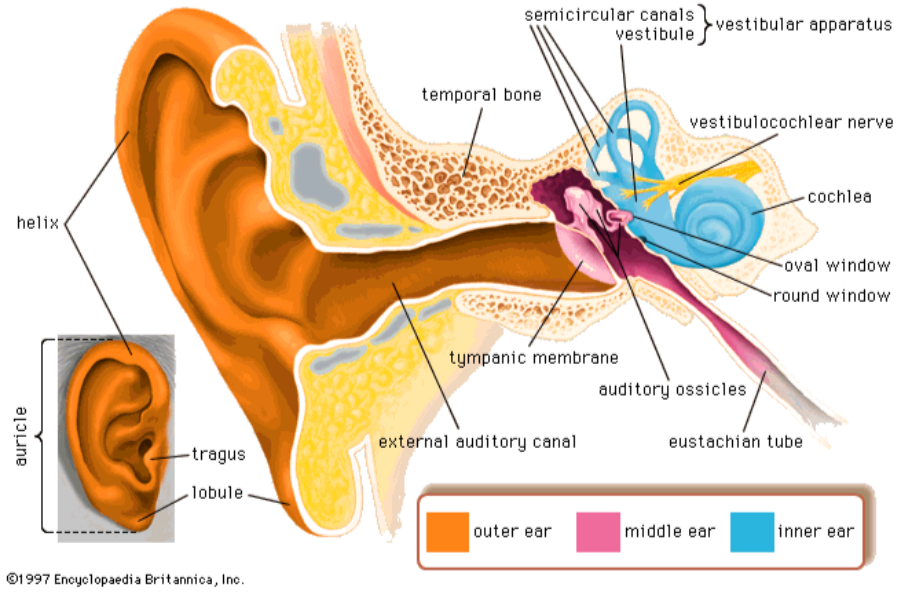


Figure 1.2: Human Ear (taken with permission from [Britannica2001]).

spatial orientation information, namely information about angular position (tilt) and linear motion of the head. The semicircular canals detect angular acceleration of the head. The three semicircular canals are oriented in three mutually perpendicular planes, thus measuring angular acceleration in space.

The hair cell is the functional unit of the vestibular sensory system. It converts spatial and temporal patterns of mechanical energy applied to the head into neural information.

The semicircular ducts communicate at both ends with the utricle, and are dilated at one end to form the ampulla. Inside the ampulla lies the crista ampullaris, composed of hair cells and a gelatinous structure, the cupula, as indicated in figure 1.3. When angular acceleration of the head occurs, with components in each semicircular duct plane, the endolymph inertia will deviate the cupula, bending the hairs of the crista. With the usual rapid, high frequency rotations of the head, the rotational inertia of the endolymph acts to deviate the cupula as the angular velocity of the head builds. The angular momentum gained by the endolymph during the brief acceleration acts to drive the cupula back to its resting position when the head decelerates to a stop. The cupula-endolymph system thus normally functions as an integrating angular accelerometer, *i.e.*, it converts angular

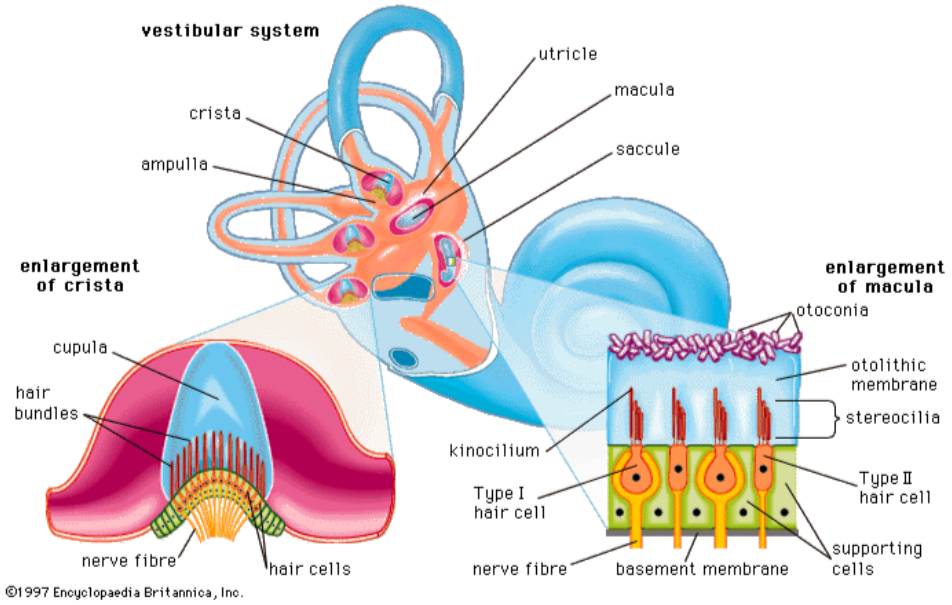


Figure 1.3: Human Vestibular System (taken with permission from [Britannica2001]).

acceleration data into a neural signal proportional to the angular velocity of the head.

The utricle and saccule have a similar arrangement. There are patches of hair cells, the macula, lining the bottom of the utricle in a close to horizontal plane, and lining the medial wall of the saccule in a vertical plane. Above each macula there are gelatinous structures, the otolithic membranes. These membranes act as a proof mass, and bend the macular hairs, sending neural signals proportional to angular position and linear motion of the head.

The above described vestibular system is therefore capable of sensing three-dimensional angular acceleration, linear acceleration and tilt.

1.2.2 Human Vision and Vestibular System

In humans, the retinal image is stabilised mainly by vestibulo-ocular reflexes, primarily those of semicircular-duct origin. A simple demonstration can help one appreciate the contribution of the vestibulo-ocular reflexes to retinal-image stabilisation. Holding the extended fingers half a meter or so in front of the face, one can move the fingers slowly

from side to side and still see them clearly because of visual (optokinetic) tracking reflexes. As the frequency of movement increases one eventually reaches a point where the fingers cannot be seen clearly - they are blurred by the movement. This point is about 60 $deg.s^{-1}$ or 1 or 2 Hz for most people. Now, if the fingers are held still and the head is rotated back and forth at the frequency at which the fingers became blurred, the fingers remain perfectly clear. Even at considerably higher frequencies of head movement, the vestibulo-ocular reflexes initiated by the stimulation of the semicircular ducts keep the image of the fingers clear.

For lower frequencies of movement of external world features relative to the body, or body motion relative to the world, gaze stabilisation is done by the visual system with the optokinetic tracking reflexes. As the frequency increases, the vestibulo-ocular reflexes assume a predominant role. In normal human activity, the higher frequencies of relative motion are due to head and body motion, where the vestibular system can provide a suitable stimulus for the gaze stabilisation reflexes.

The eye movement resulting from the vestibulo-ocular reflex is compensatory, that is, it adjusts the angular position of the eye to compensate for changes in angular position of the head, preventing slippage of the retinal image over the retina. Because the amount of angular deviation of the eye is physically limited, rapid movements of the eyes in the direction opposite to the compensatory motion are employed to return the eye to its initial position or to advance it to a position from which it can sustain a compensatory sweep for a suitable length of time. Due to their very high angular velocity, the rapid eye movements of the vestibulo-ocular reflex are not perceived as motion.

Many everyday behaviours give evidence of the interaction of the visual and vestibular system. The sensation of vertigo occurs when the visual stimulus conflicts with the vestibular information. Looking down from a high cliff one tends to sway so as to obtain a visual stimulus, but since the viewed scene is very far away, even large amplitude movements fail to provide any visual stimulus. But the large swaying motion will trigger the vestibular system, giving an alarm that the body if out of balance. This is further evidence that the visual system has a predominant role in spatial orientation.

Another example is the fact that figure skaters when spinning keep the head and eyes fixed, and perform a rapid rotation of the head, halfway through the body rotation. This

way the endolymph in the semicircular canals will not be set in motion along with the body rotation, preventing dizziness when the rotation stops and enabling the skater to keep a good spatial orientation.

Motion sickness is yet another example, it occurs when we experience strong vestibular stimulus without the corresponding visual cues, such as when sitting in the back seat of a car along a winding road. The sickness itself is triggered by the fact that the body attributes the conflicting stimulus to some kind of poisoning, and empties the stomach as a defensive measure.

Human sense of motion is derived from two main factors: the contribution of the vestibular system and retinal visual flow. Visual and vestibular information are integrated at very basic neural levels. The inertial information enhances the performance of the vision system in tasks such as gaze stabilisation, and the visual cues aid the spatial orientation and body equilibrium. While this is usually beneficial, when the vestibular system response is surpassed, such as in a maneuvering fighter airplane, the resulting spatial disorientation is difficult or impossible to correct by higher-level neural processing.

1.2.3 Performance of Human Inertial Sensors

It is important to have some idea of the performance of the human inertial sensors to better evaluate the suitability of inertial sensors in some robotic applications. But measuring the actual vestibular perceptual thresholds is difficult; they are determined by many factors such as mental concentration, fatigue, other stimulus capturing the attention, and vary from person to person [Gillingham1996]. Reasonable threshold values for perception of angular acceleration are 0.14, 0.5 and 0.5 $deg.s^{-2}$ for yaw, roll, and pitch motions, respectively. A 1.5 *deg* change in direction of applied gravity force is perceptible by the otolith organs under ideal conditions. Values of 0.01 g for vertical and 0.006 g for horizontal acceleration are appropriate representative thresholds for perceptible intensity of linear acceleration. These are valid for sustained and relatively low frequency stimulus.

These performances are not suitable for stand alone inertial navigation, but combined with vision cues they contribute to human spatial orientation and body equilibrium. The inertial cues enhance the performance of the vision system in gaze stabilisation, tracking

and visual navigation.

The currently available low cost inertial sensors, accelerometers and gyroscopes, are capable of similar performances [Lobo2002MSc]. Notice that gyroscopes measure angular velocity and not angular acceleration.

1.3 Our Work

In this work we try to set a common framework for research into the integration of inertial sensor data in computer vision systems, identify the main issues and overview all the different aspects of combining the two sensing modalities.

Inertial sensors coupled to cameras can provide valuable data about camera ego-motion and how world features are expected to be oriented. Object recognition and tracking benefits from both static and inertial information. Several human vision tasks rely on the inertial data provided by the vestibular system. Artificial systems should also exploit this sensor fusion.

In our work we explored some of the benefits of combining the two sensing modalities, and how gravity can be used as a vertical reference. In [Lobo2002MSc] an overview of the current inertial sensor technology was given, focusing on low cost sensors suitable for robotic applications, and results using the inertial vertical reference in vision systems presented. In [Lobo2003PAMI] a framework is set for vision and inertial sensor cooperation. The use of gravity as a vertical reference is explored, enabling camera focal distance calibration with a single vanishing point, vertical line segmentation, and ground plane segmentation. In [Lobo2003JRAS] world vertical feature detection and 3D mapping is presented. In [Lobo2004JRS] we continue to explore the use of inertial data in vision systems, and present a method for fast alignment and segmentation of depth maps obtained from correlation based stereovision. In [Lobo2005InerVis] we focus on how the two sensors can be cross-calibrated so that they can be used in static and dynamic situations.

In this thesis we report our results on the use of inertial data in vision systems, and try to set a common framework, identify the main issues and overview all the different aspects of combining the two sensing modalities.

The inertial and vision sensor models are presented, and the relationship between them analysed. Calibration is studied in more detail, exploring the synergies of combined cross-calibration, presenting a simple calibration procedure.

In vision based systems used in mobile robotics, the perception of self-motion and structure of the environment is essential. Inertial sensors can provide valuable data about camera ego-motion, as well as absolute references for structure feature orientations.

We explore the use of the inertial vertical reference provided by gravity in robotics vision systems. Knowing the geometry of a stereo rig, and its pose from the inertial sensors, the collineation of level planes can be recovered, providing enough restrictions to segment and reconstruct 3D vertical features and levelled planar patches.

To perform independent motion segmentation for a moving robotic observer we explored the fusion of optical flow and stereo techniques with data from the inertial and magnetic sensors. The magnetic sensor complement the vertical reference to provide an absolute 3D rotation reference. A depth map registration and motion segmentation method is proposed, and experimental results of stereo depth flow segmentation obtained from a moving observer are presented.

To summarise, the key contributions are:

- a common framework for inertial-vision sensor integration;
- calibration methods for integrated inertial and vision systems;
- vertical feature segmentation and 3D mapping;
- ground plane segmentation;
- 3D depth map registration;
- independent motion segmentation.

The implemented calibration methods are made available to the public domain in the InerVis Matlab Toolbox [Lobo2006], see appendix C.

Ongoing work is being done taking a biomimetic approach, i.e., to try to mimic the way biological systems fuse multimodal data based on neurological and psychophysical studies, going beyond the bioinspired use of inertial cues in vision systems [Lobo2006ICVW].

1.4 An Overview of Related Work

Integration of visual and inertial sensing modalities offer new application directions in robotics. It can lead to robust solutions for image segmentation and 3D structure recovery from images. It can improve estimation of ego-motion of an autonomous system in several important cases like navigation, surveillance, 3D human-computer interaction. The advantages of integrating the two sensing modalities in robotic applications are based on the complementary characteristics of cameras and inertial sensors.

The benefits of combining the two sensing modalities have been reported by the researcher community on different applications and domains.

Neurological Studies

To better exploit the benefits of combining the two sensing modalities in artificial systems, a clear understanding of biological systems is important. Vestibular information is necessary not only for vestibular reflexes but also in various cognitive functions for our adequate behaviour in three-dimensional space. In [Fukushima1997] the regions of the cerebral cortex where vestibular information is represented is investigated. Perception and action influence each other [Hurley2001], making some biological highly coupled and complex, from which direct models for sensor fusion are not easily derived. In [Leone1998] and [Angelaki1999] the role of gravity in visual perception and how the brain deals with the ambiguity between inclination and body acceleration is investigated. In [Harris2000] and [Reymond2002] the motion perception inferred from visuo-vestibular cues is studied. The perceived relative motion is important for posture control [Kelly2005]. Taking advantage of improved brain imaging techniques, a better understanding of the visual motion and self-movement interactions has been pursued [Beer2002] [Previc2000].

Computer vision

In [Vieville1989] the use inertial sensors in computer vision applications was proposed, further works studied the cooperation of the inertial and visual systems in mobile robot navigation by using the vertical cue, rectifying images and improving self-motion estimation for 3D structure reconstruction [Vieville1990] [Vieville1993IROS] [Vieville1993ICCV] [Vieville1995] [Vieville1997]

Hardware and sensors

Inertial sensors technology has been steadily improving [Yazdi1998] [Barbour2001], enabling new integrated vision and inertial devices [Chalimbaud2005], and also the development of vestibular prostheses for human patients [Wall2003].

Visual Motion and Gaze Control

Comparison of camera rotation estimate given by image optical flow with output from a low cost gyroscope was done for gaze stabilisation of a rotating camera [Panerai1998]. In [Panerai2000] the integration of inertial and visual information in binocular vision systems was studied. In [Panerai2002] the integration of optical flow with inertial sensing is applied to learning of visual stabilisation reflexes in robots with moving eyes. More recently, a high speed gaze control system based on the Vestibulo-Ocular Reflex has been proposed [Viollet2005].

Pose Estimation

A gyroscope sensor was used to discriminate rotation and translation effects on the image and improve the accuracy of 3D shape recovery [Mukai2000]. In [Kurazume2000] inertial sensors are used for image stabilisation and attitude estimation of remote legged robots. In [Rehbinder2003] pose estimation is done using line-based dynamic vision and inertial sensors. In [Grimm2004] the pose of an ergonomic pen-like human-computer interface is tracked in real time using vision and a set of accelerometers.

Virtual and Augmented Reality

Virtual reality applications have always required user motion sensors. Augmented reality, where virtual reality is overlaid onto a realtime view, is particularly sensitive to any mismatch between real and estimated user motion. Precise user attitude and translation can be obtained with serval sensor suits, using external vision and specific markers, radio transponders, ultrasound beacons, laser beacons, etc. Aiming for low cost self contained systems, MEMs inertial sensors are being used in combination with computer vision techniques. The ultimate goal is to have a visuo-inertial tracker that can operate in arbitrary unprepared environments relying on natural features, suitable for augmented reality applications. In [You2001] a two-channel complementary motion extended Kalman filter

is used to combine the low-frequency stability of vision sensors with the high-frequency tracking of gyroscope sensors, hence, achieving stable static and dynamic six-degree-of-freedom pose tracking. Augmented reality systems rely on hybrid trackers to successfully fuse real time imagery with dynamic 3D model [Lang2002] [Neumann2003] [Jiang2004].

Hybrid Trackers

Many hybrid self-trackers based on inertial and vision sensors have been proposed [Hoff1996] [Azuma1999] [Chai2002] [Naimark2002] [Foxlin2003VR] [Ribo2004JRS] [Hogue2004] [Alenya2004JRS] [Klein2004]. The vision tracking relies on either specific targets, line contours or more demanding natural landmarks, and both visual and inertial estimators interact to produce a hybrid tracker. Some commercial hybrid self-tracker systems are being prepared [Foxlin2003VR] [Foxlin2004].

Autonomous Vehicles and Navigation

Vision systems for automated vehicles have also incorporated inertial sensors exploring the benefits of visuo-inertial tracking, in automobiles [Dickmanns1998] [Goldbeck2000], agricultural vehicles [Hague2000], robotic helicopters [Muratet2005] [Corke2004JRS] and other airborne vehicles [Nygards2004JRS] [Graovac2004JRS].

Navigation systems, for which inertial sensors were first developed, also benefit from visual cue integration [Goedeme2004JRS] [Stratmann2004JRS] [Roumeliotis2002] [Diel2005].

Structure and Motion

Structure from motion is a well studied computer vision problem where the integration of inertial sensors can reduce ambiguities and improve robustness [Qian2001] [Qian2002]. The dual problem of motion estimation from observed structure long been pursued, and recent work explores the complementarity of inertial and visual sensing for motion estimation [Jung2001] [Strelow2002] [Strelow2003] [Chroust2004JRS] [Chen2004].

1.5 Overview of the Thesis

In the next chapter we will present the basic entities and sensor models used, establishing the basic data obtainable from the camera sensor and from the inertial sensors. Chapter 3

establishes the relationship between camera and inertial data. There is a close relationship between inertial and visual sensing, both in static and dynamic situations. In this chapter we will explore this data relationship, setting the framework for the applications described in the following chapters. The following chapter focuses on sensor calibration and how cameras and inertial sensors can be cross-calibrated so that they can be used in static and dynamic situations. Simulation and real data results are presented to show the validity and simple requirements of the proposed calibration methods. Chapter 5 presents the use of inertial sensed gravity as a vertical reference for monocular and stereo vision systems, establishing an artificial horizon, enabling segmentation of vertical features, providing restrictions for stereo correspondence of ground plane points and 3D vertical features, and alignment of stereo depth maps with ground segmentation. The next chapter presents a depth map registration and motion segmentation method. Experimental results of stereo depth flow segmentation obtained from a moving observer are presented. The presented work and results are discussed in the last chapter, drawing some conclusions and proposing future work. Appendices provide a summary of the notation used, present the *InerVis WebIndex*, a web site created to support the research in this field, InerVis workshops we organised, and the *InerVis Matlab Toolbox* created to make available the calibration methods presented in this work.

Chapter 2

Vision and Inertial Sensor Models

In this chapter we will present the basic entities and sensor models used. It is important to define the basic data obtainable from the camera sensor and from the inertial sensors, accelerometers and gyroscopes. We start with a brief introduction on inertial sensors and the current MEMS (Micro Electro Mechanical Systems) low cost sensors that enable inertial sensor integration in artificial vision systems. Appendix A provides a summary of the mathematical notation used.

2.1 Inertial Sensors

Gyroscopes and accelerometers are known as inertial sensors since they exploit the property of inertia, i.e., resistance to a change in momentum, to sense angular motion in the case of the gyro, and changes in linear motion in the case of the accelerometer. Inclinometers are also inertial sensors and measure the orientation of the acceleration vector.

From inertial measurements one can only determine an estimate for linear acceleration and angular velocity. Linear velocity and position, and angular position, can be obtained by integration. Inertial navigation systems (INS) implement this process of obtaining velocity and position information from inertial sensor measurements [Lawrence1998]. Internal sensing using inertial sensors is not dependent on any external references, except for the gravity field, which does provide an external reference.

Inertial sensors have long been used for navigation in aerospace and naval applications.

Over the last fifteen years, the electronic and silicon micromachining development, pushed by the needs of the automotive industry, brought about low cost, batch fabricated, silicon sensors [Yazdi1998].

The improvement of surface and bulk micromachining fabrication methods, sensor designs, along with integrated electronics, has produced sensors better performing sensors.

This development has enabled many new applications for MEMS inertial sensors, such as vehicle security systems, sports training devices, computer peripherals, and many other applications where shock, vibration, rotation or tilt sensing is relevant. Single chip inertial systems to be integrated in inertial aided GPS personal navigation systems are being developed [Allen1998].

There are presently three main types of micromachined low cost accelerometers. These are the capacitive, piezoelectric and piezoresistive types. The piezoelectric sensors have a large dynamic range but no DC response, making them unsuitable for inertial navigation systems. In the piezoresistive sensors the acceleration causes a sensing mass to move with respect to a frame, creating stress in a piezoresistor, which changes its resistor value. The capacitive sensors rely on the displacement of capacitive plates due to the acceleration, creating a mismatch in the capacitive coupling. Piezoresistive sensors require bulk micromachining, but capacitive sensors can be surface micromachined providing lower cost sensors will full signal conditioning electronics.

Figure 2.1a shows a dual axis capacitive force balanced accelerometer. In November 2005, Analog Devices introduced the new ADXL330, a 3-axis MEMS accelerometer. Improvement in design and fabrication methods has enabled this low cost, low profile, low power device aiming the vast market of hand-held electronics. A mobile phone with such a sensor can have enhanced data entry and display control, situational awareness and power management.

MEMS gyroscopes have also been implemented, the basic principle of MEMS Vibrating Structure Gyroscopes (VSG) is producing radial linear motion and measuring the Coriolis effect induced by rotation. If a sensing element is made to vibrate in a certain direction, say along the x-axis, rotating the sensor around the z-axis will produce vibration in the y direction with the same frequency. The amplitude of this vibration is determined by the rotation rate. The geometry used takes into account, amongst other factors, the

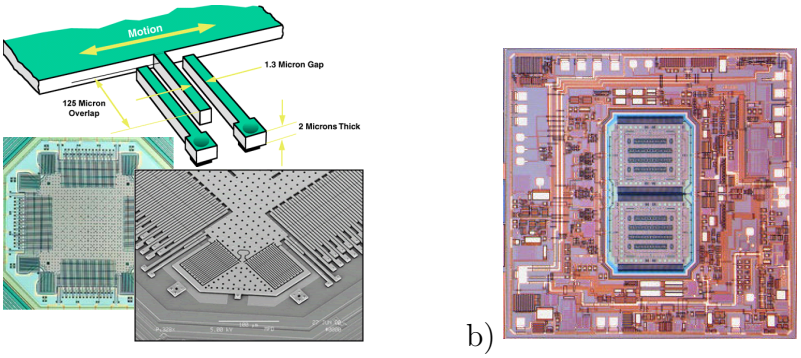


Figure 2.1: a) ADXL202 dual axis 2g capacitive force balanced accelerometer; b) ADXRS150 vibrating structure angular rate sensor; [AnalogDevices].

cancelling out of unwanted accelerations. In October 2002, Analog Devices introduced a MEMS gyroscope (fig.2.1b) with integrated signal processing electronics in a single piece of silicon [AnalogDevices].

The development of MEMS inertial sensors has shown a steady improvement, performance of micromachined gyroscopes has improved by a factor of ten every two years since 1991 [Yazdi1998]. A three-axis $\mu - g$ capacitive accelerometer has been implemented in a single chip hybrid module [Chae2005].

A more detailed overview of micromachined inertial sensors is provided in [Yazdi1998] and [Lobo2002MSc]. In [Barbour2001][Barbour1999] the technology trends in inertial sensors is overviewed, where MEMS sensors show a potential for improvement and growth of applications compared to the high maturity level of other inertial sensor technology. Optical MEMS sensors have been under development for some time, but the small size has limited the successful implementation of these MOEMS (Micro Optical Electromechanical Systems) [Barbour2001]. Currently the technology to make a very small inertial grade gyro and accelerometer does not exist, however interferometric MOEM accelerometer and resonator gyro are expected to meet inertial grade performance for future inertial measurement units [Nayak2005]. Capacitive accelerometer technology is expected to meet medium grade applications, with micro machined accelerometers based on piezoresistive sensing still playing an important role for low cost and moderate performance applications [Nayak2005].

2.2 Data from Inertial Sensors

Inertial sensors provide direct measurements of angular velocity $\boldsymbol{\omega}$ and linear acceleration $\mathbf{a}$. The next table shows the derived quantities that can be obtained by integration or derivation of the measurements from the inertial measuring unit (IMU).

Table 2.1: Data from Inertial Sensors

$\frac{d}{dt}$	angular acceleration rate of linear acceleration (jerk)	$\boldsymbol{\varphi} = \ddot{\boldsymbol{\theta}}$ $\mathbf{j} = \dot{\mathbf{a}} = \dddot{\mathbf{x}}$
	angular velocity linear acceleration + gravity	$\boldsymbol{\omega} = \dot{\boldsymbol{\theta}}$ $\mathbf{a} + \mathbf{g} = \ddot{\mathbf{x}} + \mathbf{g}$
$\int dt$	angular position (attitude) linear velocity	$\boldsymbol{\theta}$ $\mathbf{v} = \dot{\mathbf{x}}$
$\iint dt$	position	$\mathbf{x}$

2.2.1 Vertical Reference from Gravity

The measurements $\mathbf{a}$ taken by the accelerometers in an inertial unit include the sensed gravity vector $\mathbf{g}$ summed with the body's acceleration $\mathbf{a}_b$:

$$\mathbf{a} = -\mathbf{g} + \mathbf{a}_b \tag{2.1}$$

Notice that the accelerometer will measure the reactive (upward) force to gravity. Assuming the system is motionless, then $\mathbf{a}_b = 0$ and the measured acceleration $\mathbf{a}$ gives the gravity vector in the system's frame of reference. So, with a_x,a_y and a_z being the accelerometer measurements along each axis, the vertical unit vector will be given by

$$\hat{\mathbf{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = -\frac{\mathbf{g}}{\|\mathbf{g}\|} = \frac{1}{\sqrt{a_x^2 + a_y^2 + a_z^2}} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{2.2}$$

2.2.2 Inertial Navigation

At the most basic level, an inertial system simply performs a double integration of sensed acceleration over time to estimate position. Assuming a set of accelerometers measuring acceleration along three orthogonal axis we have

$$\boldsymbol{x} = \int \dot{\boldsymbol{x}}\, dt = \iint \ddot{\boldsymbol{x}}\, dt = \iint \boldsymbol{a}_{sensed} dt \tag{2.3}$$

where $\boldsymbol{x}$ is the position, $\dot{\boldsymbol{x}}$ the velocity, and $\ddot{\boldsymbol{x}}$ the acceleration vectors.

But if body rotations occur, they must be taken into account. The measured accelerations are given in the body frame of reference, initially aligned with the navigation frame of reference. In gimballed systems the accelerometers are kept in alignment with the navigation frame of reference, using the gyros to servo a stabilised platform. In strapdown systems the gyros measure the body rotation rate, and the sensed accelerations are computationally converted to the navigation frame of reference. The strapdown system has an Inertial Measurement Unit (IMU) with 3D orthogonal sets of accelerometers and rate gyroscopes. Fig. 2.2 shows a block diagram of a strapdown inertial navigation system.

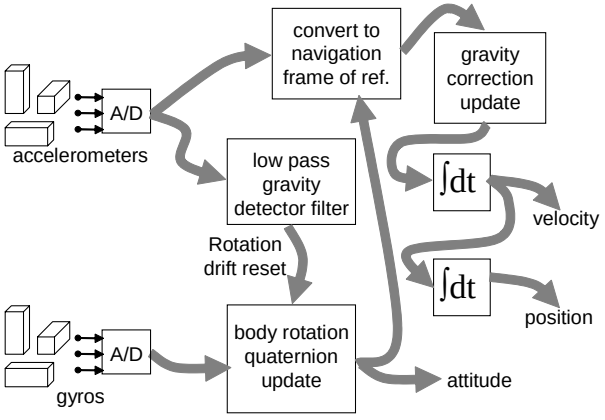


Figure 2.2: Simplified Strapdown Inertial Navigation System

The mechanisation of this rigid body angular motion has to account for the non-commutativity of finite rotations, mathematical singularities and numerical instability. Shuster discusses the various derivations for the rotation vector [Shuster1993] and Savage presents a complete mechanisation using quaternions [Savage1984].

Strap-down systems based on MEMs low-cost inertial sensors offer low performance. To cope with the accumulated drift, some assumptions can be made on the systems's dynamics. If the norm of the sensed acceleration is about 9.8 $m.s^{-2}$ then we can assume that the accelerometers only measure $\boldsymbol{g}$, and the attitude can be directly determined, and reset the accumulated drift in the attitude computation. Assuming pure vibrationless rotations never occur, we could also adjust the gyro offset, since they tend do drift with time and temperature. A low threshold can also be applied to the system, assuming that the system never accelerates or rotates below a certain value, preventing the error accumulation in the rotation update and position integration.

2.2.3 Rotation Update

By performing the rotation update using the IMU gyro data, gravity can be separated from the sensed acceleration.

Quaternions provide a convenient representation for 3D rotations. Quaternion algebra was developed by W. R. Hamilton in the nineteenth century as an extension of imaginary numbers to higher dimensions. A quaternion $\mathring{\mathbf{q}}$ can be written as

$$\mathring{\mathbf{q}} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} = (q_0, \mathbf{q}) \tag{2.4}$$

where q_1, q_2 and q_3 are the components of the imaginary or vector part $\mathbf{q}$ of the quaternion, $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are quaternion vector operators, analogous to unit vectors along orthogonal coordinate axes, and q_0 is the scalar part. The quaternion vector operators, which correspond to the $\mathbf{i}$ in complex numbers, are all square roots of -1.
The magnitude of a quaternion is defined as

$$\|\mathring{\mathbf{q}}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{2.5}$$

The complex conjugate $\mathring{\mathbf{q}}^*$ of quaternion $\mathring{\mathbf{q}}$ is given by

$$\mathring{\mathbf{q}}^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k} = (q_0, -\mathbf{q}) \tag{2.6}$$

and the inverse $\mathring{\mathbf{q}}^{-1}$

$$\mathring{\mathbf{q}}^{-1} = \frac{1}{\mathring{\mathbf{q}}} = \frac{\mathring{\mathbf{q}}^*}{\mathring{\mathbf{q}}\mathring{\mathbf{q}}^*} \tag{2.7}$$

for unit quaternions, *i.e.* $\|\mathring{\mathbf{q}}\| = 1$, $\mathring{\mathbf{q}}\mathring{\mathbf{q}}^* = 1$ and the inverse is the conjugate, $\mathring{\mathbf{q}}^{-1} = \mathring{\mathbf{q}}^*$.

Vectors can be represented by purely imaginary quaternions. A point in space given by the vector $\mathbf{p}$ can be represented by the quaternion $\mathring{\mathbf{p}} = (0, \mathbf{p})$. In our notation, when multiplying vectors with quaternions, the corresponding imaginary quaternion is assumed.

Unit quaternions can be used to represent rotations. The rotation ϕ about a unit vector $\mathbf{u}$ is given by the unit quaternion

$$\mathring{\mathbf{q}} = cos\frac{\phi}{2} + sin\frac{\phi}{2}\mathbf{u} \tag{2.8}$$

and the rotation update for a space point $\mathbf{p}$ is given by

$$\mathbf{p}_{rotated} = \mathring{\mathbf{q}}\mathbf{p}\mathring{\mathbf{q}}^{-1} = \mathring{\mathbf{q}}\mathbf{p}\mathring{\mathbf{q}}^* \tag{2.9}$$

If the quaternion $\mathring{\mathbf{q}}(k)$ represents the body rotation relative to the navigation frame at sample interval k, then the body accelerations can by converted to the navigation frame of reference by

$$\mathbf{a}_{nav} = \mathring{\mathbf{q}}(k)\,\mathbf{a}_{body}\mathring{\mathbf{q}}(k)^* \tag{2.10}$$

The set of orthogonal gyros provide a measurement of the body rotation rate vector given by

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathsf{T}} \tag{2.11}$$

and $\|\boldsymbol{\omega}\| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$ gives the magnitude of the rotation rate and $\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ the unit vector around which the rotation occurs. The rotation increment during a sampling interval Δt is given by the quaternion

$$\Delta\mathring{\mathbf{q}} = cos\left(\frac{\omega\Delta t}{2}\right) - sin\left(\frac{\omega\Delta t}{2}\right)\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \tag{2.12}$$

provided that $\|\boldsymbol{\omega}\| \neq 0$. The quaternion $\mathring{\mathbf{q}}(k)$, that represents the body rotation relative to the navigation frame at sample interval k, can now be updated by

$$\mathring{\mathbf{q}}\,(k+1) = \mathring{\mathbf{q}}\,(k)\,\Delta\mathring{\mathbf{q}} \tag{2.13}$$

and using equation 2.10 the measured body accelerations are converted to the navigation frame, the gravity component is removed, and integration provides body velocity and position in the navigation frame.

2.3 Data from Camera Sensor

Cameras can be seen as ray direction measuring devices. The pinhole camera model considers one center of projection, where all rays originated from world points converge. The image will be equivalent to a plane cutting that pencil of rays, projecting images of world points onto a plane.

If we consider a unit sphere around the optical center we can model the images as being formed on its surface. Using the unit sphere gives an interesting model for central perspective and provides an intuitive visualisation of projective geometry [Kanatani1993] [Stolfi1991]. It also has numerical advantages when considering points at infinity, such as vanishing points. With the spherical model, data from different camera configurations, such as omnidirectional images from catadioptric mirrors or several cameras with a common center of projection, can be incorporated into a unified model, with better spacial observability.

2.3.1 Planar Perspective Model

In the pinhole camera model, shown in fig.2.3, a projection point $\mathbf{p}_i = (u, v)^\mathsf{T}$ in the camera image is related with a 3D point $\mathbf{P} = (X, Y, Z)^\mathsf{T}$ by the perspective relations

$$u = S_u f \frac{X}{Z} + u_0 \qquad v = S_v f \frac{Y}{Z} + v_0 \tag{2.14}$$

where u and v are the pixel coordinates, with the image center given by $(u_0, v_0)^\mathsf{T}$, f is the camera focal distance, S_u and S_v are the scale factors associated with the physical

dimensions of the light sensor picture elements (*pixels*), and $\mathbf{P}$ is expressed in the camera frame of reference.

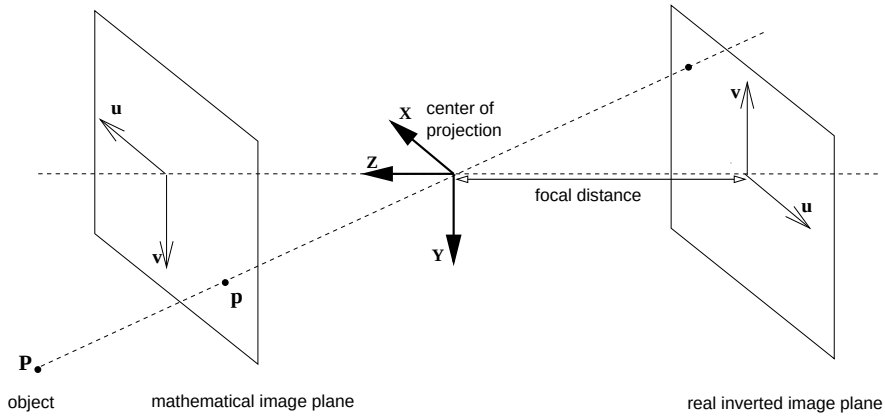


Figure 2.3: Camera perspective projection.

This camera model ignores lens distortion and assumes there is no skew. Assuming that image acquisition maintains square pixel ratio, we can rewrite the above equation as

$$u = f\frac{X}{Z} \qquad v = f\frac{Y}{Z} \tag{2.15}$$

where u and v are the pixel coordinates with origin at the image center and f is the camera effective focal distance (*i.e.*, includes the pixel scale factor). This can be written as a projective mapping, up to scale factor s as

$$s\mathbf{p}_i = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \mathbf{C}\begin{bmatrix} \mathbf{I} & 0 \end{bmatrix}\mathbf{P} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.16}$$

If the 3D point $\mathbf{P} = (X, Y, Z)^\mathsf{T}$ was not given in the camera's frame of reference, $\begin{bmatrix} \mathbf{I} & 0 \end{bmatrix}$ in the above equation would become $\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$ to take into account the rotation $\mathbf{R}$ and translation $\mathbf{t}$ to the cameras frame of reference, *i.e.*, the camera's extrinsic parameters. Matrix $\mathbf{C}$ represents a simplified model of the camera's intrinsic parameters.

The scale factor is arbitrary, and reflects the fact that only the projective ray for each image point is know, and the image plane can be scaled by any non zero s. Since only the

orientation of the projective ray is known, any representation of that orientation is valid. Representing image points by the associated unit vector of their projective ray leads to the unit sphere model presented in the next section.

2.3.2 Unit Sphere Model

If we consider the intersection of projective rays with a unit sphere around the optical center we can model the images as being formed on its surface. The image plane can be seen as a plane tangent to a sphere of radius f, the camera's focal distance, concentric with the unit sphere, as shown in fig. 2.4.

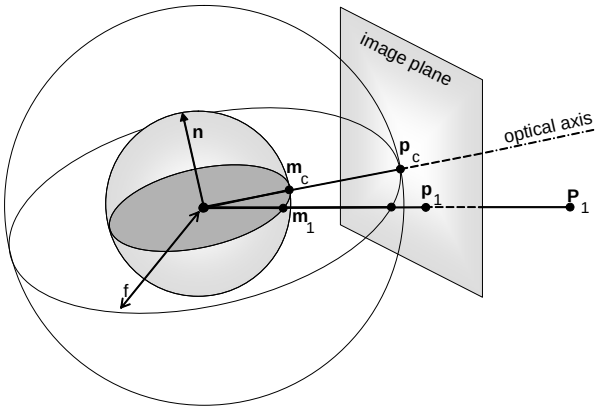


Figure 2.4: Point projection onto Unit Sphere.

Image Points

A world point $\mathbf{P}_i$ will project on the image plane as $\mathbf{p}_i$ and can be represented by the unit vector $\mathbf{m}_i$ placed at the sphere's center, the optical center of the camera, as shown in fig. 2.4. With image centered coordinates $\mathbf{p}_i = (u_i, v_i)$ we have

$$\mathbf{P}_i \rightarrow \mathbf{m}_i = \frac{\mathbf{P}_i}{\|\mathbf{P}_i\|} = \frac{1}{\sqrt{x_i^2 + y_i^2 + f^2}} \begin{bmatrix} u_i \\ v_i \\ f \end{bmatrix} \tag{2.17}$$

Note that $\mathbf{m} = (m_1, m_2, m_3)^\mathsf{T}$ is a unit vector and the projection is not defined for $\mathbf{P} = (0, 0, 0)^\mathsf{T}$. Projection onto the unit sphere is related to projection onto a plane by

$$(u, v)^{\mathsf{T}} = \left(f\frac{m_1}{m_3}, f\frac{m_2}{m_3} \right)^{\mathsf{T}} \tag{2.18}$$

Given f, the projection to a sphere can be computed from the projection to a plane and conversely. To avoid ambiguity m_3 is forced to be positive, so that only points on the image side hemisphere are considered.

Image Lines

Image lines can also be represented in a similar way. Any image line defines a plane with the center of projection, as shown in fig. 2.5.

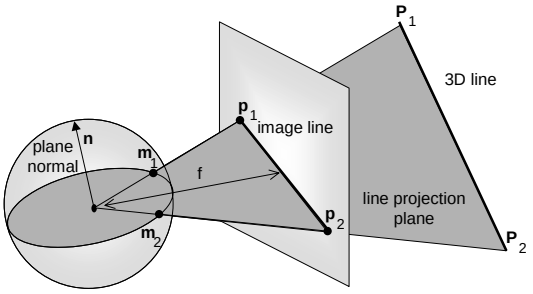


Figure 2.5: Line projection onto Unit Sphere.

A vector $\mathbf{n}$ normal to this plane uniquely defines the image line and can be used to represent the line. For a given image line $ax + by + c = 0$, the unit vector is given by

$$\mathbf{n} = \frac{1}{\sqrt{a^2 + b^2 + (c/f)^2}} \begin{bmatrix} a \\ b \\ c/f \end{bmatrix} \tag{2.19}$$

We can write the unit vector of an image line with points $\mathbf{m}_1$ and $\mathbf{m}_2$ as

$$\mathbf{n} = \mathbf{m}_1 \times \mathbf{m}_2 \tag{2.20}$$

Vanishing Points

Since the perspective projection maps a 3D world onto a plane or planar surface, phenomena that only occur *at infinity* will project to very finite locations in the image. Parallel

lines only meet at infinity, but in the image plane, the point where they meet can be quite visible and is called the *vanishing point* of that set of parallel lines.

A space line with the orientation of an unit vector $\boldsymbol{m}$ has, when projected, a *vanishing point* with unit sphere vector $\pm\boldsymbol{m}$, as shown in fig. 2.6. Since the vanishing point is only determined by the 3D orientation of the space line, projections of parallel space lines intersect at a common vanishing point.

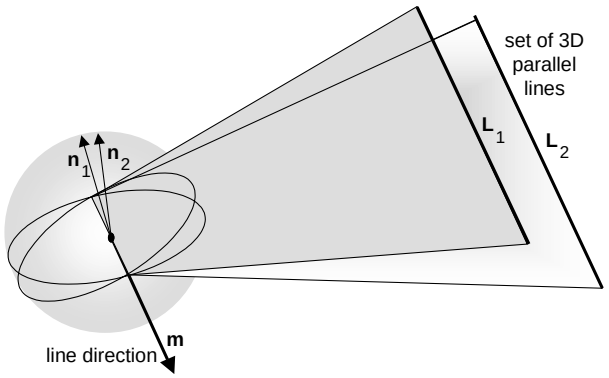


Figure 2.6: Vanishing point of a set of 3D parallel lines.

As seen in fig. 2.6, the normals to the line projection planes will all lie in the same plane, orthogonal to the vanishing point $\boldsymbol{m}$.

The vanishing point of a set of 3D parallel lines with image lines $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ is given by

$$\boldsymbol{m} = \boldsymbol{n}_1 \times \boldsymbol{n}_2 \tag{2.21}$$

2.4 Summary/Conclusions

The current MEMS low cost sensors enable inertial sensor integration in artificial vision systems. We presented the basic data obtainable from the camera sensor and from the inertial sensor. Appendix A provides a summary of the mathematical notation used.

Chapter 3

Camera and Inertial Data Relationship

Inertial and vision are two distinct sensing modalities, but when both observe the world, their data has some interesting relationships. Inertial sensors coupled to cameras can provide valuable data about camera ego-motion and how world features are expected to be oriented. These pose and motion parameters can also be inferred from the image flow and known scene features. In this chapter we will explore this data relationship, setting the framework for the applications described in the following chapters.

3.1 Camera Attitude and Static Inertial Cues

How does gravity show up in the camera?

A static camera is capable of observing one important inertial cue: gravity. The vertical vanishing point of any vertical world features defines the gravity reference for the camera. The image horizon line in another cue for camera attitude. The path of objects in free fall or ballistic motion also provide a vertical reference.

Figure 3.1 depicts some of these visual gravity cues. With some prior knowledge about expected scene features, the visual gravity cues can be detected and a vertical reference defined for the camera.

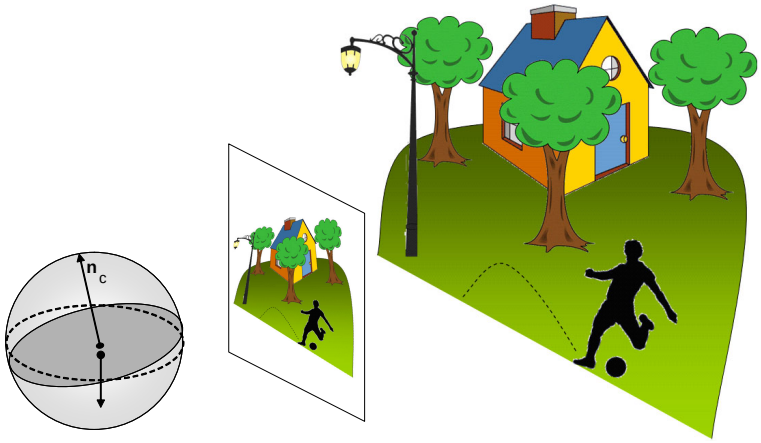


Figure 3.1: Gravity cues in the camera captured image.

3.1.1 Vanishing Point of Vertical Lines

As we saw in the previous chapter, parallel lines in the world define vanishing points in the image plane, than can be easily represented i the unit sphere model. Figure 3.2 shows how a set of vertical lines, which are near parallel in the image plane, define a unit sphere vector for the vertical.

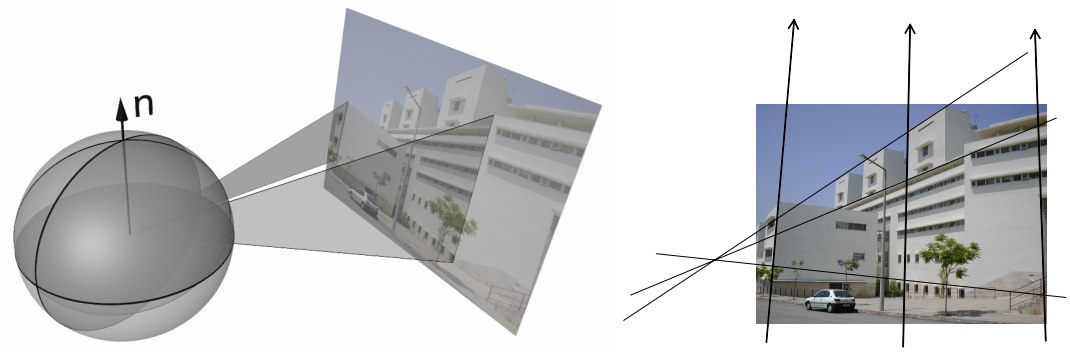


Figure 3.2: Vertical reference from vanishing point of a set of 3D vertical lines.

As seen in fig. 3.3, at set of 3D vertical lines will define normals to the line projection planes within same plane horizontal plane, orthogonal to the vertical vanishing point.

The vanishing point of a set of 3D vertical lines with image lines $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ is given

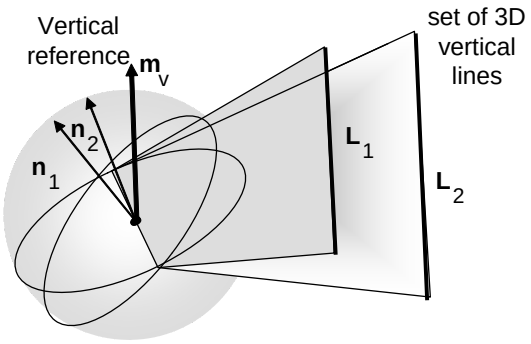


Figure 3.3: Vertical reference orthogonal to vertical line projection plane normals.

by

$$\boldsymbol{m}_v = \boldsymbol{n}_1 \times \boldsymbol{n}_2 \tag{3.1}$$

The vertical reference $\hat{\boldsymbol{n}}$ corresponds to the *north pole* of the unit sphere. A set of world vertical features will project to image lines $\boldsymbol{n_i}$ with a common vanishing point $\boldsymbol{m_{vp}} = \hat{\boldsymbol{n}}$.

With appropriate vertical line detection, $\boldsymbol{m}_v$ provides a vertical reference in the camera frame of reference. Alternatively, detecting sets of horizontal parallel lines, or the image horizon, leads to the same vertical reference, providing a common point with static inertial sensors that only detect gravity.

3.1.2 Horizon Line

Figure 3.4 shows how a set of parallel lines that define a vanishing point that belong to the horizon line.

The horizon line can be found by having two distinct vanishing points of a levelled plane as seen in figure 3.5. Knowing the vertical in the camera's referential and the focal distance, an artificial horizon line also can also be traced with a single vanishing point. A planar surface with a unit normal vector $\hat{\boldsymbol{n}}$, not parallel to the image plane has, when projected, a *vanishing line* given by

$$n_x u + n_y v + n_z f = 0 \tag{3.2}$$

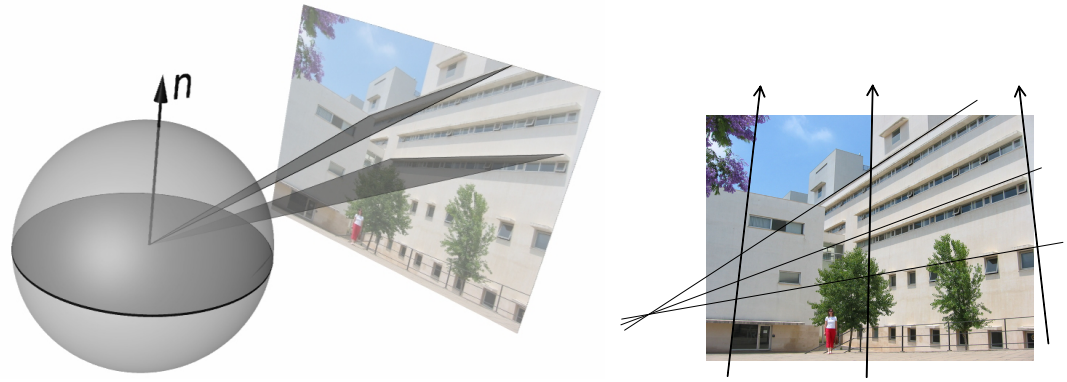


Figure 3.4: Vanishing point of a set of 3D parallel horizontal lines.

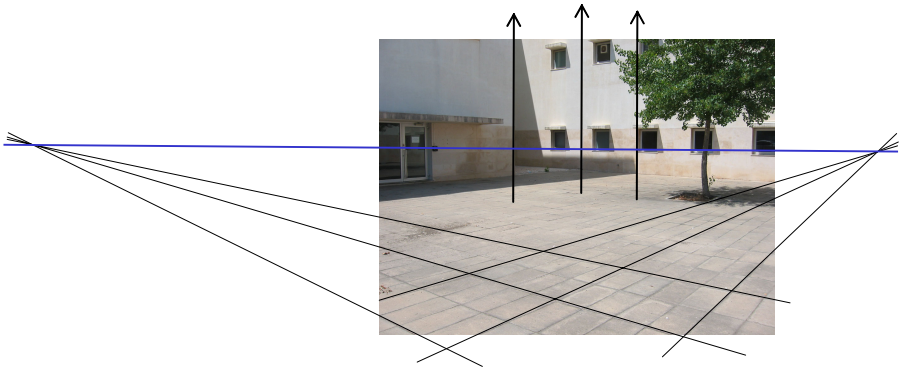


Figure 3.5: Vanishing points of two sets of 3D parallel horizontal lines defining the horizon line.

where f is the focal distance, u and v image coordinates and $\hat{\boldsymbol{n}} = (n_x, n_y, n_z)^{\mathsf{T}}$. Since the vanishing line is determined alone by the orientation of the planar surface, the horizon line is the vanishing line of all levelled planes, parallel to the ground plane.

3.2 Camera and Motion

How does linear and angular motion show up in the camera?

A moving camera is capable of observing important ego-motion cues. The extent to which they are useful will depend on the visual sensor response time, and the *a priori* knowledge about the observed scene and its structure. Taking into account successive

image frames, image patterns will change due to scene or camera motion. Even a single frame blurred image can provide a motion cue.

3.2.1 Ego Motion and Spherical Motion Field

When the camera sensor moves relative to the observed scene, image features will have a corresponding motion across the image.

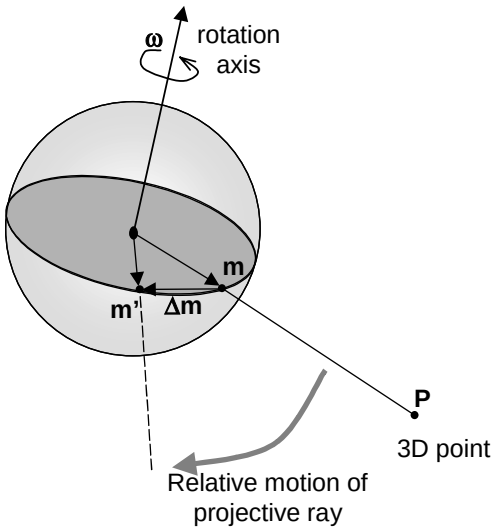


Figure 3.6: Projected unit sphere point motion with camera pure rotation.

If the camera experiences a pure rotation $\boldsymbol{\omega}$, the fixed world $\boldsymbol{P}_i$ given in the camera referential will have a motion vector given by

$$\dot{\boldsymbol{P}}_i = -\boldsymbol{\omega} \times \boldsymbol{P}_i \tag{3.3}$$

as shown in fig. 3.6. The world point after the rotation $\boldsymbol{P}'_i$ is given by $\boldsymbol{P}_i - \boldsymbol{\omega} \times \boldsymbol{P}_i$. The unit sphere point after the rotation $\boldsymbol{m}'_i$ is given by

$$\boldsymbol{m}'_i = \frac{\boldsymbol{P}_i - \boldsymbol{\omega} \times \boldsymbol{P}_i}{\|\boldsymbol{P}_i - \boldsymbol{\omega} \times \boldsymbol{P}_i\|} = \boldsymbol{m}_i - \boldsymbol{\omega} \times \boldsymbol{m}_i \tag{3.4}$$

Since the rotation is centered in the camera projective center, the induced image motion does not depend on the 3D point depth.

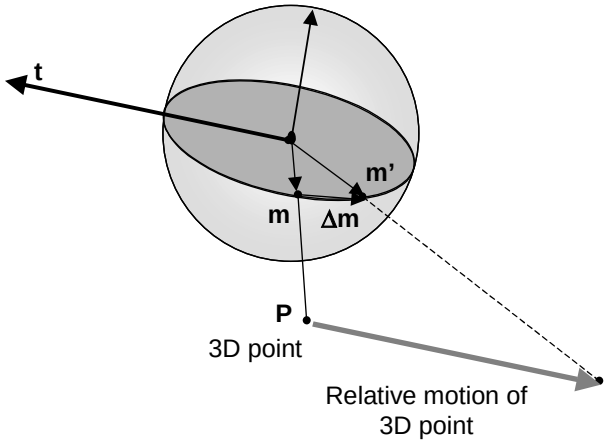


Figure 3.7: Projected unit sphere point motion with camera translation.

If the camera experiences both rotation $\boldsymbol{\omega}$ and translation $\boldsymbol{t}$ the fixed world $\boldsymbol{P}_i$ given in the camera referential will have a motion vector given by

$$\dot{\boldsymbol{P}}_i = -\boldsymbol{t} - \boldsymbol{\omega} \times \boldsymbol{P}_i \tag{3.5}$$

as shown in fig. 3.7. Projecting onto the unit sphere as before, the motion field on the unit sphere $\dot{\boldsymbol{m}}_i$ is given by

$$\dot{\boldsymbol{m}}_i = \frac{1}{\|\boldsymbol{P}_i\|}((\boldsymbol{t}.\boldsymbol{m}_i)\boldsymbol{m}_i - \boldsymbol{t}) - \boldsymbol{\omega} \times \boldsymbol{m}_i \tag{3.6}$$

This equation describes the velocity vector $\dot{\boldsymbol{m}}_i$ for a given unit sphere point $\boldsymbol{m}_i$ as a function of camera ego motion $(\boldsymbol{t}, \boldsymbol{\omega})$ and depth $\|\boldsymbol{P}_i\|$.

3.2.2 Image Focus of Expansion and Contraction

When the camera is moving with linear velocity $\boldsymbol{t}$ and not rotating, from (3.6) we see that the image point

$$\boldsymbol{m}_{FOE} = \frac{\boldsymbol{t}}{\|\boldsymbol{t}\|} \tag{3.7}$$

will have no motion, i.e., $\dot{\boldsymbol{m}}_{\boldsymbol{FOE}} = 0$, and all others will be expanding or contracting to this point. This point is known as the the image focus of expansion (FOE). When the

system is also rotating, the FOE will have depth independent velocity

$$\dot{\boldsymbol{m}}_{FOE} = -\boldsymbol{\omega} \times \boldsymbol{m}_{FOE} = -\boldsymbol{\omega} \times \frac{\boldsymbol{t}}{\|\boldsymbol{t}\|} \tag{3.8}$$

3.2.3 Image Center of Rotation

When the camera is moving with angular velocity $\boldsymbol{t}$ and no linear translation, from (3.6) we see that the image point

$$\boldsymbol{m}_{COR} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \tag{3.9}$$

will have no motion, i.e., $\dot{\boldsymbol{m}}_{\boldsymbol{COR}} = 0$, and all others will be rotating around this point. This point is known as the the image center of rotation (COR). When the system is also translating at velocity $\boldsymbol{t}$, the COR will have depth dependent velocity

$$\dot{\boldsymbol{m}}_{COR} = \frac{1}{\|\boldsymbol{P}_{FOE}\|}((\boldsymbol{t}.\boldsymbol{m}_{COR})\boldsymbol{m}_{COR} - \boldsymbol{t}) \tag{3.10}$$

where $\boldsymbol{P}_{FOE}$ in the 3D point in view along the image ray given by $\boldsymbol{m}_{COR}$.

The definition of the FOE and COR can be useful during visual based navigation tasks.

3.2.4 Optical Flow

Optical flow is the apparent motion of brightness patterns in the image. Generally, optical flow corresponds to the motion field, but not always. Shading, changing lighting and some texture patterns might induce optical field different from the motion field. However since what can be observed is the optical field, the assumption is made that optical flow field provides a good estimate for the true projected motion field.

Optical flow computation can be made in a *dense* way, by estimating motion vectors for every image pixel, or *feature based*, estimating motion parameters only for matched features.

To compute spherical optical flow, the sensed image could be re-sampled onto the sphere surface, but this would introduce unwelcome artifacts. A better approach is to perform the optical flow computation in the sensed image domain and map the optical flow field to the unit sphere using the Jacobian of the transformation [Gluckman1998].

Representing the pixel intensity in the planar image sequence by $I(u, v, t)$, where $(u, v)^\mathsf{T}$ are image centered pixel coordinates at time t, and the velocity of an image pixel $\boldsymbol{p}$ as $\boldsymbol{v}_p = \dot{\boldsymbol{p}} = (\frac{du}{dt}dt, \frac{du}{dt}dt)^\mathsf{T}$, the brightness constancy constraint says that the projection of a world point has a constant intensity over a short interval of time, i.e., assuming that the pixel intensity or brightness is constant during dt, we have

$$I(u + \frac{du}{dt}dt, v + \frac{dv}{dt}dt, t + dt) = I(u, v, t) \tag{3.11}$$

If the brightness changes smoothly with u, v and t, we can expand the left-hand-side by Taylor series to

$$I(v, u, t) + \frac{\partial I}{\partial u}\frac{du}{dt}dt + \frac{\partial I}{\partial v}\frac{dv}{dt}dt + \frac{\partial I}{\partial t}dt + O(dt^2) = I(u, v, t) \tag{3.12}$$

ignoring the higher order terms we have

$$\frac{\partial I}{\partial x}\frac{du}{dt}dt + \frac{\partial I}{\partial y}\frac{dv}{dt}dt + \frac{\partial I}{\partial t}dt = 0 \tag{3.13}$$

i.e.,

$$\nabla \boldsymbol{I} \cdot \boldsymbol{v}_p + \frac{\partial \boldsymbol{I}}{\partial t}dt = 0 \tag{3.14}$$

where $\nabla \boldsymbol{I}$ is the image gradient at pixel $\boldsymbol{p}$. These spatial and time derivatives can be estimated using a convolution kernel on the image frames.

But for each pixel we only have one constraint equation, and two unknowns. Only the *normal flow* can be determined, i.e., the flow along the direction of image gradient. The flow on the tangent direction of an isointensity contour can not be estimated. This is the so called *aperture problem*. To determine optical flow uniquely additional constraints are needed.

The problem is that a single pixel cannot be tracked, unless it has a distinctive brightness with respect to all of its neighbours. If a local window of pixels is used, a local constraint can be added, i.e., single pixels will not be tracked, but windows of pixels instead.

Barron *et al.* [Barron1994] present a quantitative evaluation of optical flow techniques, including the Lucas-Kanade method, that uses local consistency to overcome the aperture

problem [Lucas1981]. The assumption is made that a constant model can be used to describe the optical flow in a small window.

The assumption is made that a constant model can be used to describe the optical flow in a small window Ω. A weighted least-squares fit of all local first-order brightness constraints (3.14) is made to this constant $\boldsymbol{v}_p$, $\boldsymbol{p} \in \Omega$, by minimising

$$\sum_{\boldsymbol{p}\in\Omega} W^2(\boldsymbol{p})(\nabla \boldsymbol{I} \cdot \boldsymbol{v}_p + \frac{\partial I}{\partial t})^2 \tag{3.15}$$

where $W(\boldsymbol{p})$ is a window weighing function to favour the center pixels.

The optical flow for image pixel $\boldsymbol{p}$ is given by

$$\boldsymbol{v}_p = (\boldsymbol{A}^\mathsf{T}\boldsymbol{W}^2\boldsymbol{A})^{-1}\boldsymbol{A}^\mathsf{T}\boldsymbol{W}^2\boldsymbol{b} \tag{3.16}$$

which is solved in closed from when $\boldsymbol{A}^\mathsf{T}\boldsymbol{W}^2\boldsymbol{A}$ is not singular. Taking into account the numerical stability of the inverse, we can reject bad cases and obtain sparse optical flow.

The obtained sparse optical flow field on a planar image can be mapped to the unit sphere using the Jacobian of the sensed image to spherical image mapping [Gluckman1998]. Considering the case of a planar image, we have to differentiate the unit sphere coordinates $\boldsymbol{m} = (m_x, m_y, m_z)^\mathsf{T}$ with respect to image coordinates $\boldsymbol{p} = (u, v)^\mathsf{T}$ to obtain the Jacobian

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial m_x}{\partial u} & \frac{\partial m_x}{\partial v} \\ \frac{\partial m_y}{\partial u} & \frac{\partial m_y}{\partial v} \\ \frac{\partial m_z}{\partial u} & \frac{\partial m_z}{\partial v} \end{bmatrix} \tag{3.17}$$

From the unit sphere projection (2.17) we get

$$\boldsymbol{J} = \begin{bmatrix} \frac{1}{\sqrt{u^2+v^2+f^2}} & \frac{1}{v}\frac{u}{\sqrt{u^2+v^2+f^2}} \\ \\ \frac{1}{u}\frac{v}{\sqrt{u^2+v^2+f^2}} & \frac{1}{\sqrt{u^2+v^2+f^2}} \\ \\ \frac{1}{u}\frac{f}{\sqrt{u^2+v^2+f^2}} & \frac{1}{v}\frac{f}{\sqrt{u^2+v^2+f^2}} \end{bmatrix} \tag{3.18}$$

The planar optical flow field $\boldsymbol{v}_p$ is mapped to the spherical optical flow field $\boldsymbol{v}_m$ by

$$\boldsymbol{v}_m = \begin{bmatrix} \frac{\partial m_x}{\partial t} \\ \frac{\partial m_y}{\partial t} \\ \frac{\partial m_z}{\partial t} \end{bmatrix} = \boldsymbol{J}\boldsymbol{v}_p = \boldsymbol{J} \begin{bmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \end{bmatrix} \tag{3.19}$$

If a different image projection geometry is used, a different $\boldsymbol{J}$ must be considered.

3.3 Frames of Reference

When combining the two sensing modalities, the frame of reference in which sensor measurements are made need to be taken into account. The sensor observed features, visual or inertial, also have implicit or explicit frames of reference to be considered. Figure 6.1 shows the several frames of reference that can be defined. Considering a moving observer with a visual and inertial sensors rigidly mounted, we have the camera $\{\mathcal{C}\}$, IMU $\{\mathcal{I}\}$, world aligned mobile system $\{\mathcal{N}\}$, and world fixed $\{\mathcal{W}\}$ frames of reference. The gravity field directly sensed by the inertial sensors, and indirectly from visual vertical features by the camera, provide some external references that help in obtaining a world aligned moving frame of reference, or navigation frame $\{\mathcal{N}\}$, and after motion compensation the world fixed $\{\mathcal{W}\}$ frame of reference.

3.4 Inertial Data in Camera Frame of Reference

The visual processing has to consider the motion parameters of the camera center of projection. Since the inertial measurements performed by the inertial sensors are given in

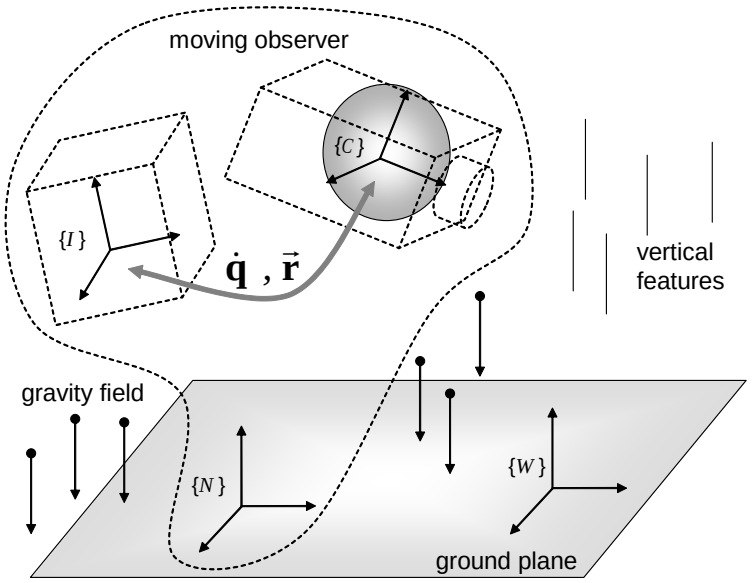


Figure 3.8: Camera $\{\mathcal{C}\}$, IMU $\{\mathcal{I}\}$, world aligned mobile system $\{\mathcal{N}\}$, and world fixed $\{\mathcal{W}\}$ frames of reference

the IMU frame of reference $\{\mathcal{I}\}$ and not in the camera frame of reference $\{\mathcal{C}\}$, the rigid body transformation between the two has to be taken into account. This transformation can be expressed by the unit quaternion $\mathring{\mathbf{q}}$ that rotates inertial measurements in the inertial sensor frame of reference $\{\mathcal{I}\}$ to the camera frame of reference $\{\mathcal{C}\}$, and translation vector $\mathbf{r}$. In the following sections the inertial sensed measurements are expressed in the camera frame of reference.

3.4.1 Non-rotating Camera Linear Acceleration

If a rigid body has no angular velocity, any point within will have the same linear acceleration. As shown in fig. 3.9, to report the inertial sensed acceleration to the camera center of projection, i.e., to have ${}^{\mathcal{C}}\mathbf{a}$, we just apply the known rotation between the two frames of reference, $\mathring{\mathbf{q}}$, i.e.,

$$ {}^{\mathcal{C}}\mathbf{a} = \mathring{\mathbf{q}}\, {}^{\mathcal{I}}\mathbf{a}\, \mathring{\mathbf{q}}^{*} \tag{3.20} $$

where ${}^{\mathcal{I}}\mathbf{a}$ is the sensed acceleration in the IMU frame of reference.

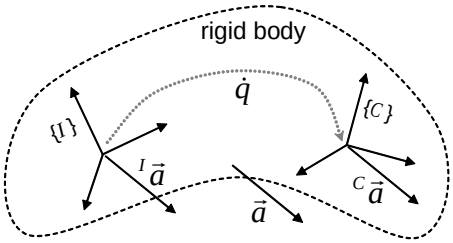


Figure 3.9: Inertial sensed acceleration in non-rotating camera frame of reference.

Camera Gravity Vertical Reference

The gravity vertical reference, given in 2.2 for the IMU frame of reference $\{\mathcal{I}\}$, is simply given by

$$^{\mathcal{C}}\hat{\mathbf{n}} = \mathring{\mathbf{q}}\,^{\mathcal{I}}\hat{\mathbf{n}}\,\mathring{\mathbf{q}}^{*} \tag{3.21}$$

3.4.2 Rotating Camera Angular Velocity

Any point of a rigid rotating body has the same angular velocity. As shown in fig. 3.10, to obtain the camera angular velocity in the camera frame of reference, $^{\mathcal{C}}\boldsymbol{\omega}$, we again just apply the known rotation between the two frames of reference:

$$^{\mathcal{C}}\boldsymbol{\omega} = \mathring{\mathbf{q}}\,^{\mathcal{I}}\boldsymbol{\omega}\,\mathring{\mathbf{q}}^{*} \tag{3.22}$$

where $^{\mathcal{I}}\boldsymbol{\omega}$ is the sensed angular velocity in the IMU frame of reference.

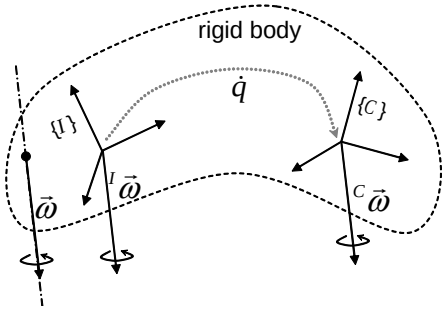


Figure 3.10: Inertial sensed angular velocity in camera frame of reference.

However, the above formulation does not take into account the center of rotation. In fig. 3.10 the center of rotation is shown to be within the rigid body, but it could be anywhere. We can always model the rigid body motion as rotating about its center of mass and experiencing centripetal acceleration with respect to the true center of rotation. As we will see below, this adds some complexity when reporting inertial measurements from one frame of reference to another.

3.4.3 Rotating Camera Linear Acceleration

If a rigid body has no angular velocity, any point within will have the same linear acceleration. But if the rigid body is rotating about some axis, a centripetal acceleration, proportional to the perpendicular distance to the rotation axis, will be added. As shown in fig. 3.13, the linear acceleration of both camera and IMU will have a component due to the rotation about some axis, so when reporting inertial sensor observations to the camera frame of reference they must be taken into account, i.e.,

$$^{\mathcal{C}}\mathbf{a} = \mathring{\mathbf{q}}\,(^{\mathcal{I}}\mathbf{a} - {}^{\mathcal{I}}\mathbf{a}_c)\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\mathbf{a}_c \tag{3.23}$$

where $^{\mathcal{I}}\mathbf{a}_c$ is the IMU centripetal acceleration, and $^{\mathcal{C}}\mathbf{a}_c$ the camera centripetal acceleration, both relative to some rotation axis.

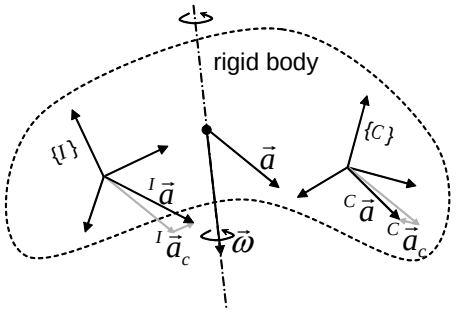


Figure 3.11: Inertial sensed acceleration in rotating camera frame of reference.

The rotation axis must be fixed relative to an inertial frame of reference, i.e., a non-accelerating non-rotating frame of reference. In other words, the inertial sensor measures the centripetal acceleration relative to the true rotation axis, and not relative to say the system center of mass, which would not be fixed relative to an inertial frame of reference.

In general, centripetal acceleration $\mathbf{a}_c$ at a point $\mathbf{r}$ with the origin on the rotation axis is given by

$$\mathbf{a}_c = \boldsymbol{\omega} \times \mathbf{v}_t = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \tag{3.24}$$

where $\boldsymbol{\omega}$ is the angular velocity and $\mathbf{v}_t$ is the tangential velocity.

If we assume that the rotation axis goes through the camera center of projection, than it will not have centripetal acceleration and its linear acceleration is given by

$$\begin{aligned} {}^{\mathcal{C}}\mathbf{a} &= \mathring{\mathbf{q}}\,({}^{\mathcal{I}}\mathbf{a} - {}^{\mathcal{I}}\mathbf{a}_c\,)\mathring{\mathbf{q}}^* \\ &= \mathring{\mathbf{q}}\,({}^{\mathcal{I}}\mathbf{a} - {}^{\mathcal{I}}\boldsymbol{\omega} \times ({}^{\mathcal{I}}\boldsymbol{\omega} \times {}^{\mathcal{I}}\mathbf{r})\,)\mathring{\mathbf{q}}^* \\ &= \mathring{\mathbf{q}}\,{}^{\mathcal{I}}\mathbf{a}\,\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\boldsymbol{\omega} \times ({}^{\mathcal{C}}\boldsymbol{\omega} \times {}^{\mathcal{C}}\mathbf{r}) \end{aligned} \tag{3.25}$$

where ${}^{\mathcal{I}}\mathbf{r}$ is the translation from the IMU to the camera in the IMU frame of reference, ${}^{\mathcal{C}}\mathbf{r}$ is the translation from the camera to the IMU in the camera frame of reference, and $\mathring{\mathbf{q}}\,{}^{\mathcal{I}}\mathbf{r}\,\mathring{\mathbf{q}}^* = -{}^{\mathcal{C}}\mathbf{r}$.

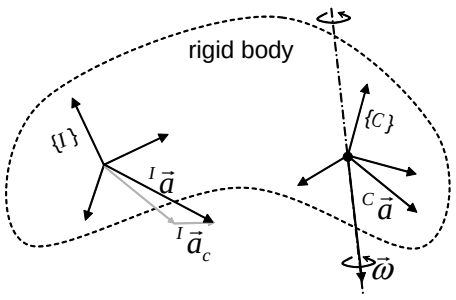


Figure 3.12: Inertial sensed acceleration in camera frame of reference, with rotation about the camera.

If we assume that the rotation axis goes though the IMU center, than no centripetal acceleration will be sensed, and the camera linear acceleration is given by

$$\begin{aligned} {}^{\mathcal{C}}\mathbf{a} &= \mathring{\mathbf{q}}\,({}^{\mathcal{I}}\mathbf{a},\,)\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\mathbf{a}_c \\ &= \mathring{\mathbf{q}}\,{}^{\mathcal{I}}\mathbf{a}\,\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\boldsymbol{\omega} \times ({}^{\mathcal{C}}\boldsymbol{\omega} \times (-{}^{\mathcal{C}}\mathbf{r})) \\ &= \mathring{\mathbf{q}}\,{}^{\mathcal{I}}\mathbf{a}\,\mathring{\mathbf{q}}^* - {}^{\mathcal{C}}\boldsymbol{\omega} \times ({}^{\mathcal{C}}\boldsymbol{\omega} \times {}^{\mathcal{C}}\mathbf{r}) \end{aligned} \tag{3.26}$$

From the above derivation, the knowledge of the rotation axis is crucial to describe the absolute motion of all the points within a rigid body, i.e., the motion relative to an inertial frame.

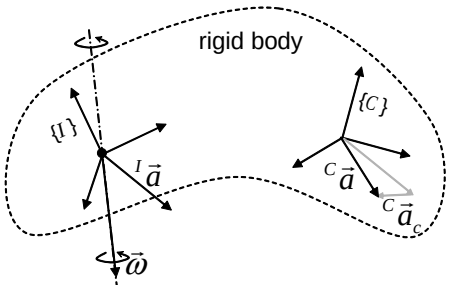


Figure 3.13: Inertial sensed acceleration in camera frame of reference, with rotation about the IMU.

Inertial navigation systems rely on the path taken from a known initial position to report to an external reference. In other words, we might know how to describe the motion of a point, and hence the whole rigid body, by integrating the measured acceleration at a given point, linear plus centripetal, with the appropriate rotation update from the gyros. But if the initial position is not known, we are not able to determine the rotation axis, and correctly report centripetal acceleration to the camera.

Consider a rotating rigid body with distributed tri-axial accelerometers as shown in fig. 3.14. Each sensor will measure a different resultant acceleration determined by its relative position to the axis of rotation.

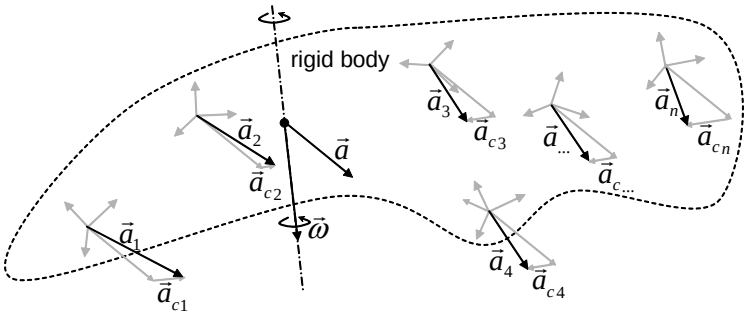


Figure 3.14: Rotating rigid body with distributed tri-axial accelerometers to find rotation axis.

If the rigid body has a pure rotation, i.e., $\mathbf{a} = 0$, than each sensor will only measure the centripetal acceleration $\mathbf{a}_i = \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_i)$.

As we will se in the following sections, gravity can be used as a common reference to

calibrate relative rotation between sensors. In this case taking sets of measurements over several static poses, the relative rotation between the several tri-axial accelerometers can be determined. A single triad of gyros can be used to measure angular velocity, and the estimated frame relative rotations used to obtain each $\boldsymbol{\omega}_i$.

3.5 Summary/Conclusions

There is a close relationship between inertial and visual sensing, both in static and dynamic situations. Gravity is a static inertial cue, also perceived by the camera as image horizon and vertical features from a gravity structured world. A moving camera will have an induced visual flow determined by the motion parameters also sensed by the inertial sensors.

When reporting inertial measurements to the camera frame of reference, the rigid transformation between the sensors has to be taken into account, the most important being the rotation. The translation between the two will not be important in some applications, but if the inertial sensor is attached to the camera system with a significant lever arm, it will have to be taken into account for fast motions.

The rigid body transformation between the IMU and the camera has to be calibrated, but direct physical measurements are difficult to perform, since the camera center of projection and inertial sensor sensing point and axis are not obvious. But rotation $\mathring{\mathbf{q}}$ and translation $\mathbf{r}$ can be derived from (3.22) and (3.26) provided something is known about the motion. Using the gravity reference, the rotation $\mathring{\mathbf{q}}$ can be estimated using a simple boresight approach, as described in the next chapter.

Chapter 4

Calibration

We now focus on how cameras and inertial sensors can be cross-calibrated so that they can be used in static and dynamic situations. The rotation between the camera and the inertial sensor can be estimated, when calibrating the camera, by having both sensors observe the vertical direction, using a vertical chessboard target and gravity. The translation between the two can be estimated using a simple passive turntable and static images, provided that the system can be adjusted to turn about the inertial sensor null point in several poses. Simulation and real data results are presented to show the validity and simple requirements of the proposed method.

4.1 Introduction

As our work proposes, inertial sensors coupled to cameras can provide valuable data about camera ego-motion and how world features are expected to be oriented. However only with appropriate sensor calibration can the two sensing modalities be integrated and used in artificial perception systems.

The rotation between the camera and the inertial sensor can be estimated by having both sensors observe the vertical direction, using a vertical visual target for the camera, and gravity for the inertial sensors. Standard camera calibration can be performed on the same set of images, both using the same visual target, such as a vertical chessboard target, simplifying the whole calibration procedure.

The translation between the two will not be important in some applications, but if the inertial sensor is attached to the camera system with a significant lever arm, it will have to be taken into account for fast motions. Using a simple passive turntable, and positioning the integrated camera and inertial system centered on the inertial sensor, the lever arm can be estimated. Observing the inertial sensor outputs, the system can be adjusted to turn about their null point in several poses. The lever arm can than be estimated from static images of a suitably placed visual target before and after each rotation.

The problem of estimating the rotation between the inertial sensor and the camera is a particular case of the well-known orthogonal Procrustes method for 3D attitude estimation [Dorst2005]. Instead of having two sets of points we have two sets of unit vectors corresponding to the observed vertical in each sensor at several poses. In our work we used the unit quaternion derivation of the method [Horn1987].

Standard hand-eye calibration [Tsai1989][Daniilidis1999] can be applied to estimate translation, using the approach of rotating about the inertial sensor center. However, since the target is being repositioned after each turn, the method is not applied to the full data set like in traditional hand-eye calibration. We used an implementation of the full hand-eye calibration [Tsai1989] to provide a comparison in the results using only a camera with fixed lever arm, by keeping a constant pivot point.

Closely related to our work, Lang and Pinz concurrently presented a method for 3-axis inertial sensor calibration based on model fitting, and a method to find the rotation between vision and inertial system based on rotation differences [Lang2005]. However their rotation estimation requires motion and is more complex than ours, although suitable for any tracking system that delivers relative or absolute orientation of the moving target. Since the common reference is the observed/sensed gravity, our boresight approach is more direct, simpler to perform and accurate.

4.2 Stand Alone Sensor Calibration

Before we consider the cross-calibration of cameras and inertial sensors, we will take a look at how each sensor can be calibrated individually.

4.2.1 Camera Calibration

Camera calibration has been extensively studied, and standard techniques established. For this work, camera calibration was performed using the Camera Calibration Toolbox for Matlab [Bouguet2006]. The C implementation of this toolbox is included in the Intel Open Source Computer Vision Library [Intel2006].

The calibration uses images of a chessboard target in several positions and recovers the camera's intrinsic parameters, as well as the target positions relative to the camera, as shown in fig. 4.1.

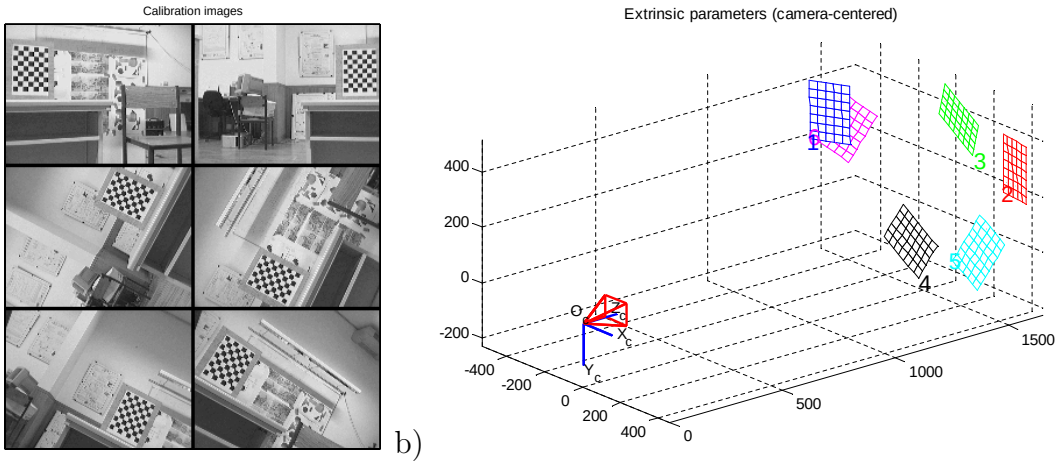


Figure 4.1: a) Images with vertical chessboard target used for calibration. b) Reconstructed target positions relative to the camera.

The calibration algorithm is based on Zhang's work in estimation of planar homographies for camera calibration [Zhang1999], but the closed-form estimation of the internal parameters from the homographies is slightly different, since the orthogonality of vanishing points is explicitly used and the distortion coefficients are not estimated at the initialisation phase.

The calibration toolbox will also be used to recover camera extrinsic parameters, from the reconstructed target positions, in the subsequent relative pose calibration.

4.2.2 Inertial Sensor Calibration

Inertial navigation systems also have established calibration techniques, but rely on high-end sensors and actuators. When considering a complete inertial navigation system, initial calibration and alignment are more elaborate [Nebot1997]. Nevertheless, in order to use off-the-shelf inertial sensors attached to a camera, appropriate modelling and calibration techniques are required.

Inertial sensors measure linear acceleration and angular velocity. An inertial measurement unit (IMU) has three orthogonal accelerometers and three orthogonal rate gyros.

To estimate velocity and position integration over time has to be performed, leading to unbounded error. The gyros keep track of rotations, so that linear velocity and position are computed in the correct frame of reference. Appropriate calibration has to be performed to minimise the error buildup.

When using inertial sensors, scale factor, bias and axis-alignment need to be known. For low cost inertial sensors these parameters are not always provided by the manufacturer, and when using discrete components their alignment has to be measured.

To use inertial sensors measurements, assuming a linear model, scale factor, bias and axis-alignment need to be known. For low cost inertial sensors these parameters are not always provided by the manufacturer, and when using discrete components their alignment has to be estimated.

Some of the inertial sensors parameters can be determined by performing simple operations and measuring the sensor outputs.

In [Vieville1989], where the use of an inertial system in a robotic system is analysed, a set of calibration procedures is presented for accelerometers and gyros. In this seminal work, that sets as a future objective the study of the cooperation between vision and inertial sensing, the data provided by the inertial sensors in studied within the context of mobile robotic applications.

Using gravity as a reference, horizontal aligned accelerometers should have zero output, and vertical ones a full $1g$. Placing the IMU in particular directions with respect to gravity, sufficient data can be collected to calibrate, without any special hardware. This static calibration only requires the ability to orient the accelerometers in particular directions

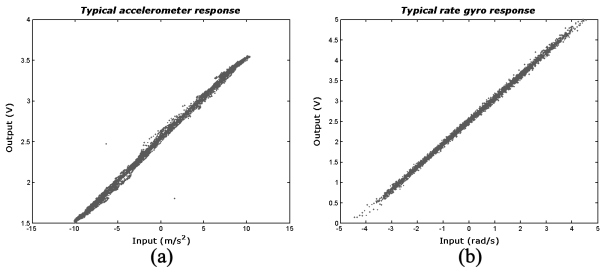


Figure 4.2: Sensors' response: (a) accelerometer, (b) rate gyro

with respect to gravity, and to maintain the system without any movement during the static measurements.

For gyros no such reference is available; there is the earth rotation induced Coriolis force, but it is too small for the kind of rate gyros considered. However the sensor bias or offset can be determined by measuring the output of a static gyro sensor.

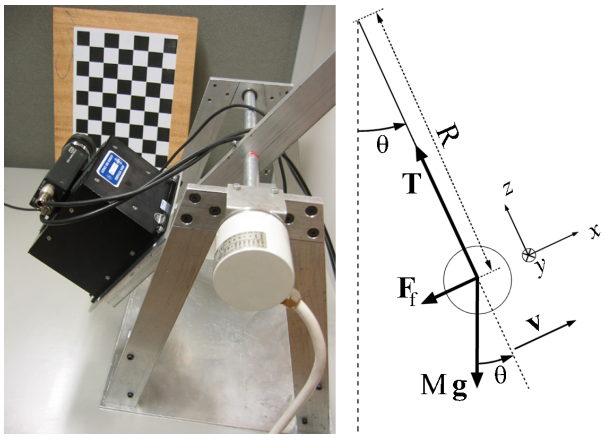
To estimate all the parameters, a dynamic calibration is required. However if a controlled turn rate device is not available, performing a rotation in the vertical plane enables the use gravity as a reference. Using a mechanical axis of rotation, that can be oriented in any direction, the vertical reference will provide the calibration. See [Vieville1989] for the mathematical derivation of this calibration procedure for inertial sensors.

The above setup also solves the problem of determining the alignment between accelerometers and gyros, by relating gyros sensing axis with accelerometer alignment. Having a fixed horizontal rotating axis, continuous rotation provides the gyros sensing axis, and stops along the way provide the relative pose of this sensing axis with the accelerometers.

Calibration with a Pendulum

Some of the inertial sensors parameters can be determined by performing simple operations as described above and measuring the sensor outputs, others can not be so easily determined.

Observing the sensors response which is illustrated in figure 4.2, for a particular accelerometer and a particular rate gyro, it can be seen that this response is practically linear, and so a linear model can be used for the inertial sensors. This model is satisfac-



(a) (b)

Figure 4.3: (a) Pendulum used to calibrate the inertial sensors, (b) Forces acting on a moving pendulum

tory for use with autonomous mobile robots.

Equation (4.1), represents a simple model for each set of three non-coplanar accelerometers or rate gyros, which accounts for the three main errors in these sensors: bias, scale factors and cross-axis sensitivity.

$$\begin{aligned} \boldsymbol{z}_o &= \boldsymbol{M} \cdot \boldsymbol{z}_i + \boldsymbol{b} \\ &= \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} \cdot \begin{bmatrix} z_{ix} \\ z_{iy} \\ z_{iz} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \end{aligned} \tag{4.1}$$

The quantities to be measured are represented by the vector $\boldsymbol{z}_i$, while $\boldsymbol{z}_o$ represents the actual output from the sensors. Vector $\boldsymbol{b}$ represents the bias for each individual sensor, while s_{kk} is the sensitivity (or scale factor) for the sensor oriented along axis k, and s_{kl} the cross sensitivity, resulting from axis misalignments, relating axis k and l.

In this work, a pendulum is used in order to determine the inertial sensors' parameters - see figure 4.3.

The pendulum was chosen since it is relatively straightforward to determine the real quantities the sensors are measuring. To get an indication of the quantities the inertial

sensors should be measuring, it is instrumented with a high-resolution absolute encoder attached to its axis, so that the angular position of the pendulum is known and consequently, the pose of the inertial measuring unit.

In figure 4.3(b) the forces acting on the moving pendulum are represented. A friction force, $\boldsymbol{F}_f$, is represented with its direction opposite to the direction of the pendulum's instantaneous velocity, accounting for all kinds of friction inherent to the pendulum's motion.

The sum of all forces acting on the pendulum induces an acceleration which characterises the pendulum's motion equation. From this motion equation, the acceleration components along the x and z axis, as illustrated in figure 4.3, can be written as

$$a_x = -\|\boldsymbol{g}\|\sin\theta - \frac{\|\boldsymbol{F}_f\|}{M} sgn(\boldsymbol{v}) \tag{4.2}$$

$$a_z = \frac{\|\boldsymbol{T}\|}{M} - \|\boldsymbol{g}\|\cos\theta = \frac{v^2}{R} \tag{4.3}$$

In these equations, $sgn()$ is the sign function, given by

$$sgn(v) = \begin{cases} +1, & v \geq 0 \\ -1, & v < 0 \end{cases} \tag{4.4}$$

The accelerometers measure the acceleration sensed by a proof mass internal to the measuring unit which in turn is attached to the pendulum. This means that the measured accelerations are caused by forces acting on the measuring unit's case, but not on the proof mass. In this particular scenario, since the gravity force acts both on the proof mass and on the case, the accelerometers only measure the accelerations caused by the other forces: the tension, $\boldsymbol{T}$, and the friction force, $\boldsymbol{F}_f$. The measured accelerations along the x and z axis, $\tilde{a}_x$ and $\tilde{a}_z$, are given by

$$\begin{aligned} \tilde{a}_x &= -\frac{\|\boldsymbol{F}_f\|}{M} sgn(\boldsymbol{v}) = a_x + \|\boldsymbol{g}\|\sin\theta \\ &= \alpha R + \|\boldsymbol{g}\|\sin\theta \end{aligned} \tag{4.5}$$

$$\begin{aligned} \tilde{a}_z &= \frac{\|\boldsymbol{T}\|}{M} = \frac{v^2}{R} + \|\boldsymbol{g}\|\cos\theta \\ &= \omega^2 R + \|\boldsymbol{g}\|\cos\theta \end{aligned} \tag{4.6}$$

where ω and α represent the angular velocity and angular acceleration of the pendulum.

The values for θ, ω and α are measured by the encoder readings, and its derivatives. The measurements of the rate gyros, are the components of the angular velocity of the pendulum, meaning that the only rate gyro with a non-zero measurement should be the one oriented perpendicularly to the plane of motion. Using figure 4.3 as a reference, only the rate gyro along the y axis should measure a non-zero quantity, i.e.

$$\tilde{\boldsymbol{\omega}} = \begin{bmatrix} 0 \\ \tilde{\omega}_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{d\theta}{dt} \\ 0 \end{bmatrix} \tag{4.7}$$

By attaching the measuring unit to the pendulum in three different orthogonal orientations, sufficient data can be collected to calibrate the three accelerometers and the three rate gyros of the inertial measuring unit. The procedure consists in determining the nine scale factors, s_{kl}, and the three bias, b_k, of the sensor model described in (4.1). Rewriting the system of equations (4.1) as a function of the unknowns s_{kl} and b_k. The resulting system of equations is given by

$$\begin{aligned} \boldsymbol{z}_o &= \boldsymbol{A} \cdot \boldsymbol{M}' \\ &= \begin{bmatrix} z_{i,x} & 0 & 0 \\ z_{i,y} & 0 & 0 \\ z_{i,z} & 0 & 0 \\ 0 & z_{i,x} & 0 \\ 0 & z_{i,y} & 0 \\ 0 & z_{i,z} & 0 \\ 0 & 0 & z_{i,x} \\ 0 & 0 & z_{i,y} \\ 0 & 0 & z_{i,z} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} s_{xx} \\ s_{xy} \\ s_{xz} \\ s_{yx} \\ s_{yy} \\ s_{yz} \\ s_{zx} \\ s_{zy} \\ s_{zz} \\ b_x \\ b_y \\ b_z \end{bmatrix} \end{aligned} \tag{4.8}$$

where $\boldsymbol{M'}$ is the vector with the twelve parameters to be determined.

Each measurement provides three equations as can be seen in (4.8). The sensor inputs, $\boldsymbol{z}_i$, are known by feeding the encoder readings, and its derivatives, into equations (4.5), (4.6) and (4.7); the sensor outputs, $\boldsymbol{z}_o$, are directly measured. Only the twelve parameters in vector $\boldsymbol{M'}$ are unknown. To obtain a solution for $\boldsymbol{M'}$, at least four measurements have to be known, but since the measurements are disturbed by random noise, a much bigger set of measurements should be used.

A least squares solution can be obtained for the parameters, by using equation (4.9), where $\boldsymbol{A}^{\dagger}$ denotes the pseudo-inverse of matrix $\boldsymbol{A}$ obtained through the use of the singular value decomposition.

$$\boldsymbol{M'} = \boldsymbol{A}^{\dagger} \cdot \boldsymbol{z}_o \tag{4.9}$$

It should be noted that two systems of equations have to be solved: one to determine the parameters of the accelerometers, and another to determine the parameters of the rate gyros.

Temperature Dependence

A well known fact is that inertial sensors parameters are temperature dependent. This model does not account for that, and usually there is a non-linear relation between the parameters and the temperature, which can be different for each of the individual sensors. The proposed solution for being able to cope with different working temperatures, is to build a lookup table containing the parameters for several working temperatures, and then determining the appropriate parameters for the current temperature by interpolation of the table's contents.

Tests and Results

The tests were performed using a DMU-FOG inertial unit from Crossbow Technology coupled with a Sony XC-999 CCD video camera, shown in figure 4.3.

The inertial unit was attached to the pendulum in three distinct orientations in order to obtain a significant set of measurements for each sensor and the correspondent pendulum

Table 4.1: Comparison of the obtained inertial sensors' parameters at two different temperatures with the ones supplied by the manufacturer.

	Accelerometers		
	X	Y	Z
Sensitivity (g/V)			
Manuf. Supplied $(29.82°C)$	1.008	1.000	1.017
Obtained $(29.68°C)$	1.015	1.026	1.022
Obtained $(24.45°C)$	0.999	1.027	1.030
Null Offset (V)			
Manuf. Supplied $(29.82°C)$	2.485	2.519	2.455
Obtained $(29.68°C)$	2.539	2.514	2.456
Obtained $(24.45°C)$	2.526	2.510	2.446

	Rate Gyros		
	X	Y	Z
Sensitivity $(deg.s^{-1}/V)$			
Manuf. Supplied $(29.82°C)$	102.731	101.643	102.388
Obtained $(29.68°C)$	102.202	102.085	102.216
Obtained $(24.45°C)$	102.115	102.155	102.054
Null Offset (V)			
Manuf. Supplied $(29.82°C)$	2.499	2.499	2.499
Obtained $(29.68°C)$	2.500	2.500	2.499
Obtained $(24.45°C)$	2.502	2.500	2.500

position. With (4.5), (4.6), (4.7) and the pendulum angular position, given by the absolute encoder measurements, the expected sensor outputs can be estimated.

Since the inertial measurement unit used in this work is a medium-grade unit, the manufacturer supplies an individual calibration table which can be used as a ground truth to evaluate our calibration procedure.

Table 4.1 presents the parameters supplied by the manufacturer and compares them to the ones obtained using the calibration method described in this paper. It should be noted that in the table, the sensitivity is compared in (g/V) and $(deg.s^{-1}/V)$, which are the inverses of the scale factors, s_{kk}, as defined in equation (4.1).

In order to evaluate the temperature dependence of the sensors parameters, table 4.1 presents the obtained parameters for two different temperatures. The internal tempera-

ture of the inertial unit stabilises after some time (from five to ten minutes) and only after that time were the calibration tests performed, in order that all the data be obtained at the same constant temperature. Since the manufacturer only presents the calibration parameters for an internal temperature of 29.82 $^{\circ}C$, these should only be compared with the ones obtained at a similar internal temperature (29.68 $^{\circ}C$), which was the stabilised internal temperature of the unit when the room temperature was around 22 $^{\circ}C$. Regarding parameters variations with temperature, one can easily observe that these variations differ for each individual sensor; considering also that the stabilised internal temperature of the unit varies slightly for normal operation conditions, a lookup table for the parameters can be a simple and straightforward solution to compensate for temperature variations.

The manufacturer does not present any parameters relating to axis alignment in their unit. However, from the results of our method the system exhibits a mean cross-axis sensitivity of about 0.6%. These small cross-axis errors can cause high drifts over time if the inertial data measurements are to be used to estimate position, by integrating in time the sensors' data.

To demonstrate the effect of this cross-axis sensitivity and the differences between using our calibration or the manufacturer's calibration data, a test was performed where the pendulum swang for some time with the unit's internal temperature close to that specified in the manufacturer's calibration sheet. During the experiment, the pendulum's motion was sometimes forced, and other times the pendulum was left oscillating freely. The sensors' data was recorded and afterwards the rate gyros outputs were integrated over time in order to obtain the pendulum's angle.

Figure 4.4 presents the results obtained by the simple integration of the inertial data after being calibrated with the parameters obtained by us for the testing temperature. The inability to distinguish between the two curves shows that the real angle, obtained by the encoder readings, only has slight differences from the one obtained by the integration of the rate gyro output.

In figure 4.5, the results for the same experiment are presented, but this time using the manufacturer's supplied calibration sheet to calibrate the inertial data. As can be easily seen, the estimation has a significant drift, much higher than the drift achieved when using the calibration data obtained by the procedure described above.

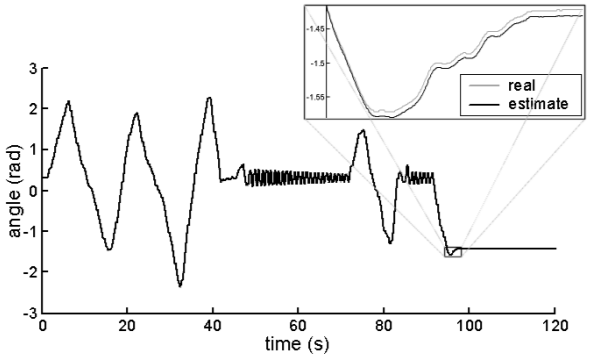


Figure 4.4: Results for the integration of inertial data calibrated by our method.

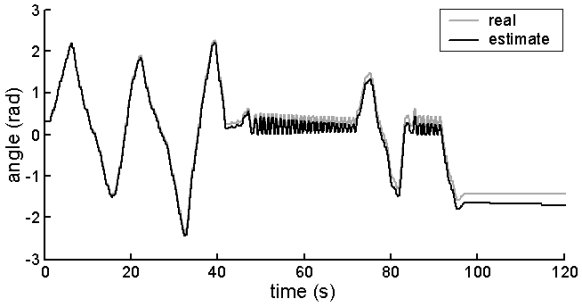


Figure 4.5: Results for the integration of inertial data calibrated by manufacturer.

These results proved satisfactory, and good enough for many mobile robotics applications. The calibration procedure presented has been able to reduce drastically the drift obtained by the integration of inertial data, by determining the sensors' parameters with a reasonably high accuracy. More details and results of this calibration method are presented in [Alves2003ICAR] and [Alves2003].

4.3 Relative Pose Calibration between Visual and Inertial Sensors

In the following sections we will present our method for calibration of rotation and translation between the camera and the inertial sensors. Using the gravity reference a static boresight approach is proposed, requiring a simple setup and avoiding fast blurred images

and controlled active rate generators. This only relates the accelerometers' with the cameras, the relative pose between the accelerometers and gyros can be done independently, using horizontal and vertical rotation axis as described above.

4.3.1 Calibration of Rotation between IMU and Camera

In order to determine the rigid rotation between the INS frame of reference $\{\mathcal{I}\}$ and the camera frame of reference $\{\mathcal{C}\}$, both sensors are used to measure the vertical direction, as shown in fig. 4.6. When the IMU sensed acceleration is equal in magnitude to gravity, the sensed direction is the vertical. For the camera, using a specific calibration target such as a chessboard target placed vertically, the vertical direction can be taken from the corresponding vanishing point.

This boresight static approach can be easily performed, not requiring any additional equipment, apart from the chessboard target, obtained using a standard printer, already used for camera calibration.

If n observations are made for distinct camera positions, recording the vertical reference provided by the inertial sensors and the vanishing point of scene vertical features, the absolute orientation can be determined using the orthogonal Procrustes method for 3D attitude estimation. We will use Horn's closed-form solution for absolute orientation using unit quaternions [Horn1987], applied here only to unit vectors. Since we are only

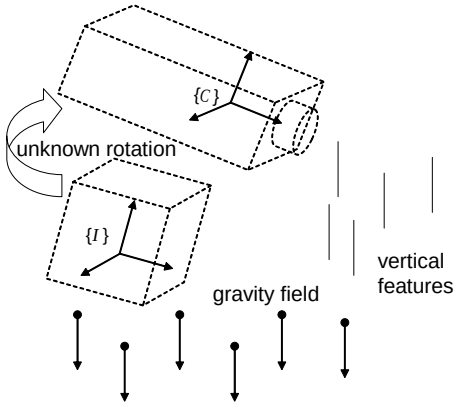


Figure 4.6: IMU and camera observing gravity.

observing a 3D direction in space, we can only determine the rotation between the two frames of reference.

Let ${}^{\mathcal{I}}\mathbf{v}_i$ be a measurement of the vertical by the inertial sensors, and ${}^{\mathcal{C}}\mathbf{v}_i$ the corresponding measurement made by the camera derived from some scene vanishing point. We want to determine the unit quaternion $\mathring{\mathbf{q}}$ that rotates inertial measurements in the inertial sensor frame of reference $\{\mathcal{I}\}$ to the camera frame of reference $\{\mathcal{C}\}$. We want to find the unit quaternion $\mathring{\mathbf{q}}$ that maximises

$$\sum_{i=1}^{n}(\mathring{\mathbf{q}}\,{}^{\mathcal{I}}\boldsymbol{v}_i\,\mathring{\mathbf{q}}^*)\cdot{}^{\mathcal{C}}\boldsymbol{v}_i \tag{4.10}$$

which can be rewritten as

$$\sum_{i=1}^{n}(\mathring{\mathbf{q}}\,{}^{\mathcal{I}}\boldsymbol{v}_i)\cdot({}^{\mathcal{C}}\boldsymbol{v}_i\,\mathring{\mathbf{q}}) \tag{4.11}$$

The quaternion product can be expressed as a matrix. Using ${}^{\mathcal{I}}\boldsymbol{v}_i = ({}^{\mathcal{I}}x_i, {}^{\mathcal{I}}y_i, {}^{\mathcal{I}}z_i)^T$ and ${}^{\mathcal{C}}\boldsymbol{v}_i = ({}^{\mathcal{C}}x_i, {}^{\mathcal{C}}y_i, {}^{\mathcal{C}}z_i)^T$ we define

$$\mathring{\mathbf{q}}\,{}^{\mathcal{I}}\boldsymbol{v}_i = \begin{bmatrix} 0 & -{}^{\mathcal{I}}x_i & -{}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}z_i \\ {}^{\mathcal{I}}x_i & 0 & {}^{\mathcal{I}}z_i & -{}^{\mathcal{I}}y_i \\ {}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}z_i & 0 & {}^{\mathcal{I}}x_i \\ {}^{\mathcal{I}}z_i & {}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}x_i & 0 \end{bmatrix} \mathring{\mathbf{q}} = {}^{\mathcal{I}}\,\mathbf{V}_i\mathring{\mathbf{q}} \tag{4.12}$$

and

$${}^{\mathcal{C}}\boldsymbol{v}_i\,\mathring{\mathbf{q}} = \begin{bmatrix} 0 & -{}^{\mathcal{C}}x_i & -{}^{\mathcal{C}}y_i & -{}^{\mathcal{C}}z_i \\ {}^{\mathcal{C}}x_i & 0 & -{}^{\mathcal{C}}z_i & {}^{\mathcal{C}}y_i \\ {}^{\mathcal{C}}y_i & {}^{\mathcal{C}}z_i & 0 & -{}^{\mathcal{C}}x_i \\ {}^{\mathcal{C}}z_i & -{}^{\mathcal{C}}y_i & {}^{\mathcal{C}}x_i & 0 \end{bmatrix} \mathring{\mathbf{q}} = {}^{\mathcal{C}}\,\mathbf{V}_i\mathring{\mathbf{q}} \tag{4.13}$$

Substituting in (4.11)

$$\sum_{i=1}^{n}({}^{\mathcal{I}}\mathbf{V}_i\mathring{\mathbf{q}})\cdot({}^{\mathcal{C}}\mathbf{V}_i\mathring{\mathbf{q}}) \tag{4.14}$$

or

$$\sum_{i=1}^{n}\mathring{\mathbf{q}}^T\,{}^{\mathcal{I}}\mathbf{V}_i^T\,{}^{\mathcal{C}}\mathbf{V}_i\mathring{\mathbf{q}} \tag{4.15}$$

factoring out $\mathring{\mathbf{q}}$ we get

$$\mathring{\mathbf{q}}^T \left(\sum_{i=1}^{n} {}^{\mathcal{I}}\mathbf{V}_i^T \, {}^{\mathcal{C}}\mathbf{V}_i \right) \mathring{\mathbf{q}} \tag{4.16}$$

So we want to find $\mathring{\mathbf{q}}$ such that

$$\max \mathring{\mathbf{q}}^T \, \boldsymbol{N} \, \mathring{\mathbf{q}} \tag{4.17}$$

where

$$\boldsymbol{N} = \sum_{i=1}^{n} {}^{\mathcal{I}}\mathbf{V}_i^T \, {}^{\mathcal{C}}\mathbf{V}_i \; . \tag{4.18}$$

Having

$$S_{xx} = \sum_{i=1}^{n} {}^{\mathcal{I}}x_i \, {}^{\mathcal{C}}x_i \; , \; S_{xy} = \sum_{i=1}^{n} {}^{\mathcal{I}}x_i \, {}^{\mathcal{C}}y_i \tag{4.19}$$

and analogously for all 9 pairings of the components of the two vectors, matrix $\boldsymbol{N}$ can be expressed using these sums as in (4.20). The sums contain all the information that is required to find the solution.

$$\boldsymbol{N} = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix} \tag{4.20}$$

Since $\boldsymbol{N}$ is a symmetric matrix, the solution to this problem is the four-vector $\boldsymbol{q}_{max}$ corresponding to the largest eigenvalue λ_{max} of $\boldsymbol{N}$ - see [Horn1987] for details.

Measurement Span for Rotation Estimation

The above method finds the rotation that maximises the alignment of the rotated inertial frame verticals with the camera observed verticals expressed by (4.10).

The inertial frame verticals, ${}^{\mathcal{I}}\boldsymbol{v}_i$, are easily obtained from the IMU accelerometers. The only restriction is that the system has to be motionless, or subject to constant speed, so that gravity can be used as a vertical reference.

The camera frame verticals, ${}^{\mathcal{C}}\boldsymbol{v}_i$, are not so easily obtained. Some scene element must be known to have vertical features, so that the vertical vanishing point can be determined. In our experimental work we relied on the same chessboard target used for calibrating the camera, but now placing it vertically. For the n observations, the target does not have to remain in the same position, but must be vertical.

A single pair of measurements, i.e., $n = 1$, provides a valid rotation for the given observation, but prone to degenerate cases, depending on the system pose and rotation between frames. Using more observations at distinct system poses avoids this, and improves the estimate by reducing estimation error, assuming that the measurements have zero mean Gaussian noise. The camera poses used need not span the entire 3D attitude space, a few poses with the system at different rotations relative to the inertial vertical are sufficient to avoid ill conditioned cases.

Weighing Observation Error in Rotation Calibration

When the expected errors in the observed verticals are known, it is convenient to introduce weights in the above calculations. Introducing weights in to (4.10) we have

$$\sum_{i=1}^{n} w_i \, (\mathring{\mathbf{q}} \, {}^{\mathcal{I}}\boldsymbol{v}_i \, \mathring{\mathbf{q}}^*) \cdot {}^{\mathcal{C}}\boldsymbol{v}_i \tag{4.21}$$

where w_i represents the confidence in measurements ${}^{\mathcal{I}}\mathbf{v}_i$ and ${}^{\mathcal{C}}\mathbf{v}_i$. This will mean that the products in the sums that compose $\boldsymbol{N}$ are weighed, i.e.,

$$\boldsymbol{N} = \sum_{i=1}^{n} w_i \, {}^{\mathcal{I}}\mathbf{V}_i^T \, {}^{\mathcal{C}}\mathbf{V}_i \; . \tag{4.22}$$

So we will have

$$S_{xx} = \sum_{i=1}^{n} w_i \, {}^{\mathcal{I}}x_i \, {}^{\mathcal{C}}x_i \; , \; S_{xy} = \sum_{i=1}^{n} w_i \, {}^{\mathcal{I}}x_i \, {}^{\mathcal{C}}y_i \tag{4.23}$$

and analogously for all 9 pairings of the components of the two vectors, and the rotation will be given by the four-vector $\boldsymbol{q}_{max}$ corresponding to the largest eigenvalue λ_{max} of $\boldsymbol{N}$, as before.

But w_i represents the confidence in both measurements ${}^{\mathcal{I}}\mathbf{v}_i$ and ${}^{\mathcal{C}}\mathbf{v}_i$. Since each camera sensor observation can only contribute to the rotation estimation if the corresponding inertial sensor observation is valid, given the individual weighs ${}^{\mathcal{I}}w_i$ and ${}^{\mathcal{C}}w_i$ and (4.22), we can define w_i as $w_i =^{\mathcal{I}} w_i \, {}^{\mathcal{C}}w_i$.

To set the weights ${}^{\mathcal{I}}w_i$ and ${}^{\mathcal{C}}w_i$ we have to take into account how the measurements ${}^{\mathcal{I}}\mathbf{v}_i$ and ${}^{\mathcal{C}}\mathbf{v}_i$ are obtained.

The camera observed verticals are obtained from the vanishing point of some image vertical feature, in our case the chessboard target used in the calibration. The camera calibration toolbox used [Bouguet2006] provides an error measure on the recovered extrinsic parameters, that result from the minimisation of the reprojection error (through gradient descent). Associated with the camera extrinsic parameters $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$, as defined in section 2.3.1, we have the uncertainties stored in $\boldsymbol{R}_{e_i}$ and $\boldsymbol{t}_{e_i}$ that represent approximately three times the standard deviations of the errors of estimation. The error in the camera observation of the vertical can be represented by considering the magnitude of the rotation uncertainty angle θ_e.

The inertial sensor observed verticals results from the measurements of the three orthogonal accelerometers. Taking a set of measurements and averaging the result reduces the error from sensor noise or mechanical noise (vibrations or motion). The spread in the measurements θ_e, i.e., three times the root mean square angular deviation of the measurements from the mean, provides a level of confidence in the average value taken to get ${}^{\mathcal{I}}\mathbf{v}_i$.

For each sensor we can consider a maximum threshold θ_{max} on the rotation angle error θ_e and define w_i as

$$w_i = \begin{cases} 1 - \frac{\theta_e}{\theta_{max}} & , \theta_e < \theta_{max} \\ 0 & , \theta_e \geq \theta_{max} \end{cases} \tag{4.24}$$

The selection of an appropriate threshold θ_{max} has to take into account the observed input errors and number of frames available for calibration.

Rotation Calibration Summary

- N static observations at distinct positions:
  - vertical chessboard target;
  - save image and corresponding inertial data.
- Perform standard camera calibration for image set.
- Compute rotation quaternion:
  - use target vertical vanishing points ${}^{\mathcal{C}}\boldsymbol{v}_i$ detected for camera calibration;
  - inertial data from accelerometers provide ${}^{\mathcal{I}}\boldsymbol{v}_i$;
  - rotation quaternion $\mathring{\mathbf{q}}$ given by (4.17).

Figure 4.7: Summary of required steps to perform calibration of rotation between camera and IMU using the proposed algorithm.

Rotation Calibration Summary

Figure 4.7 provides a summary of required steps to perform calibration of rotation between camera and IMU using the proposed algorithm.

Error Sensitivity and Simulation Results

In order to validate the proposed method and perform noise sensitivity tests, simulations where performed under varying conditions.

For each simulation run a random rotation $\mathring{\mathbf{q}}$ is applied to a random set of simulated inertial observed verticals, ${}^{\mathcal{I}}\mathbf{v}_i$, to obtain a corresponding set of camera observed verticals, ${}^{\mathcal{C}}\mathbf{v}_i$. These simulated camera observations are corrupted by applying a random rotation with a normal distributed magnitude (with zero mean and set standard deviation) about a random axis, i.e. a uniformly distributed 3D axis. The rotation quaternion that relates the two sets is estimated as $\hat{\mathring{\mathbf{q}}}$ by the above method. The error in the estimation can be measured by considering the rotation required to correct the estimate to the true value, $\mathring{\mathbf{q}} = \mathring{\mathbf{e}} * \hat{\mathring{\mathbf{q}}}$. With $\theta_e = 2\cos^{-1}(e_s)$, where $e_s$is the scalar component of $\mathring{\mathbf{e}}$, we take $\delta_\theta = |\theta_e|$

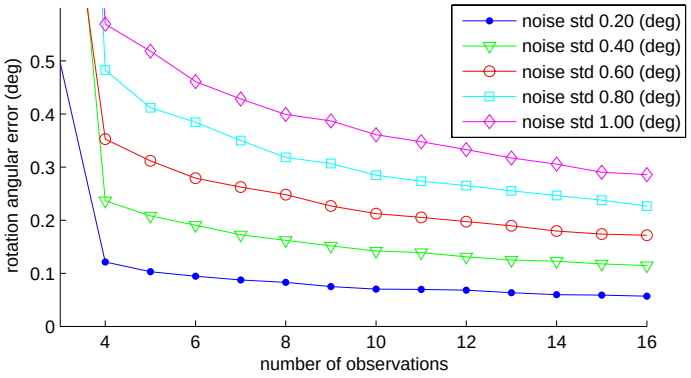


Figure 4.8: Simulation rotation estimation mean error for increasing number of observations.

as the error measure.

Figure 4.8 shows simulation results of several takes with different number of observations used. The increasing error lines correspond to increasing rotation error added to the simulated observed camera verticals. For each setting the method runs 1 000 times and the mean error is evaluated.

The above simulation was performed with a random set of simulated camera observed verticals, uniformly distributed on the unit sphere. To better evaluate the method, simulations were performed with restricted sets of simulated observations.

Figure 4.9 shows simulation results with simulated camera observed verticals, restricted to a 20 *deg* patch of the unit sphere. The geometric dilution of precision from such a narrow field of observation leads to the poorer results, but since the added noise has a normal distribution with a maximum standard deviation of 1 *deg* a good estimate of the rotation is still obtained.

To approach a degenerate case of having all observation in the same plane, another simulation was performed with camera observed verticals restricted to a patch of the unit sphere corresponding to a full great circle ring with 1 *deg* thickness. Figure 4.10 shows the results. Since the noise level ($\pm std$) is above the out of plane distribution of the observation, the degenerate single plane observations dominate and lead to the high error in the estimated rotation.

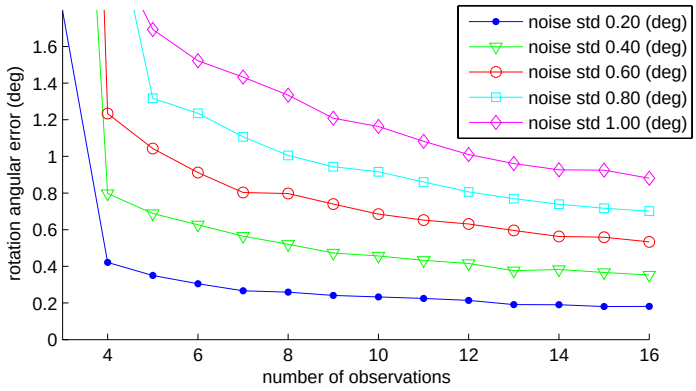


Figure 4.9: Simulation rotation estimation mean error for increasing number of observations, with simulated camera observed verticals, restricted to a 20 *deg* patch of the unit sphere.

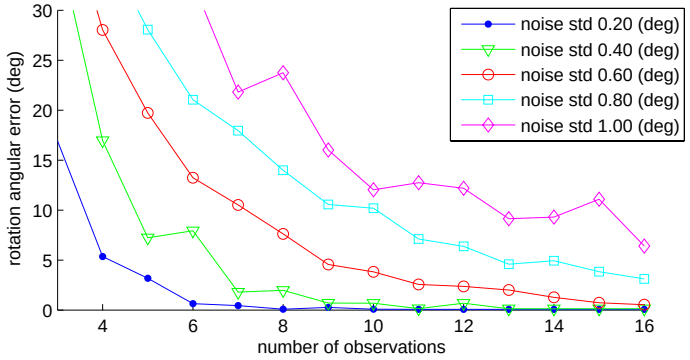


Figure 4.10: Simulation rotation estimation mean error for increasing number of observations, with simulated camera observed verticals, restricted to a patch of the unit sphere corresponding to a full great circle ring with 1 *deg* thickness.

Figure 4.11: Required setup for *out of the box* camera and inertial to camera rotation calibration.

Real Data Results

The rotation estimation can be performed together with the camera calibration with the simple setup shown in fig. 4.11. The code used is available from the implemented InerVis Matlab Toolbox [Lobo2006], that adds on to the Camera Calibration Toolbox [Bouguet2006].

After rigidly fixing an inertial sensor to a camera rig, the calibration was performed with the proposed method.

The camera was calibrated with images of a vertical chessboard target from several camera positions. Figure 4.1 show some of the images used in this calibration and corresponding extrinsic parameters.

Since the target is placed vertically, the camera verticals from the target vertical vanishing points ${}^{\mathcal{C}}\boldsymbol{v}_i$ are given by the reconstructed target positions.

A total set of 16 images and accelerometer data was taken, and the estimated rotation was $\mathring{\mathbf{q}} = -0.7149 < 0.010013, 0.023479, 0.69876 >$, indicating a -88.73 *deg* rotation about the axis $(0.0143, 0.0336, 0.9993)$, i.e. a near right angle about the camera z-axis consistent with the layout shown in fig. 4.12.

Using the estimated rotation, the inertial sensed verticals where rotated to match with the vertical vanishing point of the chessboard target, and the observed misalignment had a root mean square error of 0.69 *deg*, as shown in fig. 4.13.

The results show that the method performs well, and is easy to implement. From our experimental tests, of which the above is just an example, the key factors are the quality

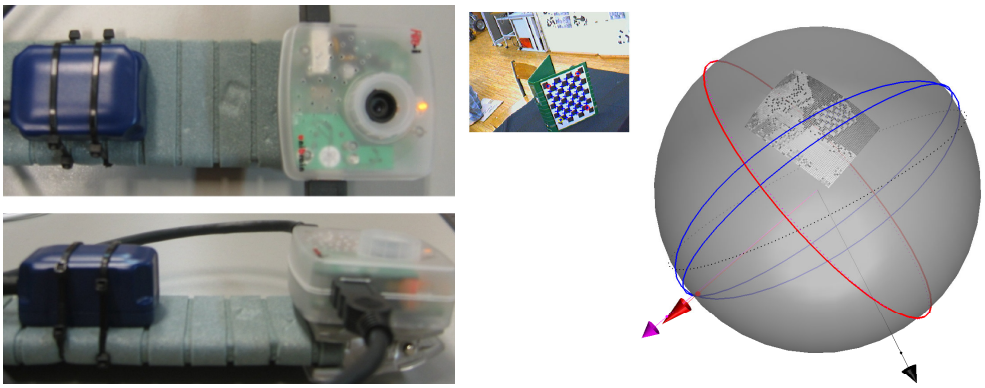


Figure 4.12: Sensor layout and unit sphere projection with vanishing point and reprojected verticals from rotation calibration.

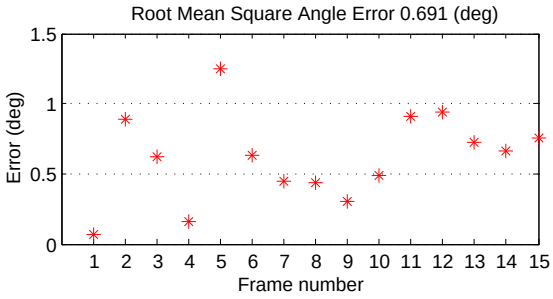


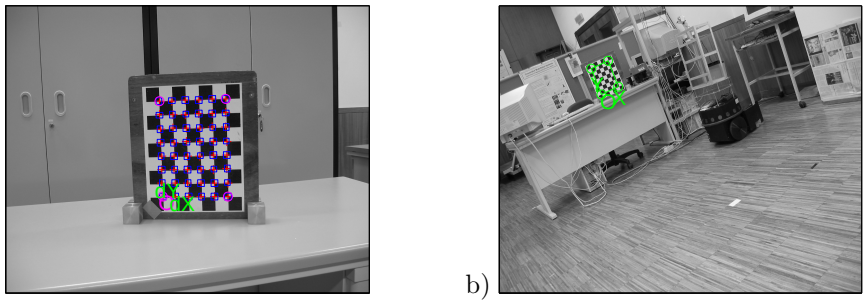
Figure 4.13: Reprojection alignment errors for verticals in each frame used in rotation estimation

of the vanishing points obtained from the camera target images, that also determine the quality of the camera calibration.

Real Data Results with Input Error Weighing

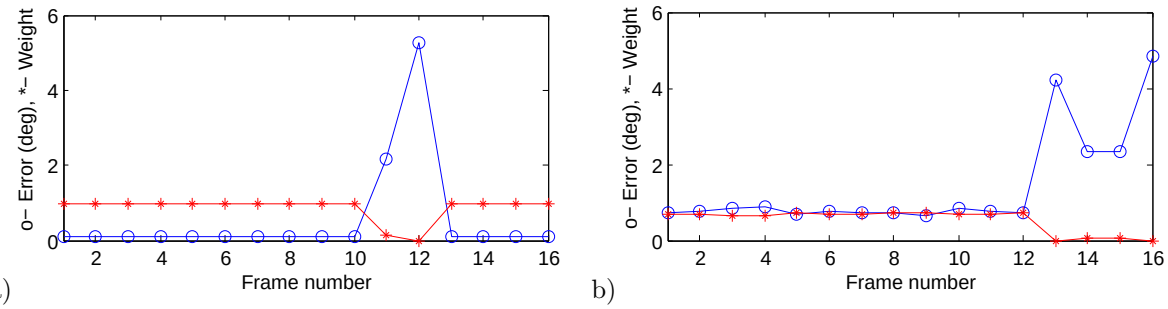
To test the weighing of the observation error in the rotation calibration, a data set was collected for which known perturbations were introduced. A total of 16 frames were taken, the last 4 with the vertical chessboard target significantly more distant to degrade the vertical varnishing point accuracy, and two other frames taken with the system not perfectly static to degrade the inertial vertical reference accuracy.

At each observation a batch of measurements is taken from the inertial sensors, and the mean vertical reference provided by the accelerometers is used. The inertial sensed



a) b)

Figure 4.14: Two frames from the set of 16 used, with the chessboard target near (a) and more distant (b).



a) b)

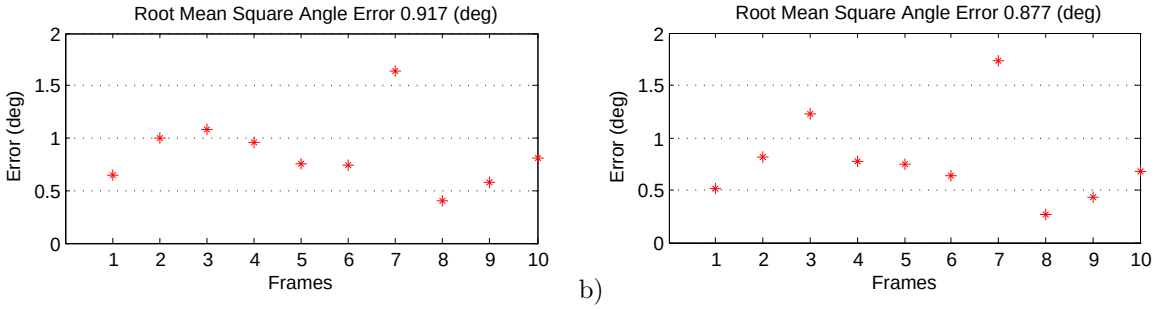
Figure 4.15: Inertial sensed vertical (a) and observed camera vertical (b) error spread and corresponding error weights for $\theta_{max} = 2.5\ deg$.

vertical error weights are derived from the spread in the measurements. Selecting for this set a $\theta_{max} = 2\ deg$ we obtained the weights shown in figure 4.15.

The chessboard target vertical vanishing points obtained from the camera calibration toolbox also have an associated error measure, from which the weights are derived. Figure 4.15 shows the error measure and corresponding weight used.

Using the complete set of 16 frames and weighing the input errors, the estimated rotation was $\mathring{\mathbf{q}}_w = -0.7149 < 0.010013, 0.023479, 0.69876 >$, without using weights the estimated rotation was $\mathring{\mathbf{q}} = -0.51153 < -0.49442, -0.49743, 0.49644 >$. The rotation without weighing errors deviates by $0.2\ deg$ from the rotation obtained.

No ground truth is available for comparison, but we can observe the reprojection alignment errors for verticals in the first 10 frames, for which the input noise is small. The results show a slight improvement, as seen in figure 4.16 where the observed misalignment root mean square error improved from $0.917\ deg$ to $0.877\ deg$.



a) b)

Figure 4.16: Reprojection alignment errors (a) without input error weighing (b) with input error weighing.

Depending on the setup and restrictions when performing the calibration weighing errors can be a good approach, but it is best to calibrate with the system perfectly static, avoiding noise in the inertial sensed vertical, and selecting views with the vertical target near and in full view to obtain good vanishing points.

4.3.2 Calibration of Translation between IMU and Camera

From (3.26) we can see that only dynamic motion will have non zero acceleration from which translation $\mathbf{r}$ can be inferred.

A static boresight approach like the one used for rotation is easier to perform. If the IMU can be set to rotate about its sensing point and axis, than the camera motion will have the same rotation and a translation depending on the lever arm $\mathbf{r}$ joining the two.

With a turntable and suitable positioning rig the IMU can be set to rotate about a null point. This requires a mechanical rig, but not a controlled dynamic motion requiring expensive equipment. The output has to be monitored and adjustments made, starting from the expected sensing axis.

After adjusting the IMU, if $2n$ observations are made for distinct camera positions, with the chessboard target fixed and placed in camera view for each pair of measurements, lever arm $\mathbf{r}$ can be estimated.

Standard hand-eye calibration [Daniilidis1999] can than formulated using homogeneous transformation matrices as solving

$$\boldsymbol{AX} = \boldsymbol{XB} \tag{4.25}$$

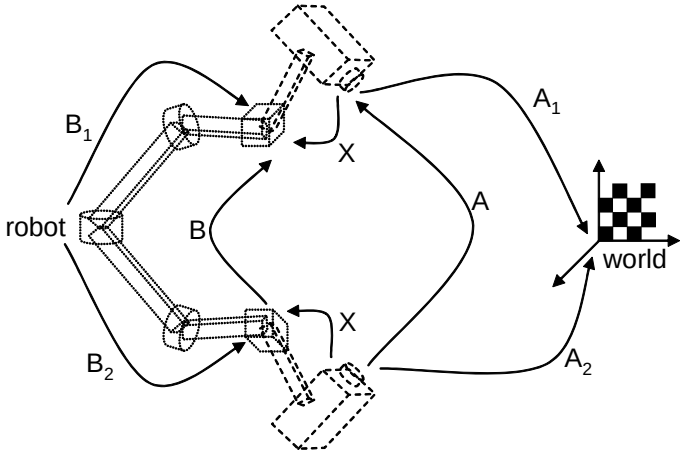


Figure 4.17: Transformations between frames in robot with camera, where $\boldsymbol{X}$ is the unknown hand-to-eye transformation.

for an unknown hand-to-eye transformation $\boldsymbol{X}$, where A is the camera (eye) relative motion transformation, and B the gripper (hand) relative motion transformation, as shown in figure 4.17.

This equation is a particular case of the Sylvester equation $\boldsymbol{AX} - \boldsymbol{XB} = \boldsymbol{C}$. Decomposing the homogeneous transformations in (4.25) into rotation and translation components (R, t) we get one matrix and one vector equation

$$\boldsymbol{R}_A \boldsymbol{R}_X = \boldsymbol{R}_X \boldsymbol{R}_B, \tag{4.26}$$

$$(\boldsymbol{R}_A - \boldsymbol{I})\, \mathbf{t}_X = \boldsymbol{R}_X \mathbf{t}_B - \mathbf{t}_A. \tag{4.27}$$

The majority of the approaches solve first for rotation (4.26) and than for translation (4.27). At least two motions with rotations about non parallel axis are required.

When performing the hand-eye calibration for a robotic manipulator the relative camera transformation $\boldsymbol{A}$ can be obtained using a fixed world target and computing the camera-to-world transformation before and after the motion, $\boldsymbol{A}_1$ $\boldsymbol{A}_2$, and making $\boldsymbol{A} = \boldsymbol{A}_2 \boldsymbol{A}_1^{-1}$. Similarly, having the transformation matrices from the fixed robot base to the gripper, $\boldsymbol{B}_1$ $\boldsymbol{B}_2$, we have $\boldsymbol{B} = \boldsymbol{B}_2^{-1} \boldsymbol{B}_1$. Keeping the robot base and target fixed, as shown in figure 4.17, a set n poses can generate $(\frac{n!}{2!(n-2)!})$ relative motions for which the above equations can be solved.

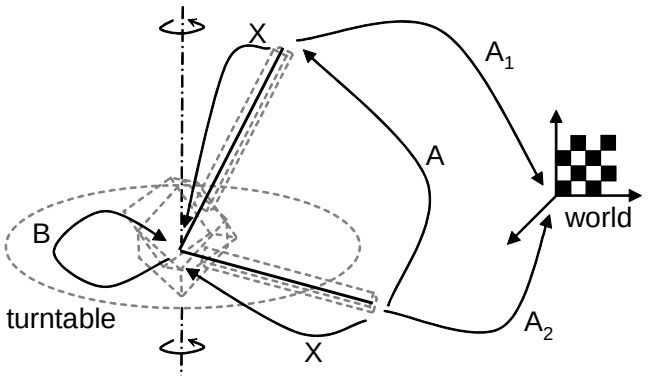


Figure 4.18: Turntable used for unknown lever arm calibration as a hand-to-eye transformation for one turn. Complete calibration requires n turns, with $2n$ static poses with rotation about IMU null point.

For our particular case we want to estimate the lever arm $\mathbf{r}$ in the camera frame of reference, and perform simple turns about the lever arm end point, adjusted to coincide with the inertial sensor center. Our *hand* does not translate, and only rotates in exactly the same way as the camera, i.e. $\mathbf{t}_B = \mathbf{0}$, $\boldsymbol{R}_A = \boldsymbol{R}_B$ and $\boldsymbol{R}_X = \boldsymbol{I}$, and the transformation considered in figure 4.17 are simplified as shown in figure 4.18.

Rewriting (4.27) for this case we have

$$(\boldsymbol{R}_A - \boldsymbol{I})\,\mathbf{r} = -\mathbf{t}_A. \tag{4.28}$$

where the relative motion parameters can be obtained from the camera-to-target visual calibration. However, since the target is being repositioned after each turn, $2n$ poses only contribute n relative motions for the estimation of $\mathbf{r}$. Each pair contributes with the projection of $\mathbf{r}$ on the rotation plane, and at least two rotations about non parallel axis are required. The above equation can be rewritten for the n relative motions $\triangle_i$ as

$$(\boldsymbol{R}_{\triangle_i} - \boldsymbol{I})\,\mathbf{r} = -\mathbf{t}_{\triangle_i}. \tag{4.29}$$

The camera translation $\mathbf{t}_{\triangle_i}$ induced by the lever arm $\mathbf{r}$ can be estimated by observing a fixed chessboard target with the camera and recovering the extrinsic parameters. The final camera position relative to its initial position gives translation $\mathbf{t}_{\triangle_i}$ and rotation $\boldsymbol{R}_{\triangle_i}$.

Solving (4.29) for n turns using the standard hand-eye method [Tsai1989] we obtain the 3D lever arm $\mathbf{r}$ in the camera frame of reference.

However, since the target is being repositioned after each turn, $2n$ poses only contribute n relative motions for the estimation of $\mathbf{r}$.

The n turns are performed as depicted in figure 4.18. For each turn the system is repositioned with a distinct pose on the turntable and adjusted to rotate about a null point. The chessboard target is also repositioned and placed in camera view for each pair of measurements, so that it is seen at the start and end of each turn.

Translation Calibration Summary

Figure 4.19 provides a summary of required steps to perform calibration of translation between camera and IMU using the proposed algorithm.

Error Sensitivity and Simulation Results

In order to validate the proposed method and perform noise sensitivity tests, simulations where performed under varying conditions.

The above described method takes a set of measured camera translations $\mathbf{t}_{\Delta_i}$ and rotations θ_{Δ_i}, induced by the unknown lever arm $\mathbf{r}$.

For each simulation run a random lever arm $\mathbf{r}$ is chosen and set of random rotations $\boldsymbol{R}_{\Delta_i}$ are applied to produce a set of simulated camera translations $\mathbf{t}_{\Delta_i}$.

With $\nu = \mathrm{SNR}^{-1} \in (0, 1)$ being the inverse of the signal to noise ratio, we disturb the simulated translation values $\mathbf{t}_{\Delta_i}$, by

$$\widetilde{\mathbf{t}_{\Delta_i}} = \mathbf{t}_{\Delta_i} + \nu \left\| \mathbf{t}_{\Delta_i} \right\| randn_{3\times 1} \tag{4.30}$$

where $randn_{n\times 1}$ is a n vector of random numbers that follow a uniform distribution, simulating white Gaussian noise with zero mean and $\sigma = 1$.

The estimated lever arm $\hat{\mathbf{r}}$ is compared with the true simulation value $\mathbf{r}$, in length and alignment, to get the error measure. Fig. 4.20 shows a set of simulation results of several takes with different noise levels and number of turns used, with 1000 runs in each take.

Translation Calibration Summary

- Perform standard camera calibration and rotation calibration with same image data set.
- 2N static observations with N turns about IMU at distinct poses:
  - position system on turntable;
  - force rapid motion and observe accelerometer output;
  - reposition until accelerometer output is null;
  - place the chessboard target in camera view for the maximum turn amplitude;
  - save image and corresponding inertial data before and after the turn;
  - repeat for N turns with distinct axis about the IMU.
- Compute translation (lever arm):
  - use target vertical vanishing points ${}^{\mathcal{C}}\boldsymbol{v}_i$ detected for camera calibration;
  - inertial data from accelerometers provide ${}^{\mathcal{I}}\boldsymbol{v}_i$;
  - solve (4.29) for N turns using the standard hand-eye method to obtain the 3D lever arm. $\mathbf{r}$

Figure 4.19: Summary of required steps to perform calibration of translation between camera and IMU using the proposed algorithm.

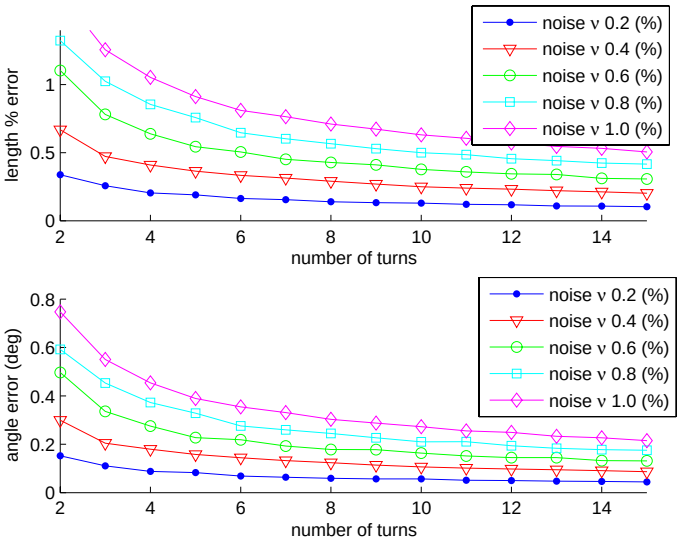


Figure 4.20: Simulation translation estimation mean error for increasing number of turns.

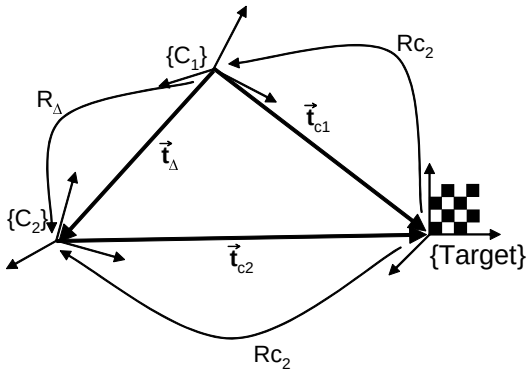


Figure 4.21: Parameters obtained from camera calibration and derived translation induced by lever arm rotation.

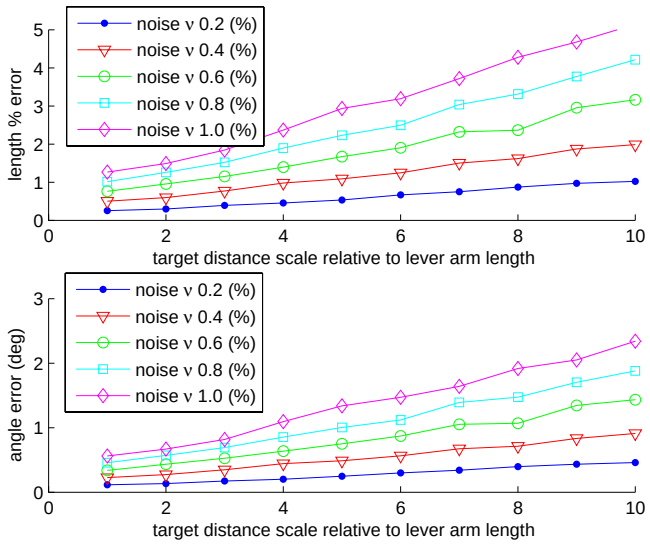


Figure 4.22: Translation estimation from simulated camera extrinsic parameters for increasing target distance scale relative to lever arm length. .

Mean length error is given as a percentage of real value and angular error by its absolute mean value.

To better understand noise sensitivity issues, we have to take into account how the rotation induced translation is measured. By observing the chessboard target and performing the camera calibration with the Matlab Camera Calibration Toolbox [Bouguet2006], we obtain the camera extrinsic parameters for each image relative to the target, as shown in fig. 4.21.

The above described camera translations $\mathbf{t}_{\Delta_i}$ and rotations $\boldsymbol{R}_{\Delta_i}$, induced by the unknown lever arm $\mathbf{r}$, can be derived from the camera extrinsic parameters as follows

$$\boldsymbol{R}_{\Delta_i} = \boldsymbol{R}c_1 \boldsymbol{R}c_2^{-1} \tag{4.31}$$

$$\mathbf{t}_{\Delta_i} = \boldsymbol{R}c_1 \left(\boldsymbol{R}c_2^{-1} \left(-\mathbf{t}c_2 \right) \right) + \mathbf{t}c_1 \tag{4.32}$$

where index 1 and 2 indicate the initial and final extrinsic camera parameters for turn i, both relative to the camera position before the turn.

Since the real data will be derived in this way, a second simulation trial was made, but now adding white Gaussian noise to $\mathbf{t}c_n$ and $\boldsymbol{R}c_n$. The behaviour of the method with

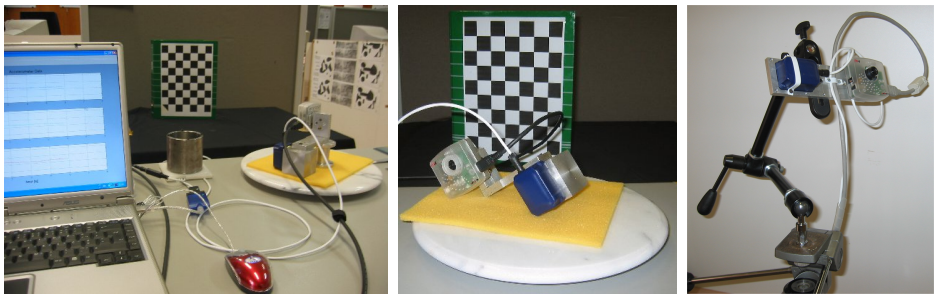


Figure 4.23: Required setup for translation calibration, passive turntable and chessboard calibration target, and alternative rotation setup with magic arm.

added noise and number of turns has already been evaluated. The critical factor when considering the geometry presented in fig. 4.21 is the dilution of precision that results when estimating the translation with (4.32). To study this effect, the simulation runs where performed for different target distances, relative to the lever arm length.

Fig. 4.22 shows simulation results of several takes with different noise levels and target distance to camera scale relative to lever arm length, using 10 turns per run, with 1000 runs in each take. Mean length error is given as a percentage of real value. The results clearly shows the limitations of the method, and that care has to be taken in positioning the target, so that the error is not amplified in the lever arm computation.

Real Data Results

Figure 4.23 shows the setup required to perform the translation calibration. The system is placed on the passive turntable in several distinct poses. Each pose is adjusted to the null point by observing the inertial sensor outputs under forced motion, so that the rotation axis coincides with the inertial sensor center. Two static camera images are than taken, before and after rotation, to obtain the lever arm induced translation used in the above described method to determine the lever arm or translation between the IMU and the camera.

Another alternative to the passive turntable is using a *magic arm* to position the system, and fixing it to some rotating base as shown in figure 4.23. This arrangement allows more flexible orientation of the system, however when positioning on the turntable

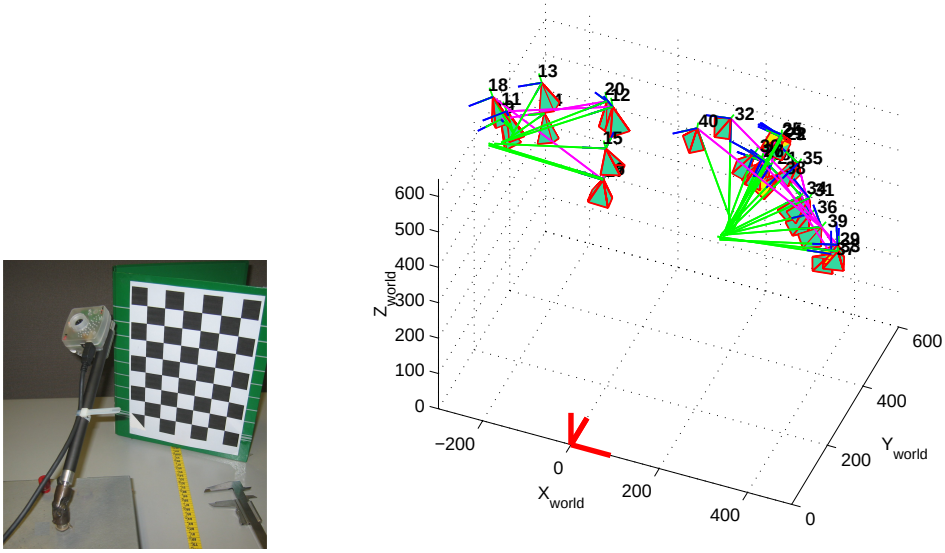


Figure 4.24: Camera reconstructed pose relative to calibration target, with the pivot point at two different positions, showing frame number, camera orientation and the estimated lever arm in green.

small adjustments are easier to make.

To better assess the calibration performance a rotating u-joint was initially used so that a fixed pivot could be used over several turns, enabling the use of standard hand-eye calibration methods for comparison, as seen in fig. 4.24. With this setup a set of 30 images was taken, corresponding to 15 distinct turns about a single pivot point with the chessboard target always in view, placed in 2 different places during image acquisition.

Our method is compared with a standard implementation of the Tsai and Lenz [Tsai1989] hand-eye calibration. Assuming the fixed pivot point and fixed target, the gripper to camera transformation will be the lever arm translation, if the camera rotation is used as the world to gripper transformation.

Table 4.2 presents the results. A total of 40 images where taken, the first 10 were used only to improve the camera calibration set, data set A has 5 turns (10 images) with a single pivot point and set B has 10 turns (20 images) with a distinct fixed pivot point. Results of lever arm estimation, $\mathbf{r} = (r_x, r_y, r_z)$ with $r = \|\mathbf{r}\|$, are shown for our method and for Tsai and Lenz applied to sets A&B, A and B, and $\bar{r}$ is the mean of the distinct

Table 4.2: Translation estimation using two data sets with fixed pivot point

	Our method					Tsai and Lenz					-
	A&B	A	B	$\bar{r}$	σ	A&B	A	B	$\bar{r}$	σ	r_m
r_x	**252.54**	**252.77**	**253.05**	**252.91**	0.14	81.70	**251.16**	**251.79**	**251.48**	0.32	**249±5**
r_y	**26.27**	29.85	**22.28**	**26.07**	3.79	-75.00	28.72	**21.67**	**25.20**	3.52	**25±3**
r_z	**-31.57**	-34.47	**-29.64**	**-32.05**	2.42	793.01	**-28.71**	**-27.78**	**-28.24**	0.46	**-31±3**
r	**255.86**	**256.85**	**255.75**	**256.26**	0.55	800.72	**254.42**	**254.25**	**254.31**	0.09	**252±5**

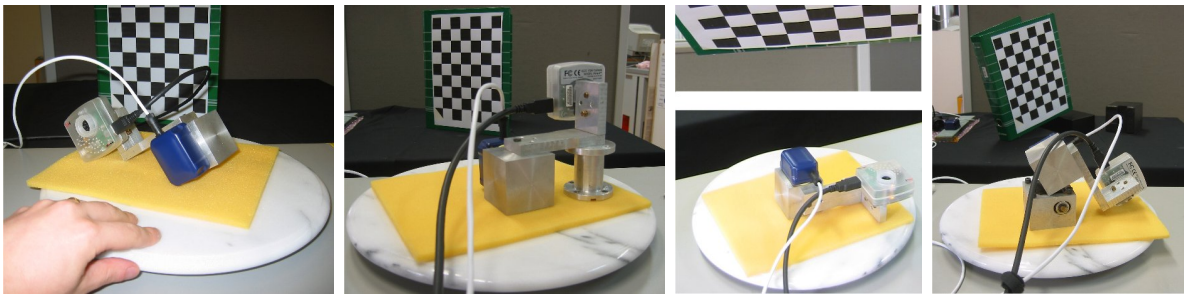


Figure 4.25: System placed on turntable in different poses for translation calibration.

estimates from set A and set B. The values shown in bold fall within the uncertainty of the direct ruler measurement r_m.

Tsai and Lenz clearly has a better performance, since it performs a global optimisation using all the images by considering the pivot point and the target are always fixed. When the method is applied to the complete data set A&B it fails completely since its not applicable. Our method just requires sets of turns between which both the target and pivot point can be repositioned. It is based only on the relative camera motion in each turn, and is therefore more sensitive and prone to errors. But, as we will see in the second example, requires a much simpler setup and can provide a good estimate of the lever arm under controlled conditions.

A second calibration was done with a passive turntable, placing the camera with attached inertial sensors in different poses as shown in figures 4.23 and 4.25, and fine adjusting the position to zero the force sensed by the accelerometers, besides gravity, placing them at the rotation center.

With the passive turntable setup a set of 30 images was taken, corresponding to 15 distinct turns. The accelerometer output was observed while manually forcing rapid turns

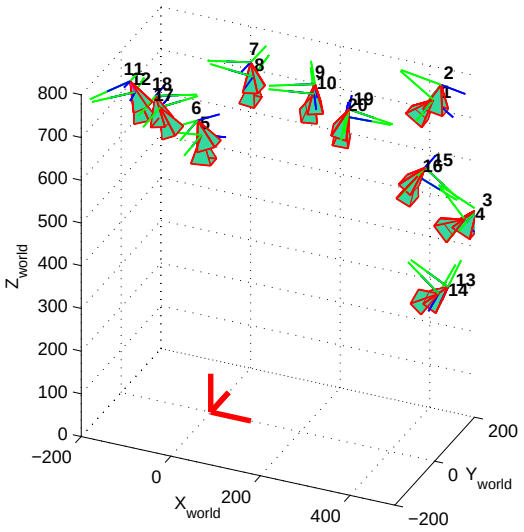


Figure 4.26: Camera reconstructed pose relative to calibration target.

Table 4.3: Translation estimation using turntable

n	1:1:15	1:2:15	1:1:10	1:2:11	5:1:15	5:2:15	mean	σ
r_x	-87.4	-86.7	-92.9	-86.6	-83.0	-83.2	-86.6	3.6
r_y	91.7	91.6	92.0	91.5	93.1	92.1	92.0	0.6
r_z	2.6	1.7	1.8	1.7	6.2	2.8	2.8	1.7
r	126.7	126.1	130.8	126.0	124.9	124.1	126.4	2.3

to adjust their position to the center of rotation. The chessboard target was conveniently placed, and the reconstruction result for the complete set is shown in figure 4.26.

In table 4.3 results are presented for several groupings of sets of measurements (sets are labelled as start:step:end), to better evaluate the estimation performance. Direct measurement of the lever arm indicated a length about 125 ± 10 mm, since the exact position of the accelerometers within the packaged sensor is not known, confirming the estimated value.

The implemented code for translation estimation will be made available in the InerVis Matlab Toolbox [Lobo2006].

4.4 Inertial Cues for Camera Calibration

4.4.1 Focal Distance from Inertial Artificial Horizon and Vanishing Point

In the next chapter we will see how the vertical reference can be used in many computer vision tasks. If the alignment between the camera and inertial sensors is known, the vertical reference can also help the calibration of the camera focal distance.

The key idea is to explore the orthogonality between the gravity vertical reference and some ground plane vanishing point.

Camera calibration using vanishing points has been widely explored, [Kanatani1993] [Wang1991] [Caprile1990] [Brillault1991] [Li1994] amongst others. The novelty in our work is using just one vanishing point, and using the inertial sensors to extract camera pose information. Calibration based on vanishing points is limited since a compromise has to be reached on the quality of each point, but since we require just one vanishing point, the best one can be chosen.

Vanishing point $\boldsymbol{p}_v = (u, v)^\mathsf{T}$, obtained from a set of parallel lines belonging to some levelled plane, and $\hat{\boldsymbol{n}} = (n_x, n_y, n_z)^\mathsf{T}$ taken from (2.2), are conjugate to each other since they correspond to 3D orthogonal sets of parallel lines. From (3.2) the focal distance f can be estimated as

$$f = -\frac{n_x u + n_y v}{n_z} \tag{4.33}$$

With a suitable calibration target scene, where ground plane parallel lines can be easily found, the focal distance can be estimated using (4.33). The image center is assumed to be fixed and known, and (u, v) are given in image centered coordinates. If no prior calibration is done to determine the image center, non-imaging techniques, such as numerical center of image or sensor coordinates, are used. The implications of this assumption depend on the camera quality and variable parameters [Willson1994].

The orthogonality of two levelled plane sets of parallel lines, when using two vanishing points, is replaced here by the orthogonality between vertical lines, with vanishing point $(n_x, n_y, n_z)^\mathsf{T}$, and a set of levelled parallel lines, with vanishing point $(u, v, f)^\mathsf{T}$. This

implies that the alignment between the IMU and the camera has to be known from construction or previous calibration.

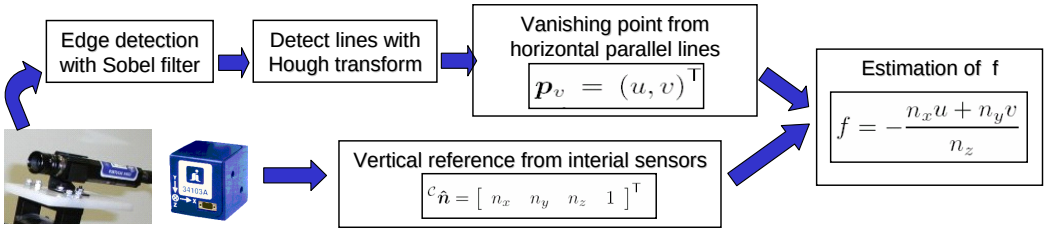


Figure 4.27: Focal distance estimation algorithm.

Figure 4.27 summarises the focal distance estimation method.

Error Sensitivity

The effect of errors in the vertical reference on the estimated focal distance can be seen by studying the Jacobian matrix

$$\begin{aligned} \boldsymbol{J} &= \left[\begin{array}{ccccc} \frac{\partial f}{\partial n_x} & \frac{\partial f}{\partial n_y} & \frac{\partial f}{\partial n_z} & \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \end{array} \right] \\ &= \left[\begin{array}{ccccc} -\frac{u}{n_z} & -\frac{v}{n_z} & \frac{n_x u + n_y v}{n_z^2} & -\frac{n_x}{n_z} & -\frac{n_y}{n_z} \end{array} \right] \end{aligned} \tag{4.34}$$

Considering a good pose tilting down about 45 *deg* with a not too distant vanishing point, having $\hat{\boldsymbol{n}} = (0, -0.70, 0.71)^\mathsf{T}$ and $(u, v)^\mathsf{T} = (100, 1000)^\mathsf{T}$, a 1 *deg* error in the vertical reference would perturb the estimated focal distance f by $\pm\Delta f$ with

$$\Delta f = \sin(1)\frac{n_x u + n_y v}{n_z^2} \approx 24.2 \tag{4.35}$$

This means a 2.5% error in the estimated value of $f \approx 986$. This error will degrade when more ill conditioned poses are used, and the solution degenerates when observability of a ground plane vanishing point is missing or the camera is perfectly horizontal, with the horizon line through the image center.

Results

The ground plane segmentation algorithm was implemented on a mobile robot equipped with a stereo active vision system with inertial sensors at the center of the baseline.

Figure 4.28 shows the setup where an inertial system prototype built at our lab [Lobo2002MSc] was coupled to a camera stereo rig to carry out the tests.

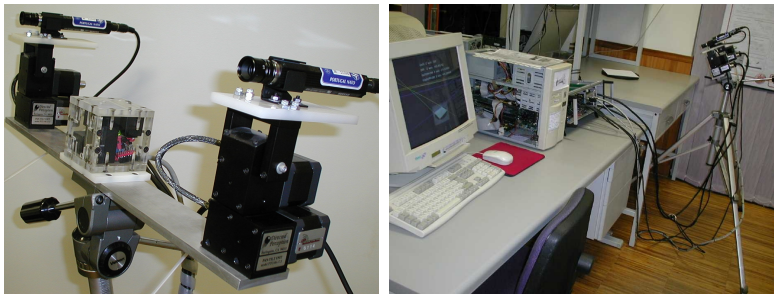


Figure 4.28: Experimental setup showing the stereo camera rig with IMU based on low cost sensors..

To test the estimation of f using one vanishing point and the vertical reference, the Camera Calibration Toolbox for Matlab [Bouguet2006] was used to provide a standard camera calibration method.

The calibration was performed with 20 images of a chessboard target in several positions, as seen in fig. 4.29. Without changing the camera, the chessboard target was removed and the calibration was performed using just one vanishing point and the inertial vertical reference. Two target positions with a near vanishing point were used, as seen in fig. 4.29, and 100 samples taken at each position. From fig. 4.30 and table 4.4 we can see that the proposed method provides a good estimate of f, within the uncertainty of the standard method used.

Table 4.4: Estimation of f

	mean	σ
20 images of chessboard target	617.57	10.36
$\hat{\boldsymbol{n}}$ & vanishing point	613.02	2.62

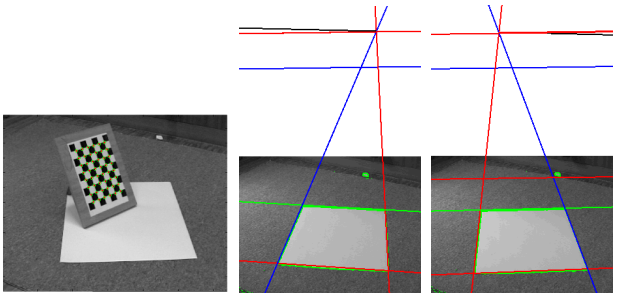


Figure 4.29: One of the 20 images used in the calibration, and estimation of f at two target positions with a near vanishing point, showing horizon line with initial guess value of f (lower) and correct horizon line given by inertial vertical reference (top).

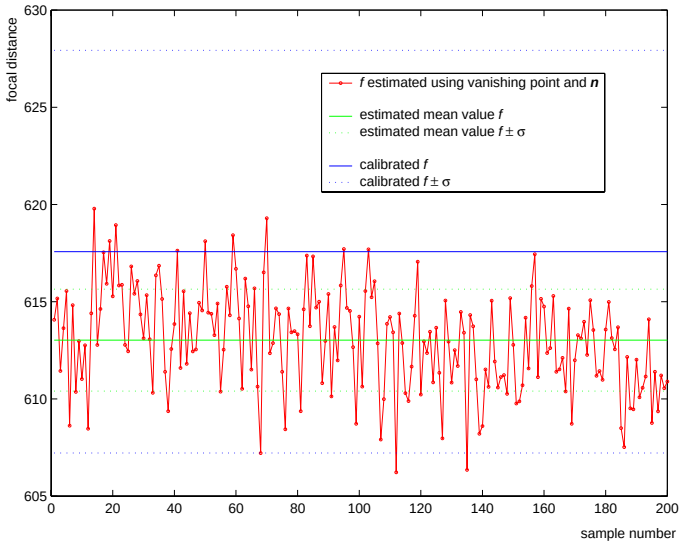


Figure 4.30: Estimation of f with just one vanishing point and $\hat{\boldsymbol{n}}$, compared with camera calibration results. Two target positions with a near vanishing point, 100 samples taken at each position.

The main sources of error are the vanishing point instability, evidenced by the stepwise results obtained in other tests [Lobo2001MFI], and the noise in the vertical reference provided by the low cost accelerometers. The results show that the proposed method is feasible. Due to its simplicity, it can be performed on-the-fly by a mobile robot in a man-made environment, where ground plane parallel lines can be easily detected. It can also aid 3D modelling and reconstruction by providing extra information about focal distance

when digitally acquiring an image, as in [Coorg1998].

4.5 Conclusions

We have seen how a simple calibration can be made with off-the-shelf cameras and inertial sensors to have a useful integrated system.

Using a pendulum with an encoded shaft, inertial sensor alignment, bias and scale factor can be estimated, for both accelerometers and gyros. With a set of static poses observing a vertical target, full camera calibration can be performed using standard techniques, and inertial sensor to camera rotation can estimated as well by registering the inertial sensed gravity. With a simple passive turntable and with $2n$ static poses of n rotations about the inertial sensor, the translation between the two sensors can also be estimated.

This only relates the accelerometers' with the cameras, the relative pose between the accelerometers and gyros can be done independently, using horizontal and vertical rotation axis as described above.

The cross calibration method works well in estimating rotation and, depending on the setup and restrictions when performing the calibration, weighing input errors can reduce the error in the estimation. The translation estimation is sensitive to the chosen target position, and care has to be taken so that the geometric configuration does not magnify the error in the visual target pose onto the final lever arm estimation.

Lever arm calibration can also be accomplished using standard Hand/Eye calibration [Daniilidis1999], like the Tsai and Lenz implementation used above for comparison [Tsai1989]. These methods, applied here in a simplified case where the camera rotation is used as the base-to-hand transformation, are clearly more stable. Our method only uses the relative camera motion in each turn, but Hand/Eye methods use the full camera and hand pose data over the complete data set. But they are also more restrictive on the setup. A simple turntable is no longer sufficient, since a fixed pivot point has to be maintained. A passive double gimbal might prove useful, but would have to accommodate for proper centering of the system, and using an active controlled manipulator might be

better. Our aim however is to have a simple procedure to estimate the lever arm, that can be performed without complicated equipment, and complement the simple procedure used for camera and rotation calibration. As suggested above, another alternative to the passive turntable is using a *magic arm* to position the system and rotate about some axis, however when positioning on the turntable small adjustments are easier to make.

Exploring the orthogonality between the vertical reference and vanishing points of horizontal lines, camera focal distance was estimated using only one vanishing point. This allows the best vanishing point to be chosen, and is less imposing on the availability of scene vanishing points. An integrated accelerometer and imaging sensor could use this method to estimate focal distance, relying on the automatic detection of one vanishing point of a set of horizontal lines, with high probability of occurring at specific camera poses in man-made structured environments. When applied to mobile robots, the vanishing point can also provide an external bearing for the navigation frame of reference. Calibration methods using specific calibration targets and multiple images can provide more precise focal distance estimates. The main sources of error in this method are the uncertainty in the vanishing point estimation, the assumed alignment of the inertial sensors and the accelerometer noise.

Chapter 5

Using Gravity as a Vertical Reference

We have already seen that gravity provides an absolute vertical reference (chapter 3), and how it can be used for camera calibration (previous chapter). In this chapter we will explore the use of this vertical reference in monocular and stereo vision. Low level monocular image processing can use the vertical reference to tune edge detection to find relevant features such as vertical or horizontal scene elements. In stereo vision the vertical reference provides an external restriction when considering ground plane or levelled plane point correspondence in the stereo pair. Results are presented for ground plane segmentation of feature points, vertical line detection and 3D vertical line segmentation.

Another approach explored was to use standard computer vision techniques to compute depth maps, and than rotate and align them using the vertical reference. The depth map points are mapped to the a vertically aligned world frame of reference. In order to detect the ground plane, an histogram is performed for the different heights. Taking the ground plane as a reference plane for the acquired maps, the fusion of multiple maps reduces to a 2D translation and rotation problem. Results are presented for the rotation followed by ground plane detection.

5.1 Unit Sphere Vertical Reference from Gravity

As we saw in 3.1 inertial data provide cues to determine the vision system's attitude. When the system is motionless or subject to constant speed, the accelerometers give the direction of the gravity vector. We can therefore determine the vertical unit vector normal to local ground levelled plane, but rotations within the horizontal plane are not sensed.

In 2.2.1 we expressed the vertical reference $\hat{\boldsymbol{n}}$ in the inertial sensor frame of reference $\{\mathcal{I}\}$, given by (2.2) using the accelerometer data.

As explained in 2.2.3, by performing the rotation update using the IMU gyro data, gravity can be separated from the sensed acceleration. In this case $\hat{\boldsymbol{n}}$ is given by the rotation update, but must be monitored using the low pass filtered accelerometer signals, for which (2.2) still holds, to reset the accumulated drift.

In order to have a camera unit sphere vertical reference from gravity we must rotate ${}^{\mathcal{I}}\hat{\boldsymbol{n}}$ to ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$ using the rotation calibration result presented in the previous chapter, i.e.,

$$ {}^{\mathcal{C}}\hat{\boldsymbol{n}} = \mathring{\mathbf{q}}\, {}^{\mathcal{I}}\hat{\boldsymbol{n}}\, \mathring{\mathbf{q}}^{*} \tag{5.1} $$

This unit sphere vertical reference, ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$, will be used in the following sections to determine the ground plane, define a robot navigation frame of reference and set the collineation of ground plane points in stereo vision. This will enable the detection of 3D ground plane patches observed by a stereo system, as well as 3D vertical line detection. This will be accomplished by solving the correspondence problem for ground plane points, and using the image projected vertical to segment vertical features. But first we will analyse the error in the accelerometer derived vertical reference.

5.1.1 Vertical Reference Error

Considering a linear model for the accelerometers we have

$$ \boldsymbol{a}_{measured} = \boldsymbol{M}\boldsymbol{a}_{real} + \boldsymbol{b} \tag{5.2} $$

were $\boldsymbol{M}$ incorporates scale factor, cross axis sensitivity inherent to the sensing element and also due to sensor misalignment, and $\boldsymbol{b}$ offset bias. Estimates for $\boldsymbol{M}$ and $\boldsymbol{b}$ might

be provided by the manufacturer, or can be obtained by sensor calibration as previously described.

But temperature drift, power supply ripple interference, and thermal noise will always degrade the signal, and $\boldsymbol{M}$ and $\boldsymbol{b}$ will not remain static. Part of this noise can be taken as having zero mean, but for instance temperature drift does not, and temperature compensation might be required in some applications.

Since we are measuring gravity, any mechanical vibrations and oscillations will introduce additional error. Low pass filtering has to be used, and for more dynamic situations gyro rotation update is required to have stabilised gravity direction. In some applications using magnetic sensors with accelerometers provides a good solution for pose tracking, exploring the different dynamics and noise characteristics of each sensor [Caruso1998], and also providing an azimuth bearing to keep track of rotations with a vertical axis not sensed by the accelerometers. The depth map registration proposed in chapter 6 uses magnetic sensors to complement the inertial data and provide a rotation update.

A data set was taken using the inertial system prototype built at our lab [Lobo2002MSc] (fig. 5.3) that uses a signal conditioned three-axis accelerometer [SummitInstruments] [AnalogDevices]. Bias and cross-axis sensitivity calibration data were available and used.

A set of 6400 measurements were taken with the sensor at rest, with no filtering. The obtained covariance matrix was

$$V(\boldsymbol{n}) = \begin{bmatrix} 0.5873 & -0.0069 & 0.0102 \\ -0.0069 & 0.5675 & 0.0071 \\ 0.0102 & 0.0071 & 0.0003 \end{bmatrix} \times 10^{-4} \tag{5.3}$$

with eigenvalues

$$\sigma_1^2 = 0.5895 \times 10^{-4} \geq \sigma_2^2 = 0.5655 \times 10^{-4} \geq 0 \tag{5.4}$$

The root-mean-square angle error is given by

$$\sigma_\theta = tan^{-1}\left(\sqrt{trV(\boldsymbol{n})}\right) = tan^{-1}\left(\sqrt{\sigma_1^2 + \sigma_2^2}\right) \tag{5.5}$$

and for this data set $\sigma_\theta = 0.6130$ *deg*. Low pass filtering the accelerometer data improved the estimate significantly, lowering the error to $\sigma_\theta = 0.1907$ *deg* when applying a Butterworth 5^{th} order filter with 10 *Hz* cutoff frequency.

Another set of measurements was made on a mobile robot performing back-and-forth motion, with no filtering. The obtained covariance matrix was

$$V(\boldsymbol{n}) = \begin{bmatrix} 0.6928 & 0.1094 & 0.3086 \\ 0.1094 & 0.7763 & 0.0183 \\ 0.3086 & 0.0183 & 0.1388 \end{bmatrix} \times 10^{-4} \tag{5.6}$$

with eigenvalues

$$\sigma_1^2 = 0.9144 \times 10^{-4} \geq \sigma_2^2 = 0.6934 \times 10^{-4} \geq \sigma_3^2 = 0.10 \times 10^{-7} \tag{5.7}$$

and for this data set the expected angle error $\sigma_\theta = 0.7265$ *deg*. Low pass filtering the accelerometer data improved the estimate significantly, lowering the error to $\sigma_\theta = 0.4611$ *deg*.

This results indicate that the accelerometer data provide a useful vertical reference for robotic systems. To deal with motion, proper low pass filtering if performed, taking into account the motion characteristics, but very slow robotic motion will present a problem.

5.1.2 Ground Plane

We saw in chapter 3 how the gravity vertical reference indicates the vertical vanishing point and horizon line in the camera frame of reference. It also gives the orientation of levelled planes, i.e., planes that vanish towards the horizon line.

Consider a world point ${}^{\mathcal{C}}\boldsymbol{P}$, given in a camera centered referential $\{\mathcal{C}\}$, that belongs to the ground plane. The plane equation is given by

$${}^{\mathcal{C}}\hat{\boldsymbol{n}}.{}^{\mathcal{C}}\boldsymbol{P} + d = 0 \tag{5.8}$$

where d is the distance from the origin to the ground plane, i.e., the system height. In some applications it can be known or imposed by the physical mount, or determined using stereo as shown bellow. The ground plane can therefore be determined in the camera system $\{\mathcal{C}\}$ frame of reference.

5.1.3 Robot Navigation Frame of Reference

When detecting world features, a convenient frame of reference has to be established. As we saw in section 3.3, we can consider a moving robot navigation frame of reference $\{\mathcal{N}\}$, aligned by the ground pane as shown in fig. 6.1. The vertical unit vector $\hat{\boldsymbol{n}}$ and system height d can be used to define $\{\mathcal{N}\}$, by choosing ${}^{\mathcal{N}}\hat{\boldsymbol{x}}$ to be coplanar with ${}^{\mathcal{C}}\hat{\boldsymbol{x}}$ and ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$ in order to keep the same heading, we have

$$ {}^{\mathcal{N}}\boldsymbol{P} = {}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}\boldsymbol{P} \tag{5.9} $$

where

$$ {}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}} = \begin{bmatrix} \sqrt{1-n_x^2} & \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{-n_x n_z}{\sqrt{1-n_x^2}} & 0 \\ 0 & \frac{n_z}{\sqrt{1-n_x^2}} & \frac{-n_y}{\sqrt{1-n_x^2}} & 0 \\ n_x & n_y & n_z & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}} = \begin{bmatrix} \sqrt{1-n_x^2} & 0 & n_x & -n_x d \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{n_z}{\sqrt{1-n_x^2}} & n_y & -n_y d \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} & \frac{-n_y}{\sqrt{1-n_x^2}} & n_z & -n_z d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.10} $$

This is obtained as follows. Consider a frame of reference $\{\mathcal{N}_c\}$ with origin at the camera optical center and ${}^{\mathcal{N}_c}\hat{\boldsymbol{x}}$ coplanar with ${}^{\mathcal{C}}\hat{\boldsymbol{x}}$ and ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$ in order to keep the same heading. A simple rotation R maps the two frames of reference as follows

$$ {}^{\mathcal{C}}\boldsymbol{P} = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} .{}^{\mathcal{N}_c}\boldsymbol{P} = \begin{bmatrix} \hat{\boldsymbol{r}}_1 & \hat{\boldsymbol{r}}_2 & \hat{\boldsymbol{r}}_3 & 0 \\ & 0 & & 1 \end{bmatrix} .{}^{\mathcal{N}_c}\boldsymbol{P} \tag{5.11} $$

where $\hat{\boldsymbol{r}}_1$, $\hat{\boldsymbol{r}}_2$ and $\hat{\boldsymbol{r}}_3$ are the X, Y, and Z axis of $\{\mathcal{N}_c\}$ given in the camera frame of reference $\{\mathcal{C}\}$. But the Z axis of $\{\mathcal{N}_c\}$ is just the vertical given by the inertial sensors:

$$ {}^{\mathcal{N}_c}\hat{\boldsymbol{z}} = \hat{\boldsymbol{r}}_3 = {}^{\mathcal{C}}\ \hat{\boldsymbol{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \tag{5.12} $$

But there are infinite possibilities for the $\{\mathcal{N}_c\}$ X and Y axis, since $\hat{\boldsymbol{n}}$ only defines the XY plane, but no heading within this plane. The X axis of $\{\mathcal{N}_c\}$ can be chosen to be coplanar with $\{\mathcal{C}\}$ X and $\hat{\boldsymbol{r}}_3$ axis, keeping the same heading, so we have:

$$\hat{\boldsymbol{r}}_1 = a.\hat{\boldsymbol{x}} + b.\hat{\boldsymbol{r}}_3 = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} a + bn_x \\ bn_y \\ bn_z \end{bmatrix} \tag{5.13}$$

since $\hat{\boldsymbol{r}}_1$ is a unit vector we have:

$$||\hat{\boldsymbol{r}}_1|| = a^2 + 2abn_x + b^2 = 1 \tag{5.14}$$

and since $\hat{\boldsymbol{r}}_1$ is orthogonal to $\hat{\boldsymbol{r}}_3$ we have:

$$\hat{\boldsymbol{r}}_1.\hat{\boldsymbol{r}}_3 = 0 = \begin{bmatrix} a + bn_x \\ bn_y \\ bn_z \end{bmatrix}^T \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = n_x a + b = 0 \tag{5.15}$$

From the above equation we get:

$$\hat{\boldsymbol{r}}_1 = \begin{bmatrix} \sqrt{1 - n_x^2} \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} \end{bmatrix} \tag{5.16}$$

Finally we have that $\hat{\boldsymbol{r}}_2$ is orthogonal to both $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_3$, and is obtained with the external product:

$$\hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{r}}_3 \times \hat{\boldsymbol{r}}_1 = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \times \begin{bmatrix} \sqrt{1 - n_x^2} \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{n_z}{\sqrt{1-n_x^2}} \\ -\frac{n_y}{\sqrt{1-n_x^2}} \end{bmatrix} \tag{5.17}$$

and so the transformation matrix is given by:

$${}^{c}\boldsymbol{T}_{\mathcal{N}_c} = \begin{bmatrix} \sqrt{1 - n_x^2} & 0 & n_x & 0 \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{n_z}{\sqrt{1-n_x^2}} & n_y & 0 \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} & -\frac{n_y}{\sqrt{1-n_x^2}} & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.18}$$

The robot navigation frame of reference $\{\mathcal{N}\}$ is just $\{\mathcal{N}_c\}$ translated by $\begin{bmatrix} 0 & 0 & d & 1 \end{bmatrix}^\mathsf{T}$, as presented in equation (5.10).

If a heading reference is available, then $\{\mathcal{N}\}$ should not be restricted to having ${}^{\mathcal{N}}\hat{\boldsymbol{x}}$ coplanar with ${}^{\mathcal{C}}\hat{\boldsymbol{x}}$ and ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$, but use the known heading reference. As previously seen, vanishing points $\hat{\boldsymbol{m}}_i$ of levelled planes are orthogonal to the vertical $\hat{\boldsymbol{n}}$, *i.e.*, $\hat{\boldsymbol{m}}_i.\hat{\boldsymbol{n}} = 0$. In scenes of man made environments the vanishing points can provide a heading reference. Proceeding as above, but replacing (5.13) with the heading given by the vanishing point $\hat{\boldsymbol{m}}$ we have

$$\hat{\boldsymbol{r}}_1 = \hat{\boldsymbol{m}} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \tag{5.19}$$

as before, since $\hat{\boldsymbol{r}}_2$ is orthogonal to both $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_3$, we have

$$\hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{r}}_3 \times \hat{\boldsymbol{r}}_1 = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \times \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} n_y m_z - n_z m_y \\ n_z m_x - n_x m_z \\ n_x m_y - n_y m_x \end{bmatrix} \tag{5.20}$$

and so the transformation matrix using the vanishing point heading is given by

$${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}_c} = \begin{bmatrix} m_x & n_y m_z - n_z m_y & n_x & 0 \\ m_y & n_z m_x - n_x m_z & n_y & 0 \\ m_z & n_x m_y - n_y m_x & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.21}$$

Translating $\{\mathcal{N}_c\}$ as before we have

$${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}} = \begin{bmatrix} m_x & n_y m_z - n_z m_y & n_x & -n_x d \\ m_y & n_z m_x - n_x m_z & n_y & -n_y d \\ m_z & n_x m_y - n_y m_x & n_z & -n_z d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.22}$$

and

$$ {}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}} = \begin{bmatrix} m_x & m_y & m_z & 0 \\ n_y m_z - n_z m_y & n_z m_x - n_x m_z & n_x m_y - n_y m_x & 0 \\ n_x & n_y & n_z & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.23} $$

Providing suitable vanishing points can be extracted from the scene, we are able to have $\{\mathcal{N}\}$ coherent with the inertial vertical and the scene heading. Using the robot's odometry, the inertial sensors and landmark matching, conversion to the world fixed frame of reference $\{\mathcal{W}\}$ can be accomplished.

5.1.4 Ground Plane in Stereo Vision

As seen above, the vertical reference provides the orientation of the ground plane relative to the camera system. With stereo vision, visual fixation of a ground plane point can be used to determine the ground plane distance [Dias1995] [Dias1998].

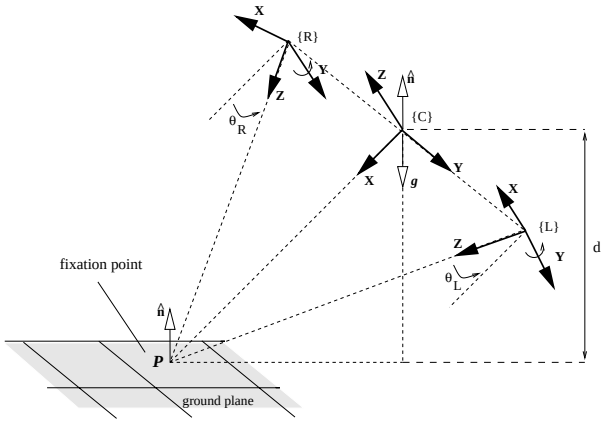


Figure 5.1: Ground plane point $\boldsymbol{P}_f$ fixated by stereo system.

For this stereo system, the camera frame of reference $\{\mathcal{C}\}$ is at the middle of the baseline with x pointing forward, as seen in fig. 5.1. Assuming a vision system with controlled symmetric verge angle θ and baseline b, fixated in a point ${}^{\mathcal{C}}\boldsymbol{P}_f$ that belongs to the ground plane, the distance d is given by the projection of ${}^{\mathcal{C}}\boldsymbol{P}_f$ on the gravity vector direction

$$d = -{}^{\mathcal{C}}\hat{\boldsymbol{n}}.{}^{\mathcal{C}}\boldsymbol{P}_f = -{}^{\mathcal{C}}\hat{\boldsymbol{n}}.\begin{bmatrix} \frac{b}{2}\cot\theta \\ 0 \\ 0 \end{bmatrix} = -n_x\frac{b}{2}\cot\theta \tag{5.24}$$

as can easily be seen in fig. 5.1. In this figure there is no lateral inclination, but (5.24) is valid for any angle, since the attitude is given by ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$.

The ground plane can therefore be determined in the camera system $\{\mathcal{C}\}$ frame of reference, using the plane orientation, given by the inertial sensors, and the plane height from some *apriori* knowledge, or by fixating the vision system on a ground plane point. All ground plane geometric parameters are therefore determined. The levelled navigation frame of reference can de shifted to have $Z = 0$ for the ground plane.

5.1.5 Collineation of Ground Plane Points

To analyse how ground plane points are projected onto the image plane, consider a world point $\boldsymbol{P} = (X, Y, 0, 1)^{\mathsf{T}}$ that belongs to the ground plane (i.e., $Z = 0$). The projection onto the camera image plane is given by

$$s\boldsymbol{p}_i = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \boldsymbol{C}\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix}_{4\times4}\boldsymbol{P} = [...]_{3\times3}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{5.25}$$

where $\boldsymbol{p}_i$ is the projective image point, s an arbitrary scale factor, $\boldsymbol{C}$ and $\boldsymbol{R}\ \ \boldsymbol{t}$ are the camera intrinsic and extrinsic parameters.

From the above equation we can see that there is a fixed mapping between ground plane points and image points. This mapping is called a collineation or planar homography of points. A ground plane point is related to the camera image by a collineation $\boldsymbol{H_c}$:

$$s\boldsymbol{p}_i = \boldsymbol{H_c}.\widetilde{\boldsymbol{P}} \tag{5.26}$$

where $\widetilde{\boldsymbol{P}} = (X, Y, 1)^{\mathsf{T}}$ and

$$\boldsymbol{H_c} = \boldsymbol{C}\begin{bmatrix} \boldsymbol{r_1} & \boldsymbol{r_2} & \boldsymbol{t} \end{bmatrix} \tag{5.27}$$

with $\boldsymbol{r_i}$ denoting the i^{th} column of the rotation matrix $\boldsymbol{R}$.
For a stereo system we can express the collineation between ground plane points and the left and right cameras

$$s\boldsymbol{p}_{li} = \boldsymbol{H_l}.\widetilde{\boldsymbol{P}} \quad and \quad s\boldsymbol{p}_{ri} = \boldsymbol{H_r}.\widetilde{\boldsymbol{P}} \tag{5.28}$$

where $\boldsymbol{p}_{li}$ and $\boldsymbol{p}_{ri}$ are the left and right projective image points.

We can consider a direct mapping $\boldsymbol{H}$ of ground plane points between the stereo pair. $\boldsymbol{H}$ can be obtained by calibration using know ground plane points [Hartley2000], or using (5.27) and known camera intrinsic and extrinsic parameters $\boldsymbol{C}$ and $\boldsymbol{R}\ \ \boldsymbol{t}$. For the direct mapping $\boldsymbol{H}$ of right image points to the left image we have

$$s\boldsymbol{p}_{li} = \boldsymbol{H}.\boldsymbol{p}_{ri} = \boldsymbol{H_l}.\boldsymbol{H_r}^{-1}.\boldsymbol{p}_{ri} \tag{5.29}$$

To obtain $\boldsymbol{H}$ we must first compute $\boldsymbol{H_l}$ and $\boldsymbol{H_r}$. From (5.27) and using ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}}$ obtained from the inertial data and ${}^{\mathcal{L}}\boldsymbol{T}_{\mathcal{C}}$ obtained form the geometric setup, $\boldsymbol{H_l}$ is given by

$$\boldsymbol{H_l} = \boldsymbol{C_L}\left[\begin{array}{ccc} \boldsymbol{r_1} & \boldsymbol{r_2} & \boldsymbol{t} \end{array}\right]_L = \boldsymbol{C_L}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.{}^{\mathcal{L}}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}}.\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.30}$$

and proceeding analogously for the right camera we obtain $\boldsymbol{H_r}$.

From (5.29) we have that the collineation between left and right images of ground plane points, for a system with symmetric verge angle θ and baseline b, is given by

$$\boldsymbol{H} = \boldsymbol{H_l}.\boldsymbol{H_r}^{-1} = \begin{bmatrix} -\frac{2n_x b\cos\theta\sin\theta + n_y b + 2d - 4d\cos^2\theta}{-n_y b + 2d} & \frac{2bn_z\cos\theta}{-n_y b+2d} & f\frac{-2\cos\theta(2d\sin\theta + bn_x\cos\theta)}{-n_y b+2d} \\ 0 & 1 & 0 \\ 2\frac{2d\sin\theta\cos\theta - n_x b + n_x b\cos^2\theta}{f(-n_y b+2d)} & \frac{2bn_z\sin\theta}{f(-n_y b+2d)} & -\frac{2n_x b\cos\theta\sin\theta + n_y b - 4d\cos^2\theta + 2d}{-n_y b+2d} \end{bmatrix} \tag{5.31}$$

where f is the camera focal distance, $(n_x, n_y, n_z)^{\mathsf{T}}$ the vertical reference provided by the inertial sensors given in the camera system frame of reference $\{\mathcal{C}\}$ (with origin at the

middle of the baseline), and d the system height to the ground plane. This equation will be fundamental for the world feature detection methods described in following sections.

This collineation can also be computed for other planes. Consider a mobile robot going up a slope. In this case the levelled ground plane is no longer relevant, but we can consider the local planar patch with normal $\boldsymbol{n}_s \neq \boldsymbol{n}$ given by the robot's steady state tilt and proceed as before.

5.2 3D Ground Plane Patch Detection

5.2.1 Stereo Correspondence of Ground Plane Points and 3D Position

Since we know the collineation of the ground plane image points from (5.31), image points can be tested across the stereo pair, identifying the ground plane points, and determining their 3D position. An algorithm for the 3D reconstruction of image detected features can be formulated. For each detected point in the right image $\boldsymbol{p}_{r_i}$, map it to the other image using the known collineation. The correspondent point in the left image if found by parsing all the left image detected points of interest $\boldsymbol{p}_{l_j}$ and testing an allowed neighbourhood window for a match, i.e., find j such that

$$\boldsymbol{p}_{l_j} = \boldsymbol{H}.\boldsymbol{p}_{r_i} \pm \boldsymbol{\delta} \tag{5.32}$$

If there is a match, the point belongs to the ground plane. If there is no match the point must be something other than the floor, possibly an obstacle. Figure 5.2 summarises the ground plane segmentation method. If the detected interest points are very dense, false positives will occur, since it will be easy to have some other point in the same neighbourhood. To overcome this, 2D correlation is performed over a small region around both image points.

From (5.28) the 3D position ${}^{\mathcal{N}}\boldsymbol{P} = (X, Y, 0, 1)^{\mathsf{T}}$ of this ground plane point is given by

$${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = \boldsymbol{H}_{\boldsymbol{r}}^{-1}\boldsymbol{p}_{r_i} \tag{5.33}$$

where ${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = (X, Y, 1)^{\mathsf{T}}$.

Errors in the estimated vertical reference will increase the uncertainty, but since the method relies on a neighbourhood test, it maintains robustness in detecting points up to the tolerance of the used search window size, and than breaks down. The 3D mapping error however will degrade with increasing error in the vertical reference. A statistical map of the detected features has to be built to deal with the uncertainty.

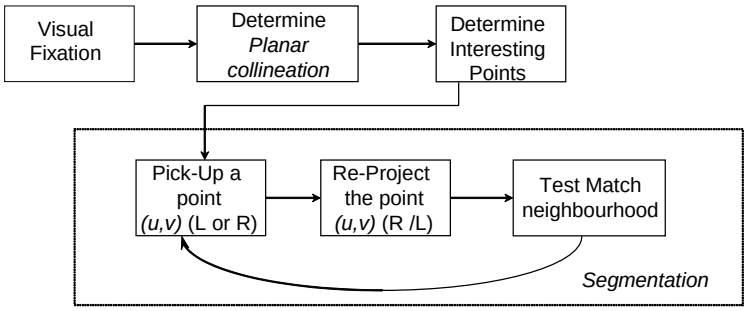


Figure 5.2: Ground plane segmentation algorithm.

5.2.2 Results

The ground plane segmentation algorithm was implemented on a mobile robot equipped with a stereo active vision system with inertial sensors at the center of the baseline.

Figure 5.3 the setup where an inertial system prototype built at our lab [Lobo2002MSc] was coupled to a camera stereo rig to carry out the tests.

The points shown in fig. 5.4 were obtained using SUSAN [Smith1997] corner detector. The points of interest in the right image were than parsed as described in the previous section. Grahams Algorithm [Rourke1993] was used for computation of the convex polygon involving the set of points. Fig. 5.5 also shows some frames from a ground plane detection sequence obtained with the system on a mobile robot, and corresponding VRML view of the ground patch.

For visualisation of the detected ground points a VRML world was generated [Ames1997]. The identified ground plane patch was mapped onto the 3D scene, as seen in fig. 5.5. The complete sequence was processed, generating polygons corresponding to the identified

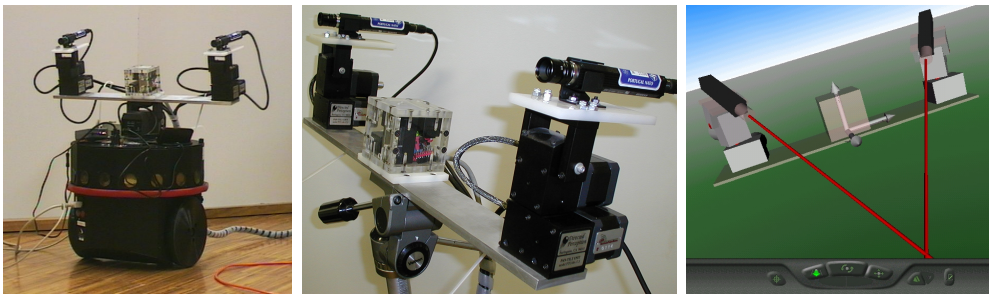


Figure 5.3: Experimental setup showing the mobile robot, the stereo camera rig with IMU based on low cost sensors, and a VRML model of the system used to map the 3D features.

ground plane patch for each frame. To update the VRML world on-the-fly, only the ground patch vertex points need to be sent, so that the polygon can be rendered. When bandwidth is not a problem the segmented image patch can also be sent and placed onto the polygon. VRML opens many other possibilities such as tele-operation or path-planning environments.

Adjusting a convex polygon to the set of points can lead to erroneous ground patch segmentation. Some changes have to be made to the algorithm and special cases taken into account, such as having multiple isolated polygons or allowing for non-convexity when points are too far apart and an obstacle could be in the way.

The results show that the method works, but is very dependant on texture so that feature points can be detected. There are many initial feature points, but only a few are correctly detected as ground plane points, with many false negatives. If instead of detecting the ground plane, an obstacle detection was being done, these unmatched points could be perceived as obstacles. This can be avoided by making assumptions on the minimum size of obstacles and detected point density. This method enables fast processing of images and feature matching across the stereo pair, since the ground plane restriction is used to limit the search space.

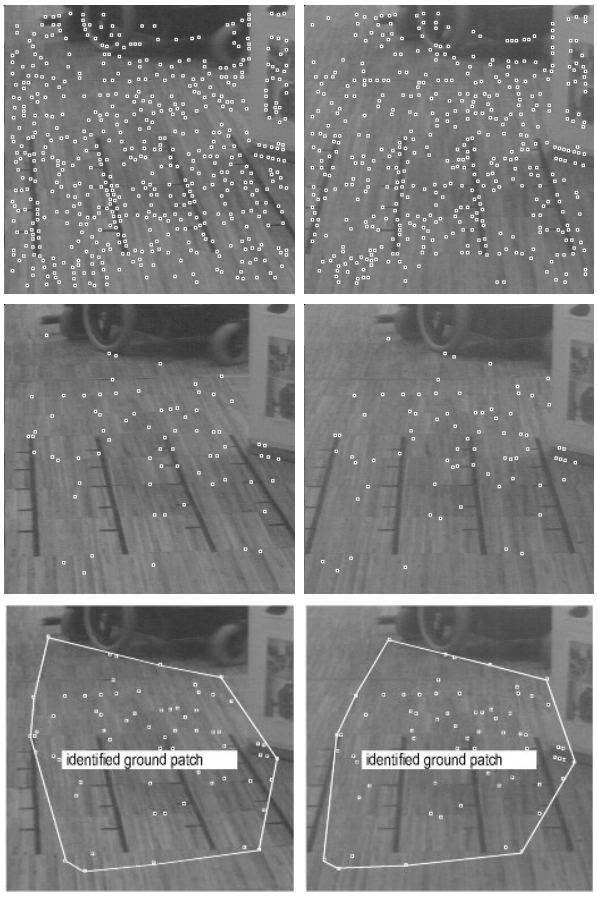


Figure 5.4: Ground plane detection. The system was initially fixated on a ground plane point and the parameters extracted. The figures show the processing steps with: **a**) stereo images with a set of initial points; **b**) detected ground points; **c**) identified ground patch.

5.3 3D Vertical Line Detection

5.3.1 Image Line Segmentation

Knowing the vertical, the vanishing point of all image lines that correspond to world vertical features is known. This vanishing point is at infinity when there is no tilt, and the vertical lines are all parallel in the image. For small tilt values, the vertical lines can be taken as parallel, speeding up the detection process. Based on this assumption, the vertical line segments found in the image will be parallel to the local image vertical $\hat{\boldsymbol{n}}_i$, the normalised image projection of the vertical $\hat{\boldsymbol{n}}$. The image vertical reference corresponds

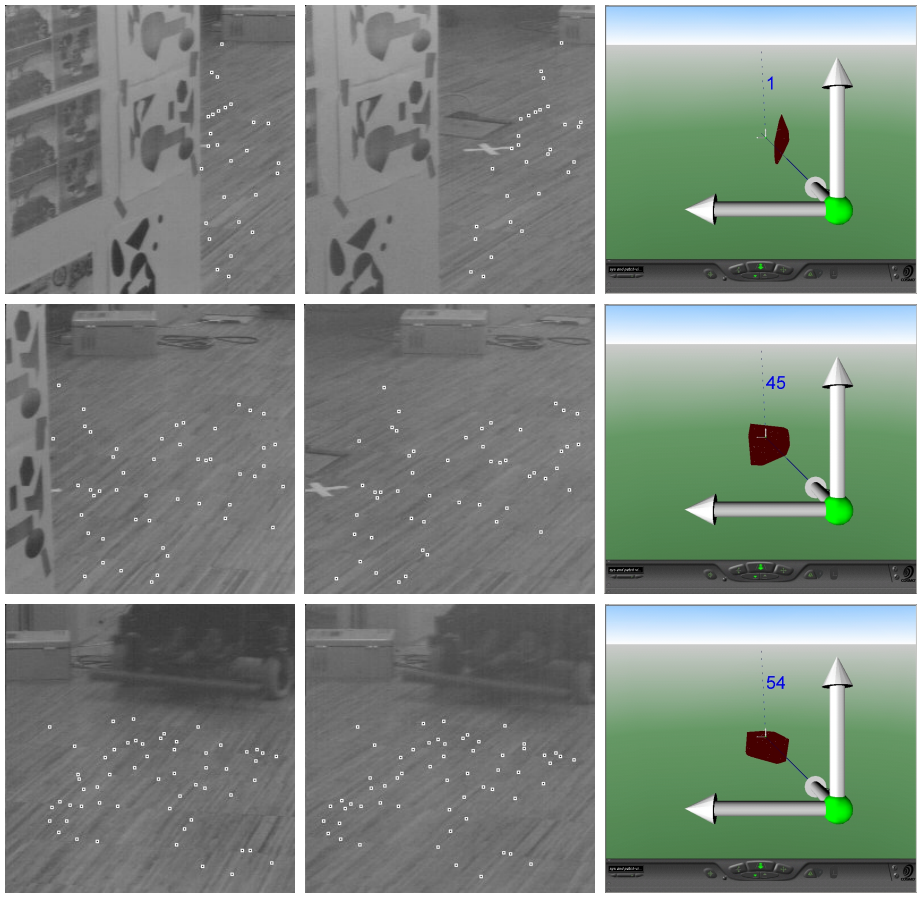


Figure 5.5: Frames from ground plane detection sequence, with the VRML view of the ground patch shown on the right side.

to the unit sphere projection of the vanishing point of all 3D vertical lines in the image plane.

In order to detect vertical lines we extracted the edges in the image using a modified Sobel filter proposed by [Jahne1997] that uses different coefficients to obtain a lower angle error in the gradient. By choosing an appropriate threshold for the gradient magnitude, the potential edge lines can be identified. The square of the gradient was used in our application to allow faster integer computation.

To only obtain the vertical edges we compare the pixel gradient with the vertical. The dot product of the gradient with the vertical should be null, so by setting a tolerance threshold value the detected edge points can be taken as vertical or not.

$$\boldsymbol{\mathcal{D}}.\hat{\boldsymbol{n}}_i < tolerance \tag{5.34}$$

But this can lead to erroneous results since the pixel gradient provides a very local information and is affected by the pixel quantisation, therefore a large tolerance is used. In order to extract the vertical lines in the image, all edge points that satisfied equation (5.34) were mapped to a rectified image table (equation (5.35)), so that continuity could be tested along the vertical edge direction. So each edge point $\boldsymbol{p}_j = (u, v)$ contributed to the table at position

$$vert\_points\,(x, y) = \left(\boldsymbol{p}_j.\hat{\boldsymbol{h}}_i, \boldsymbol{p}_j.\hat{\boldsymbol{n}}_i\right) \tag{5.35}$$

where $\hat{\boldsymbol{h}}_i$ is the horizontal unit vector, perpendicular to $\hat{\boldsymbol{n}}_i$ in the image plane. *i.e.*,

$$\hat{\boldsymbol{n}}_i.\hat{\boldsymbol{h}}_i = 0 \tag{5.36}$$

The minimum line length and allowable gaps is set and each column of the table parsed. The end result is a set of lines, given by their end-points in the original image. The parameters that need to be set are the gradient magnitude and angle tolerance thresholds, and the minimum line length and tolerated gap size.

For large tilt values, the vertical lines cannot be taken as parallel, and must be tested to comply with vanishing point $\hat{\boldsymbol{n}}$. If $\boldsymbol{m}$ is the unit vector normal to the line projection plane, the 3D line can only be vertical if

$$\hat{\boldsymbol{n}}.\boldsymbol{m} = 0 \tag{5.37}$$

but, as above, with a single view, a false vertical might be detected in rare degenerate cases.

We implemented this method with our system, working real time at 10 frames per second. Figure 5.6 shows an example of the results obtained. The results showed that the method performs well in man made environments where vertical lines segments are abundant, but required some parameter adjustment to have good results with different types of scenes. The gradient magnitude threshold used to identify edges is sensitive to

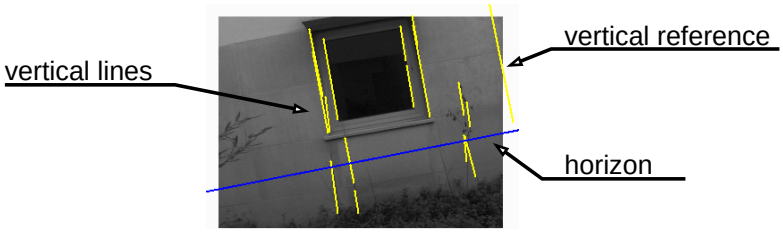


Figure 5.6: Vertical line detection.

image lighting and contrast. The line length and gap size that work well in an indoor structured environment might not work well in highly textured but less structured outdoor environments.

5.3.2 Stereo Correspondence of Vertical Lines and 3D Position

In the previous section a method was presented for vertical image line detection. But in order to have world feature detection, the image segmentation of vertical lines has to be matched across the stereo pair, and the 3D position of the feature determined.

Making the assumption that the relevant vertical features start from the ground plane, and since we know the collineation of the ground plane image points from (5.31) a common unique point is identified. The lower point or *foot* of each vertical feature in one image should map to the corresponding *foot* in the other image.

Proceeding as before, the *feet* of the vertical line features can be tested across the stereo pair using the known collineation. If there is a match, the point belongs to the ground plane and must be the *foot* of a true 3D vertical world feature. The 3D position of the *foot* of this vertical element is given by (5.33). Figure 5.7 summarises the implemented method.

With the system mounted on some mobile robot, the vertical features can be charted on a world map, constructed as the robot moves in its environment. This map is constructed in the robot's navigation frame of reference $\{\mathcal{N}\}$ as described in section 5.1.3.

Besides the error in detected points and their 3D mapping previously mentioned, the vertical edge detection also used the vertical reference, but only as a rough estimate.

Under the assumption that near vertical features are rare, this does not present a problem.

The vertical line segment detection method can produce some outliers. Before using it to update the world map, the detected feature data must be filtered to remove the outliers. A fast computational method for outlier removal was developed in complement to this work [Lobo2003JRAS]. By setting a minimum expected distance between distinct vertical features, a windowing scheme isolates each cluster and removes the outliers.

5.3.3 Results

The ground plane segmentation algorithm was implemented on the same experimental setup used for ground plane segmentation shown in figure 5.3.

We implemented the vertical world feature detector with our system, working real time at 5 frames per second. An initial setup had to be done to properly align the cameras and verge them with a known angle, using the pan and tilt units.

Figure 5.8 shows a set of results. The system was initially fixated on a ground plane point, using the pan and tilt units to verge with a known angle, so that system height could be determined. Keeping a constant height, the system was tilted sideways, and the vertical feature was correctly detected in all frames. Further tests showed that method performs well in man made environments where vertical features are abundant, but required some parameter adjustment to have good results with different types of scenes.

Using (5.33) and (5.9) the vertical features are then charted on a world map. Figure

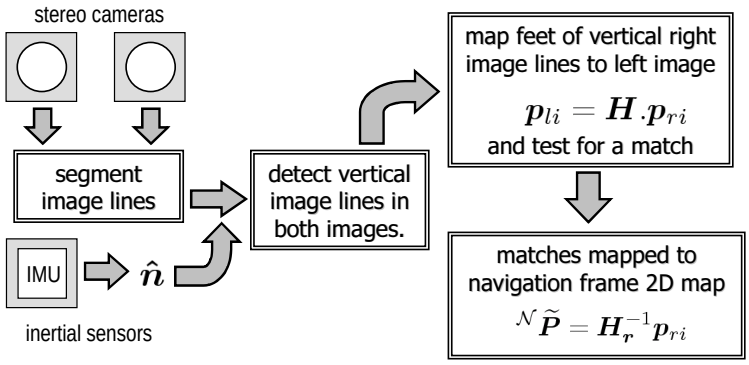


Figure 5.7: Stereo correspondence of vertical lines and 3D position algorithm.

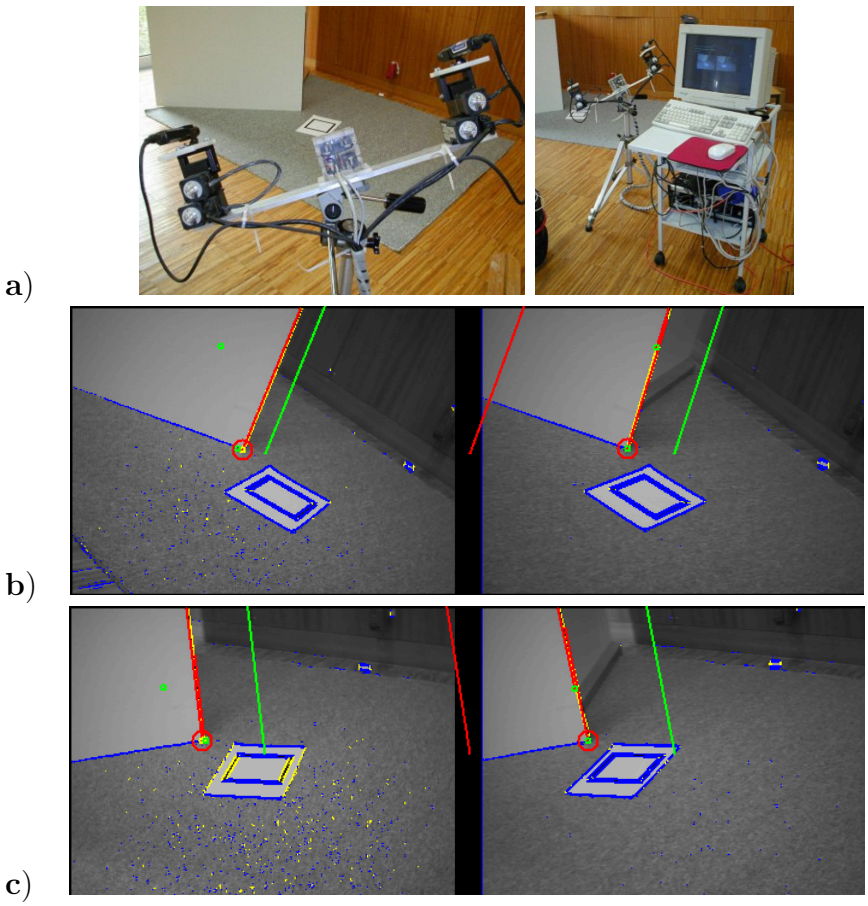


Figure 5.8: a) The experimental setup. b) and c) Vertical world feature detection. The bigger circles indicate the *foot* of a detected vertical world feature, the smaller circles the points tested, i.e., the lower end of image vertical lines.

5.9 shows the output of the vertical world feature detector that includes a map with detected features. The system was placed on a mobile robot and placed in the entry hall of our lab. The maps shows the furniture correctly mapped. The raw data shows a spread along the line of sight of the system, as expected from the geometric setup and image noise.

Proper time filtering and outlier removal has to be performed to have a consistent map. The map has than to be updated as the robot moves in its environment, as shown in this next set of results where outlier removal was performed.

An unstructured real scene was used, as seen in figure 5.10. A chair was placed in a natural environment with plants and vases. The robot was set in motion and the results are shown in figure 5.10. The lines show the ground truth chair position, the nearby circle a plant vase that occluded one of the chair legs.

The robot was placed slightly higher ground, and the detected features were very near, therefore the error spread along the X direction is less than for other tests where the geometric dilution of precision in the 3D triangulation was greater.

Part of this work was presented at [Lobo2001SIRS] and [Lobo2003JRAS].

5.4 Stereo Depth Map Alignment and Ground Segmentation

5.4.1 Rotating Depth Maps

Stereo vision systems can use correlation based methods to obtain depth maps. With the current technology, real time systems are commercially available. When the vision system is moving the maps have to be fused into single world map. Before fusing the depth maps, they must be registered to a common referential. This can be done using data fitting alone, or aided by known parameters or restrictions on the way the measurements were made.

Figure 5.11 shows the frames of reference that we need to consider. The depth maps are given in the stereo camera frame of reference, in out case we take the left camera as the reference camera, with the Z axis pointing forward along the optical axis. The depth

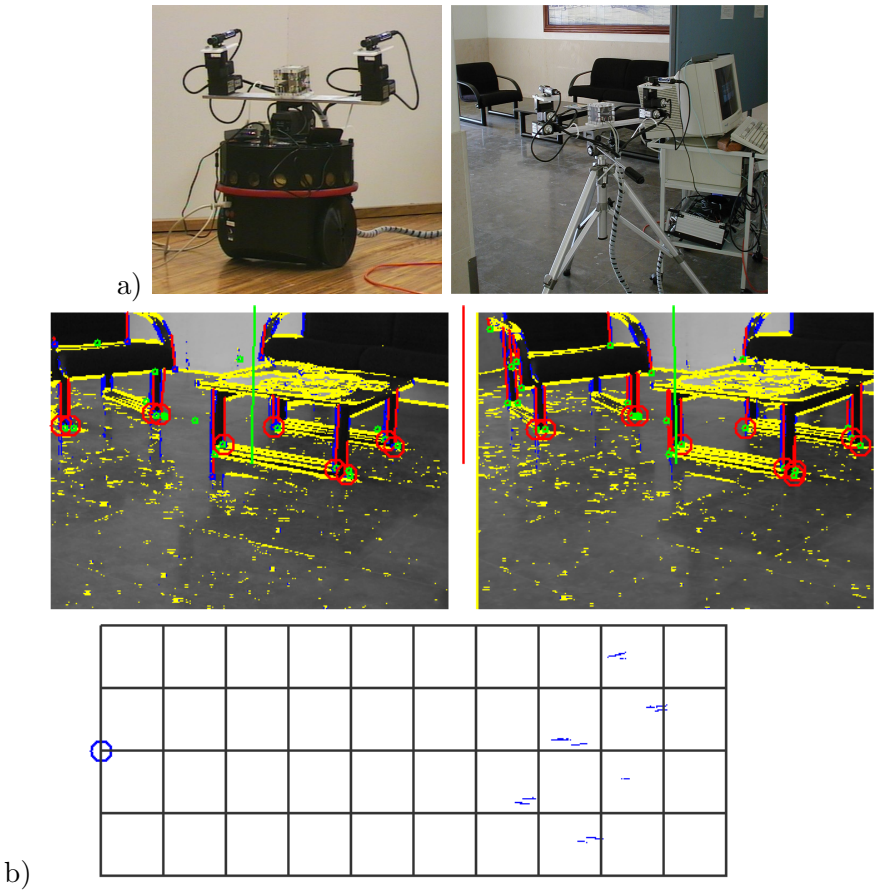


Figure 5.9: a) The experimental setup, with the system placed on a mobile robot and placed at the entry hall of our lab. b) Vertical world feature detection. The circle in the map represents the robot, and the points the detected vertical world features

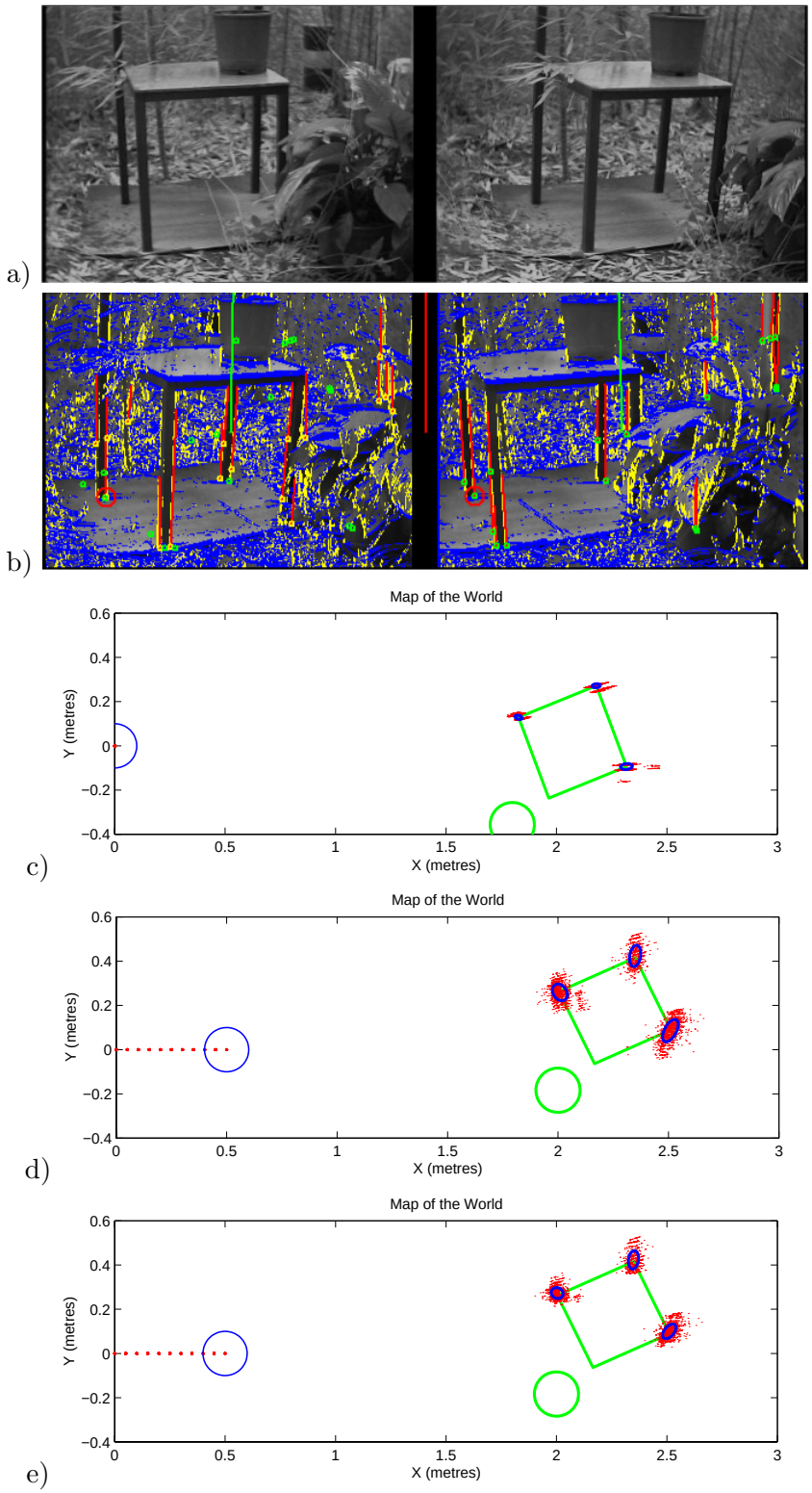


Figure 5.10: a) Stereo images. b) Detected edges and vertical lines. c) World map with robot not moving. d) World map with robot moving. e) World map with robot moving and outlier removal.

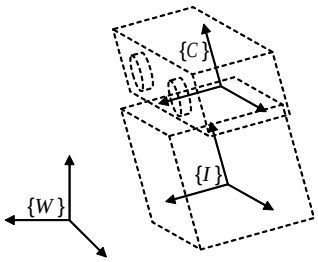


Figure 5.11: Frames of reference for stereo vision system with inertial measurement unit.

map is given by a pencil of rays with known depth from the origin, in this case the left camera optical center.

Using the stereo depth algorithm we obtain a set of points ${}^{\mathcal{C}}P_i$ in the left camera referential. Using the previous equations from section 5.1.3 we can map them to the navigation frame of reference as

$$ {}^{\mathcal{N}}P_i = {}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}P_i \tag{5.38} $$

But from (5.10) ${}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}}$ requires knowing the system height relative to the ground plane. We can only rotate so that the depth map will be aligned with the horizontal plane, i.e., rotate with ${}^{\mathcal{N}_C}\boldsymbol{T}_{\mathcal{C}}$ given by (5.18)

$$ {}^{\mathcal{N}_C}P_i = {}^{\mathcal{N}_C}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}P_i \tag{5.39} $$

5.4.2 Aligning to the Ground Plane

In these rotated depth maps, planar levelled patches will have the same depth z, so in order to detect the ground plane height, an histogram is performed for point depth.

$$ hist_z(n) = \sum (P_i \mid floor(z_{P_i}) = n) \tag{5.40} $$

The histogram's lower local peak z_{gnd} is used as the reference depth for the ground plane. The depth maps are than all translated and aligned with this reference ground plane, with only a rotation about a vertical axis and a 2D translation remaining for full registration.

5.4.3 Segmenting the Depth Map

The detected points can than be parsed and segmented as being a ground plane point, or some feature above ground. Points bellow the ground plane can be ignored or not, depending on the application.

$$P_{gnd} = P_i \mid z_{gnd} - \delta \leq floor(z_{P_i}) \leq z_{gnd} + \delta \tag{5.41}$$

$$P_{above} = P_i \mid floor(z_{P_i}) \geq z_{gnd} + \delta \tag{5.42}$$

were δ is the allowed tolerance. The points above ground can be projected in the XY plane, and further segmentation performed to identify vertical features.

5.4.4 Summary of Method for Stereo Depth Map Alignment and Ground Segmentation

Using the vertical reference, the depth maps can be segmented to identify horizontal and vertical features. The aim is on having a simple algorithm suitable for a real-time implementation. Since we are able to map the points to an inertial reference frame, planar levelled patches will have the same depth z, and vertical features the same xy, allowing simple feature segmentation using histogram local peak detection. Fig. 5.12 summarises the proposed depth map segmentation method.

5.4.5 Results

In order to obtain depth maps with known vision system pose, the stereo vision system was mounted with an inertial measurement unit, as shown in fig. 5.13. To compute range from stereo images we are using the SRI Stereo Engine [Konolige1997] with the Small Vision System (SVS) from [Videre].

A simple indoor scene was used to test our method. The stereo pair seen in figure 5.14 was obtained with the experimental setup shown in figure 5.13. Figure 5.15 shows the disparity image and reconstructed 3D points obtained with the SVS package.

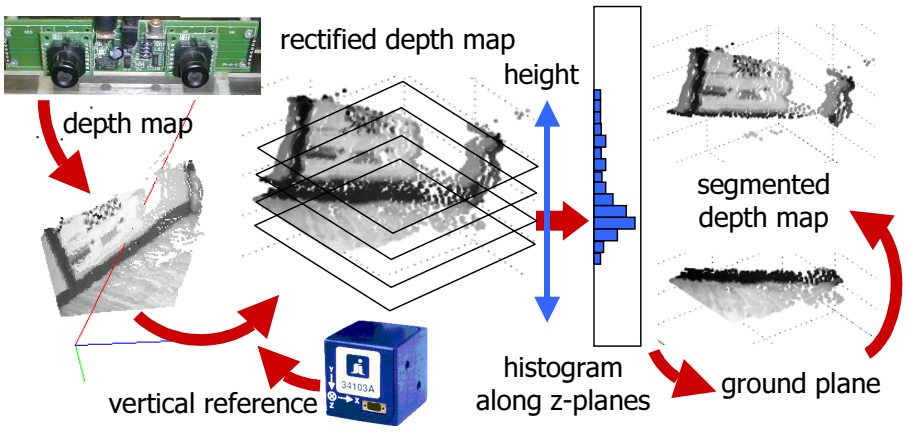


Figure 5.12: Summary of implemented method.

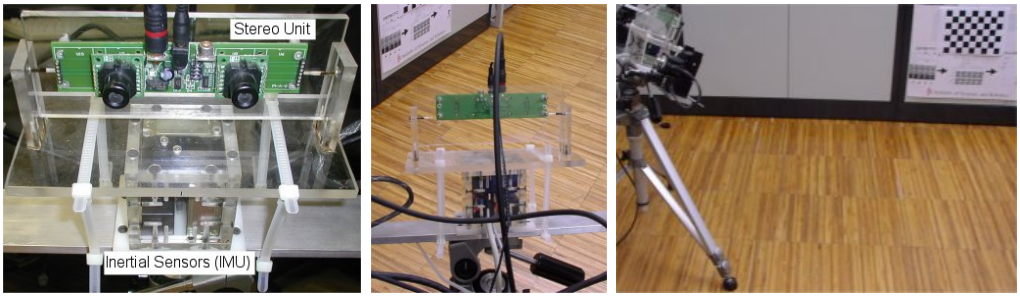


Figure 5.13: Experimental setup with inertial sensors and vision system, and scene used for the test.

Figure 5.14: Stereo rectified image pair obtained with SVS system.

Using the vertical reference provided by the inertial sensors the 3D points were transformed to a world aligned frame of reference as previously described.

In order to detect the ground plane, an histogram was done for all depths, and the peak used as a reference value, as seen in figure 5.16. The points were than parsed and segmented as ground plane points and points above ground.

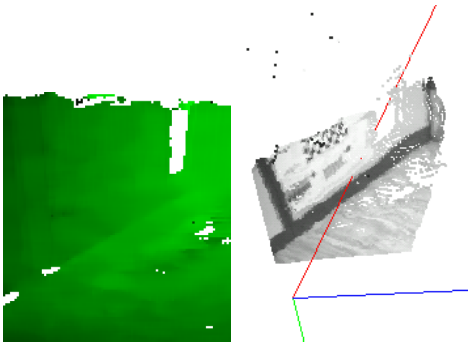


Figure 5.15: Disparity image obtained with SVS, and reconstructed 3D points

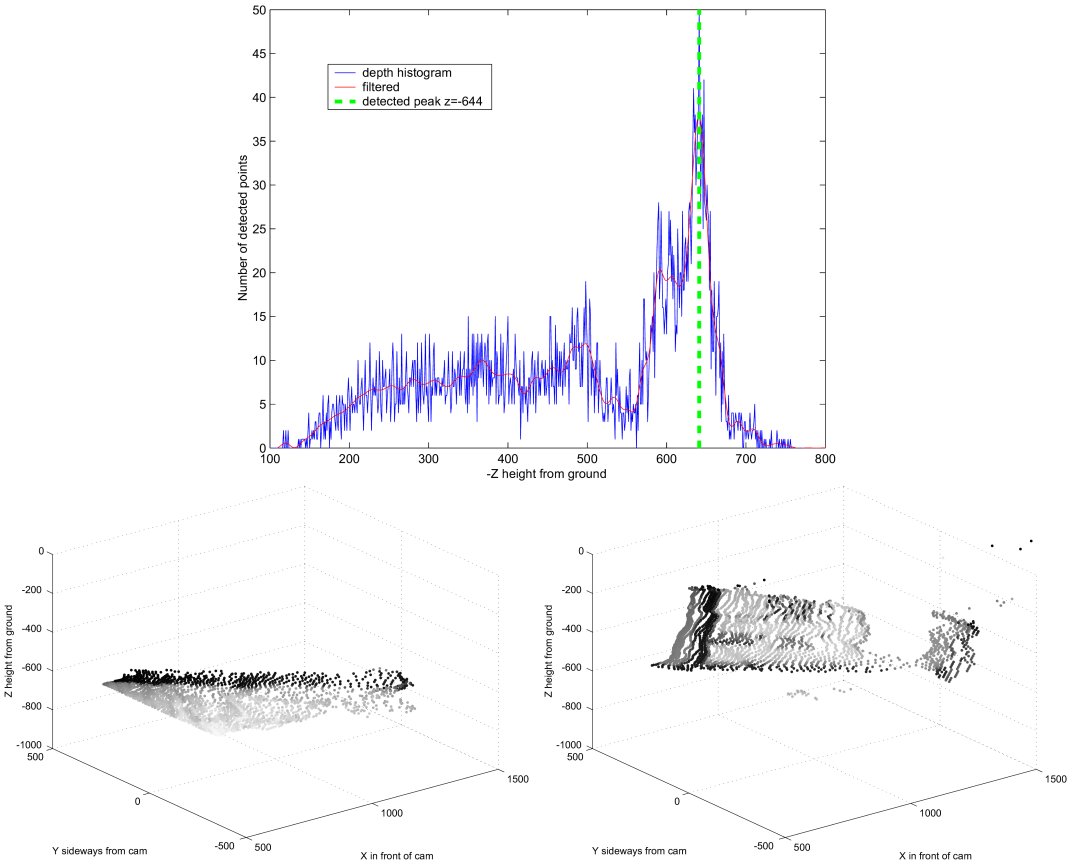


Figure 5.16: Depth histogram with detected peak (top); ground plane points (right); points above the floor, i.e., walls or obstacles (left).

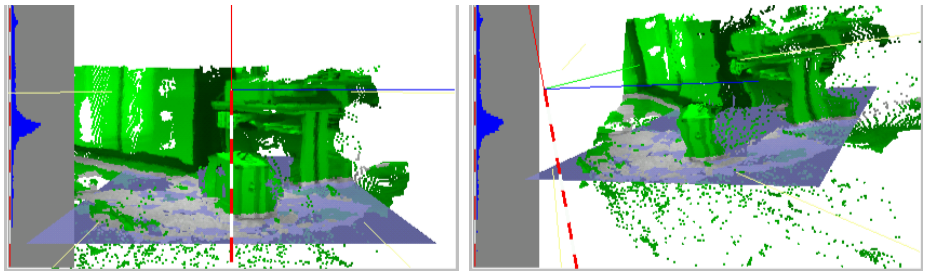


Figure 5.17: Graphical front-end with height histogram and segmented depth map.

Figure 5.17 shows the graphical front-end of the implemented system working realtime at 10 frames per second. On the left the height histogram is shown.

A linear line fit was done using the points above ground from the data set in figure 5.16, ignoring their depth, to reconstruct the wall orientation in the test scene. Figure 5.18 shows the result. More complex scenes require a previous point clustering stage, so that a simplified world model can be built, but this only has to be done in 2D.

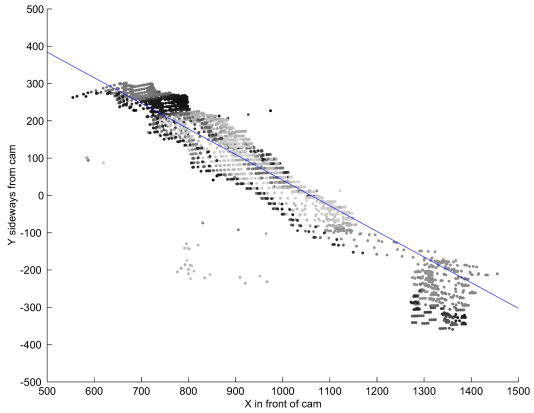


Figure 5.18: Top view of all points above the floor, and line fit for wall orientation.

With the vision system moving, the acquired depth maps have to be registered to a common frame of reference. After the alignment using the vertical reference and subsequent ground plane detection, the registration is a 2D problem, only a translation (t_x, t_y) and rotation θ are needed, see fig. 5.19.

An approximation to these 2D parameters can be found by projecting the inertial sensed parameters onto the level plane. These allows registering dynamic depth maps,

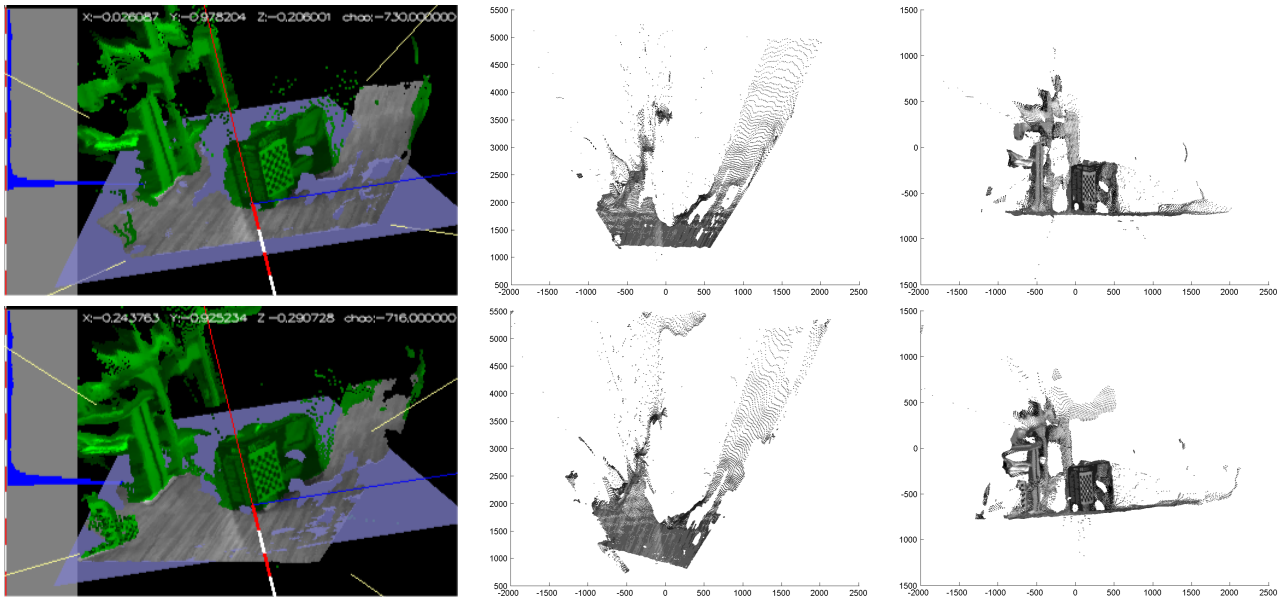


Figure 5.19: On the right the graphical front-end of implemented system, showing the height histogram for ground plane detection, the detected plane and 3D segmented depth map; on the right the top and front view of the aligned segmented depth maps, that only require a translation (t_x, t_y) and rotation θ to be correctly fused.

with moving objects, to a common frame of reference.

Real time depth map computation, rotation update, ground plane detection and real-time 3D rendering of the rotated depth maps is done currently at 10 fps, with the above described hardware and a Pentium IV at 1.5 GHz.

5.5 Discussion and Conclusions

In this chapter we focused on the use of the inertial vertical reference in vision systems.

In a stereo rig with known geometry, the vertical reference was used to compute the collineation of level plane points, enabling their detection and 3D mapping. This was used to segment and reconstruct vertical features and levelled planar patches. These 3D world features are useful to improve mobile robot autonomy and navigation. The method is fast and adaptable, unlike a fixed calibrated collineation estimated from a set of known points. The main sources of error in this method are the assumed known geometry, and

the noise in the vertical reference. When used on a mobile vehicle the error increased along the direction of motion, but still provided a useful map of vertical features for robot navigation, where the uncertainty can be modelled.

Another approach we followed was to use standard vision techniques to compute depth maps, and than rotate and align them using the inertial reference. The advantage of reducing the search space explored above is lost, but current technology provides real-time depth maps with reasonable quality, and the inertial data fusion is still very useful at a later step to align and register the obtained maps.

Results were shown of stereo depth map alignment using the vertical reference. The depth map points are mapped to the a vertically aligned world frame of reference. In order to detect the ground plane, an histogram is performed for the different heights. Taking the ground plane as a reference plane for the acquired maps, the fusion of multiple maps reduces to a 2D translation and rotation problem. The dynamic inertial cues can be used as a first approximation for this transformation, allowing a fast depth map registration method.

The aim of this work is a fast real-time system, avoiding 3D point clustering methods that are not suitable for real-time implementations. It can be applied to an automated car driving system, modelling the road, identifying obstacles and roadside features.

In the next chapter we will address the fusion of optical flow computation with the stereo data to accomplish independent motion segmentation. To fully register the depth maps, magnetic sensors are used to complement the data from the accelerometers and provide an external reference for 3D rotation, and image features used to compute translation.

Chapter 6

3D Map Registration and Independent Motion Segmentation

In vision based systems used in mobile robotics the perception of self-motion and structure of the environment is essential. In the previous chapter results were presented using the inertial vertical reference alone.

Combining inertial and earth field magnetic sensors we can have an external rotation reference, providing valuable data about camera ego-motion, as well as absolute references for structure feature orientations. The vertical reference is disturbed by body acceleration and the magnetic bearing more severely by electro magnetic disturbances and proximity to ferrous metals, but the derived rotation update can be used in some applications where the cost and size of high grade gyros for full INS computations is unsuitable.

In this chapter we explore the fusion of optical flow and stereo techniques with data from the inertial and magnetic sensors, enabling the depth flow segmentation of a moving robotic observer to accomplish independent motion segmentation.

A depth map registration and motion segmentation method is proposed, and experimental results of independent motion segmentation for a moving observer are presented.

6.1 Introduction

Stereo vision systems can use correlation based methods to obtain depth maps. With the current technology, real time systems are commercially available. When the vision system is moving the maps have to be fused into a single world map. Before fusing the depth maps, they must be registered to a common referential. This can be done using data fitting alone, or aided by known parameters or restrictions on the way the measurements were made.

In our work, correlation based stereo depth maps are obtained from a moving vision system, and rotated to a common levelled reference provided by the rotation update from inertial sensed gravity and magnetic sensed bearing. Voxel quantisation can then be performed on the resulting maps.

But there remains a 3D translation in the successive depth maps due to the motion, for which the inertial sensors only provide a rough estimate. By tracking some image targets over successive frames, the system translation between frames can be estimated by subtracting their 3D position.

The translation can also be estimated from the 3D data alone. For scenes where a base horizontal plane is always visible (e.g.: the floor or desktop), a histogram in height can be used to have a common reference along the vertical axis. This can also be performed for the horizontal axis if the orientation of visible planes is known or detected by a 2D fit to the data. The two identified planes provide the translation to merge successive depth maps.

Fully registered depth maps can therefore be obtained from the moving system. The depth flow that remains in the resultant map is due to the system covering new scenes, or to moving objects within the overlap volume of successive observations. Mismatches between the depth from stereo and depth from optical flow indicate possible independent motion. This can be used to better segment moving objects in the overlap volume and avoid artifacts from slow moving objects.

6.1.1 Related Work

Three-dimensional scene flow estimation was studied by Vedula et al. [Vedula2005] [Vedula1999]. Several scenarios are presented, and the tradeoffs between structure knowledge, correspondence matching, number of cameras and computed optical flow explored. Dense scene flow estimation using only two cameras was proposed by Li and Sclaroff by fusing stereo and optical flow estimation in a single coherent framework [Li2005]. Ye Zhang and Kambhamettu computed dense 3D scene flow and structure from multiview image sequences with nonrigid motion in the scene [Zhang2003]. Stereoscopic MPEG based video compression methods also deal with motion flow segmentation, such as the joint motion and disparity fields estimation method proposed by Yang et al. [Yang2005]. A statistical approach to background modelling was used for segmentation of video-rate stereo sequences by Eveland et al. [Eveland1998].

Our approach deals with a free moving stereo camera observer, for which the above methods are not directly applicable. Inertial sensors provide valuable data to deal with the camera motion [Lobo2004JRS]. Visual and inertial sensing are two sensory modalities that can be explored to give robust solutions on image segmentation and recovery of 3D structure from images [Lobo2003PAMI].

6.2 Registering Stereo Depth Maps

A moving stereo observer of a background static scene with some moving objects can compute at each instant a correlation-based dense depth map. The maps will change in time due to both the moving objects and the observer ego-motion. A first step to process the incoming data is to register the maps to a common fixed frame of reference $\{\mathcal{W}\}$, as shown on Figure 6.1.

The stereo cameras provide intensity images $I_l(u,v)|_i$ and $I_r(u,v)|_i$, where u and v are pixel coordinates, and i the frame time index. Having the stereo rig calibrated, depth maps for each frame can be computed. A set of 3D points ${}^{\mathcal{C}}\mathbb{P}|_i$ is therefore obtained at each frame, given in the camera frame of reference $\{\mathcal{C}\}|_i$. Each 3D point has a corresponding intensity gray level c given by the pixel in the reference camera, i.e., $c = I_l(u,v)|_i$. Each

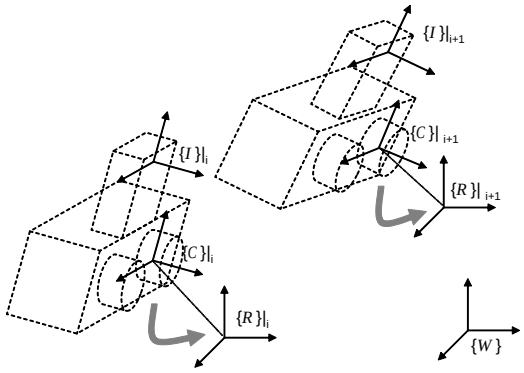


Figure 6.1: Moving observer and world fixed frames of reference.

point in the set retains both 3D position and gray level

$$(P)(x, y, z, c) \in {}^{\mathcal{C}}\mathbb{P}|_i \ . \tag{6.1}$$

6.2.1 Rotate to Local Vertical and Magnetic North

In chapter 5 we used the inertial vertical reference to rotate stereo depth maps to a levelled frame of reference. However there remained a rotation about a vertical axis for which gravity provides no cues. The earth's magnetic field can be used to provide the missing bearing [Caruso1998], however the magnetic sensing is sensitive to the nearby ferrous metals and electric currents. In fact, there is some overlap and complementarity between the two sensors, with different noise characteristics that can be exploited to provide a useful rotation update [Roetenberg2003] [Roetenberg2005].

The inertial and magnetic sensors, rigidly fixed to the stereo camera rig, provide a stable camera rotation update ${}^{\mathcal{R}}\boldsymbol{R}_{\mathcal{C}}$ relative to the local gravity vertical and magnetic north camera frame of reference $\{\mathcal{R}\}|_i$.

Calibration of the rigid body rotation between $\{\mathcal{I}\}|_i$ and $\{\mathcal{C}\}|_i$ can be performed by having both sensors observing gravity, as vertical vanishing points and sensed acceleration, as described in chapter 4.

The rotated camera frame of reference $\{\mathcal{R}\}|_i$ is time-dependent only due to the camera system translation, since rotation has been compensated for.

6.2.2 Translation from Image Tracked Target

The translation component can be obtained using a single fixed target tracked in the scene. The image feature must have the corresponding 3D point $\boldsymbol{P}_t$ in each depth map, so that translation can be estimated from

$$\Delta\vec{t} = \boldsymbol{P}_t|_{i+1} - \boldsymbol{P}_t|_i \tag{6.2}$$

with $\boldsymbol{P}_t|_{i+1} \in {}^{\mathcal{R}}\mathbb{P}|_{i+1}$ and $\boldsymbol{P}_t|_i \in {}^{\mathcal{R}}\mathbb{P}|_i$.

The fixed target can be an artificial one, or set of sparse tracked natural 3D features can be used to improve robustness, but assumptions have to be made in order to reject outliers that occur from tracking features of the moving objects.

6.2.3 Voxel Quantisation

The above equations are provided for discrete sets of points. In order to deal with noise and allow 3D volume processing, a 3D array is built representing 3D space as voxels. For each stereo frame, the corresponding cubic array of voxels $Vox|_i$ can be built. For the occupied voxels the corresponding gray level can be stored in the array. When two or more points contribute to the same voxel, the average gray level is used.

For each $\boldsymbol{P}(x, y, z, c) \in {}^{\mathcal{C}}\mathbb{P}|_i$, $Vox(x, y, z)|_i = c$ if previously empty, or $Vox(x, y, z)|_i = \bar{c}$, where $\bar{c}$ is the average gray level of the contributing points.

For a sequence of stereo frames, two cumulative voxel arrays Vox_c and Vox_v can be built for both gray level and occupancy statistics over the frames, with

$$Vox_c(x, y, z) = \bar{c}_v, \qquad Vox_v(x, y, z) = v \tag{6.3}$$

where v is the number of frames that voted voxel (x, y, z) as occupied, and $\bar{c}_v$ the average gray level from the voting frames.

6.2.4 Summary of Stereo Depth Maps Registration Method

Figure 6.2 summarises the proposed stereo depth map registration method using inertial and magnetics sensors for rotation update and image features for translation.

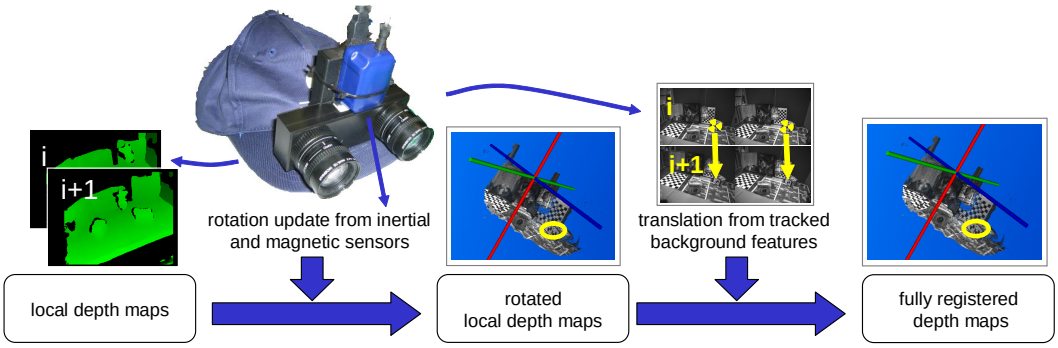


Figure 6.2: Summary of stereo depth map registration method.

6.3 Independent Motion Segmentation in Fully Registered Maps

Having the dense depth maps in a common frame of reference we can proceed to segment the moving objects seen by the moving stereo observer. Biological vision systems are very successful in movement segmentation since they efficiently resort to flow analysis and accumulated prior knowledge of the 3D structure of the scene. Artificial perception systems may also build 3D structure maps and use optical flow to provide cues for ego and independent motion segmentation. The maps will change in time due to moving objects, and eventually grow as the artificial observer covers new scene areas.

6.3.1 Background Subtraction for Voxel Segmentation

Occupancy statistics can be used to segment the set of voxels that correspond to the static scene observed by the moving system, and segment the moving objects.

Applying a threshold v_{back} on the accumulated vote count, a binary array of background voxels Vox_b can be built as

$$Vox_b(x, y, z) = 1 \text{ when } Vox_v(x, y, z) > v_{back} \ . \tag{6.4}$$

To improve noise filtering and robustness, a thinning and growing transformation is applied, removing isolated voxels and filling in gaps. The thinning filter takes out voxels

without a minimum number of neighbours, by performing a convolution with a cubic unit kernel and thresholding the result back to a binary array. The growing simply performs a convolution with a cubic unit kernel, and rebuilds the binary array with all the non-zero voxels.

For a single frame i, the set of voxels from moving objects will be given by

$$Vox_m|_i = Vox|_i \bigcap \overline{Vox_b} \ . \tag{6.5}$$

To deal with noise, thinning and growth smoothing can also be applied to $Vox_m|_i$, but smearing of the intensity gray level might not help subsequent 3D intensity based methods.

The underlying assumption is that the moving observer repeatedly covers the same scene so that background voxels are seen more times than moving objects. Experimental results show that moving objects are successfully segmented and that thinning and growth smoothing filter out noise from the correlation based stereo depth maps.

6.3.2 Optical Flow Consistency Segmentation

In section 3.2.4 we saw that optical flow is the apparent motion of brightness patterns in the image, and how it can be computed from an image sequence. When the camera is moving and observing a static scene with some moving objects, some optical flow will be consistent with the camera ego-motion observing the static scene, other might be moving objects. Since the stereo provides a dense depth map, and we reconstruct camera motion, we can compute the expected projected optical flow in the image from the 3D data.

In the perspective camera model, the relationship between a 3D world point $\boldsymbol{x} = (X, Y, Z)^\mathsf{T}$ and its projection $\boldsymbol{u} = (u, v)^\mathsf{T}$ in the 2D image plane is given by

$$u = \frac{\boldsymbol{P}_1 (x, y, z, 1)^\mathsf{T}}{\boldsymbol{P}_3 (x, y, z, 1)^\mathsf{T}} \qquad v = \frac{\boldsymbol{P}_2 (x, y, z, 1)^\mathsf{T}}{\boldsymbol{P}_3 (x, y, z, 1)^\mathsf{T}} \tag{6.6}$$

where matrix $\boldsymbol{P}_j$ is the jth row of the camera projection matrix $\boldsymbol{P}$.

When the camera moves, the relative motion of the 3D point $\frac{d\boldsymbol{x}}{dt}$ will induce a projected optical flow given by

$$\frac{d\boldsymbol{u}_i}{dt} = \frac{\delta \boldsymbol{u}_i}{\delta \boldsymbol{x}} \frac{d\boldsymbol{x}}{dt} \tag{6.7}$$

where $\frac{\delta \boldsymbol{u}_i}{\delta \boldsymbol{x}}$ is the 2×3 Jacobian matrix that represents the differential relationship between $\boldsymbol{x}$ and $\boldsymbol{u}_i$, which can be obtained by differentiating (6.6).

Image areas where the computed flow is inconsistent with the expected one indicate moving objects, and the corresponding voxels can be segmented. This approach does not require the occupancy statistics memory, since it is differential and can be applied to pairs of successive frames.

Experimental results show that this method works on sequences with significant optical flow. However, this procedure is noise sensitive and, due to its differential based estimation, it performs poorly at low speeds, where the uncertainties in camera motion and optical flow are higher.

6.3.3 Summary of Independent Motion Segmentation Methods

A summarising diagram of the procedures for both independent motion segmentation methods studied in this work is presented in figure 6.3.

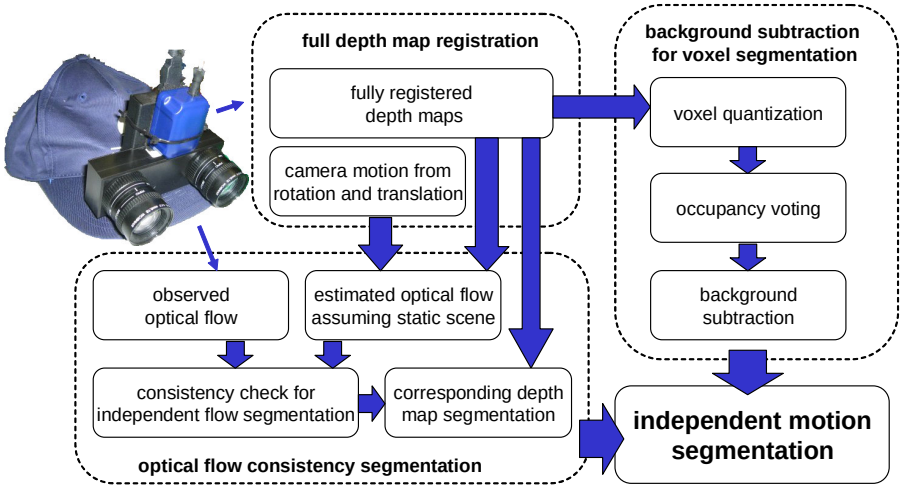


Figure 6.3: Summary of voxel background subtraction and optical flow consistency methods for independent motion segmentation.

6.4 Results

The hardware system used to acquire data from a moving observer is shown in fig. 6.4. The stereo vision is provided by the Videre MEGA-D Digital Stereo Head [Videre], and the pose from the inertial and magnetic sensor package MT9-B from Xsens [Xsens]. To compute range from stereo images we are using the SRI Stereo Engine with the Small Vision System (SVS) Software [Konolige1997].

Figure 6.4: Stereo vision system with inertial and magnetic sensors. In the middle the 3D scene with static background and swinging pendulum observed by the hand held system. The system was also mounted on a hat so that a human could perform the observation motion.

A scene was set up with a swinging cylindrical can to provide motion independent from the observer movement, as shown on Figure 6.4. The moving observer surveyed the scene performing map registration and subsequent independent motion segmentation as described below.

6.4.1 Moving Depth Map Registration

As described above, the rotation update provided by the inertial and magnetic sensor package is applied to the successive depth maps. As shown in figure 6.5b, the depth maps are correctly rotated, but shifted due to the observer translation.

The translation was estimated by tracking an image feature, and observing the translation between the corresponding 3D points in the depth maps. Figure 6.5 shows data for frames 1 and 20 of a take of 200 frames with a moving observer of a static scene with a moving pendulum.

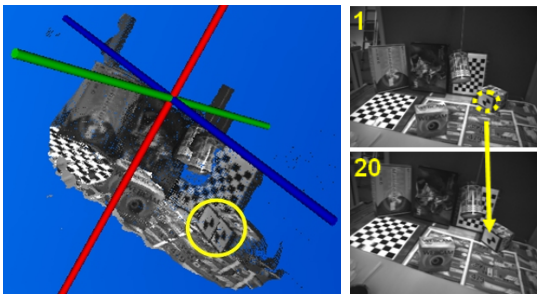


Figure 6.5: Overlaid rotated 3D depth maps from frames 1 and 20 (on the right) showing a clear mismatch, and circled image feature tracked to estimate translation.

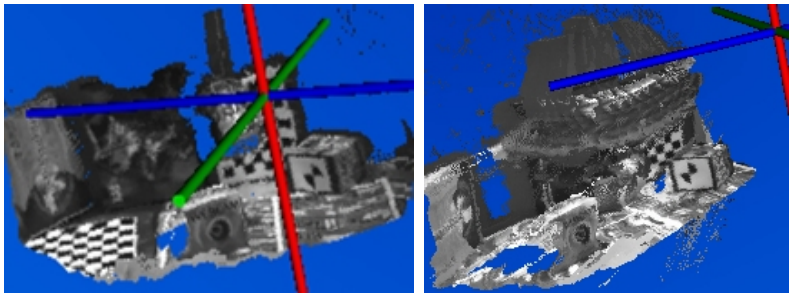


Figure 6.6: Depth maps rotated and translated to common world fixed frame of reference, for frames 1 and 20 on the left, and for full set of frames with moving pendulum on the right.

The registered depth map can be seen in Figure 6.6. The fused map from frames 1 and 20 is shown on the left. On the right the fused map corresponding to the full set of frames is shown with the moving pendulum leaving its trace.

The registration performed well, since the background shows no mismatch and the moving pendulum clearly leaves its trace.

The registered depth maps can than be quantised into voxels and contribute to voxel set Vox_v, that accumulates the number of frames that vote the voxel as occupied, and Vox_c, the average gray level from the voting frame, for subsequent computations.

The registered depth maps can than be quantised into voxels and contribute to voxel sets Vox_v and Vox_c for subsequent computations. Vox_v accumulates the number of frames that vote the voxel as occupied, and Vox_c the average gray level from the voting frames. Figure 6.7 shows a rendered view of the voxel space for a single frame.

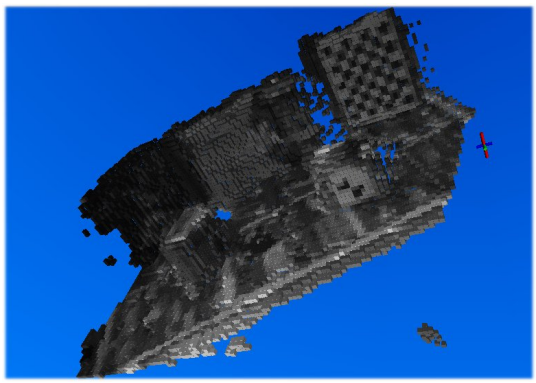


Figure 6.7: Vox Cubes.

6.4.2 Independent Motion Segmentation in Fully Registered Maps

Background Subtraction for Voxel Segmentation

The above results are shown with VRML rendering of the full set of computed points without voxel quantisation. As described above, occupancy statistics can be used to identify the static scene voxels. In a new test sequence, a one cubic meter volume of the observed space was chosen as the working volume, quantised to a $100 \times 100 \times 100$ array corresponding to $1\,cm^3$ voxels.

Figure 6.8a shows the 3D volume of all accumulated voxels for this test sequence with 130 frames, and 6.8b the ones with a vote count above the empirically chosen threshold of 30. This choice was made based on the following observations: very low thresholds will mark slow objects as background; too high will segment newly observed static background as moving objects. Frame rate, observer motion and independent motion velocities are determining factors when choosing appropriate thresholds. The result of thinning and growth filters applied to the background voxels is shown in figure 6.8c and 6.8d.

Figure 6.9 shows initial voxel set and segmented moving object voxels after thinning and subsequent smoothing, for one frame from the above sequence.

The results show that moving object voxels can be successfully segmented, however the moving observer has to cover the same scene more than once so that the background voxels can be correctly segmented.

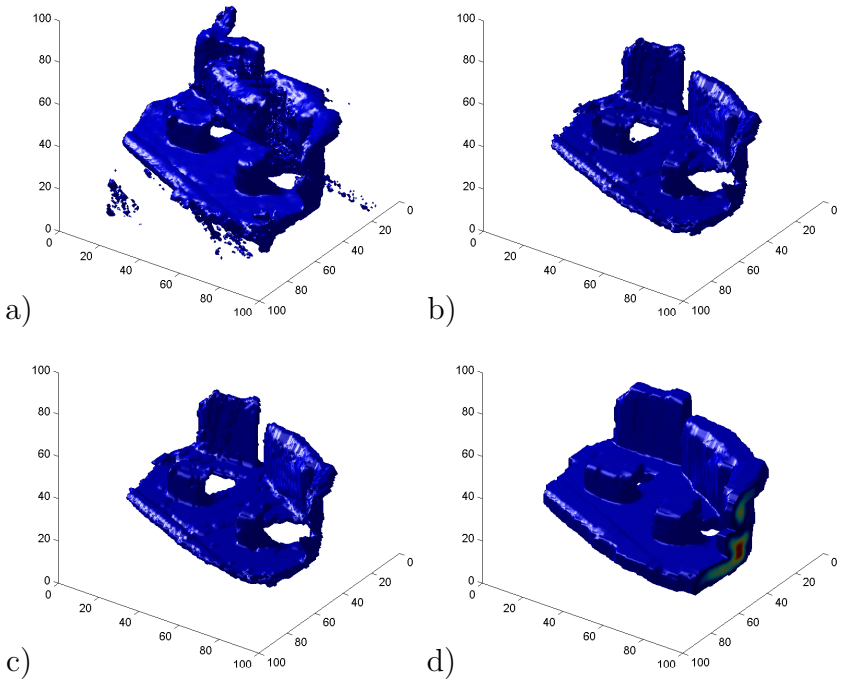


Figure 6.8: Obtaining the background voxels: a) 3D volume of all accumulated voxels in frame sequence, b) with vote count above 30, c) after thinning for at least 6 neighbours, d) and subsequent growth with a size 5^3 kernel.

Optical Flow Consistency Segmentation

Figure 6.10 shows the optical flow at one frame within a test sequence where the pendulum and observer were both moving. The image observed optical flow was estimated with the Lucas-Kanade [Lucas1981] method applied to successive frames. The predicted flow was computed considering the 3D motion of the depth map relative to the camera, and projecting onto the image using (6.7).

The difference between the observed optical flow and the predicted flow indicate areas inconsistent with a static scene. The voxels associated with these image pixels correspond to moving objects. A decision threshold is applied to the optical flow difference to segment the voxels. The result for this frame is shown in figure 6.11.

The method works, but is clearly sensitive to noise due to the differential based estimation. In the complete test sequence there, frames with small camera motion performed

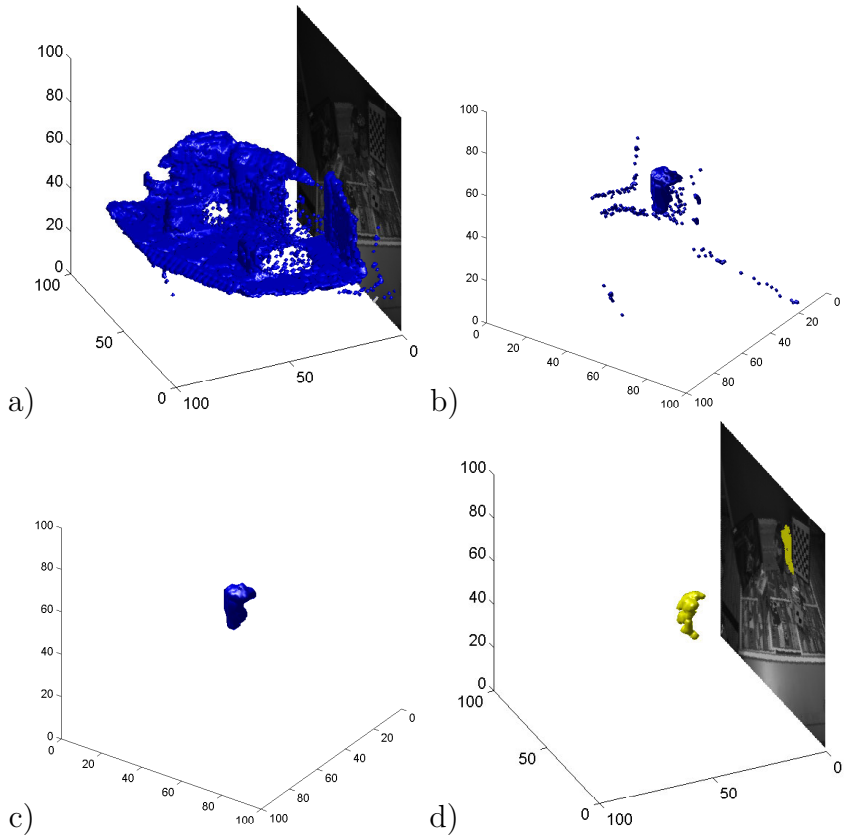


Figure 6.9: Background subtraction for voxel segmentation: a) initial voxel set for one frame, b) after background voxels subtracted, c) segmented moving object voxels after thinning for at least 6 neighbours and subsequent smoothing, d) segmented moving object voxels from a different view and mapped onto the image frame.

poorly, since the uncertainties in camera motion and optical flow computation degrade de detection of moving objects.

6.5 Conclusions

Two methods were presented for motion segmentation for a moving observer of a background static scene with some independently moving objects. The moving observer has stereo vision and, to provide a rotation update, inertial and magnetic sensors. Having compensated rotation, translation can be obtained from a single tracked image feature.

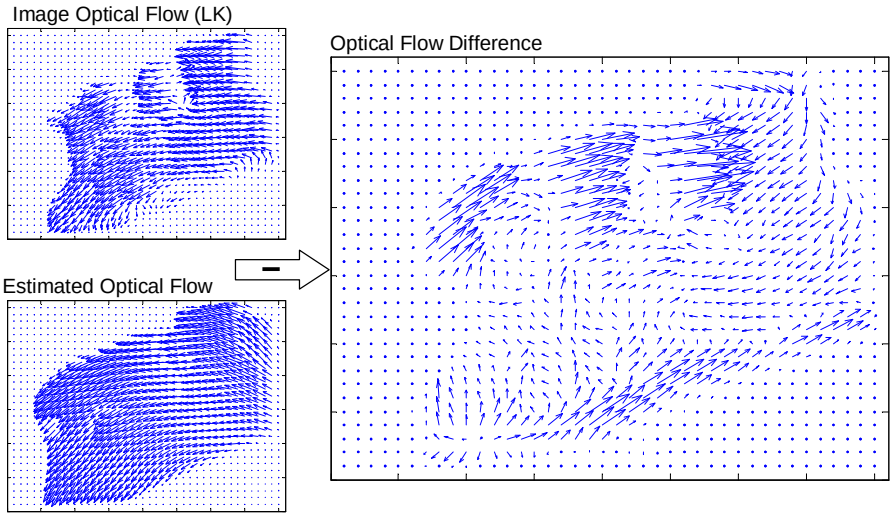


Figure 6.10: Difference between observed optical flow computed between successive frames using Lucas-Kanade image based estimation, and estimated optical flow using just the 3D data from the first frame and camera motion, indicating areas inconsistent with static scene after camera motion compensation.

Depth maps from stereo vision can therefore be registered to a common frame of reference.

Occupancy statistics can then be used to segment the voxels between the static background scene and moving objects. However, the moving observer has to cover the same scene so that the background voxels can be correctly segmented. An alternative method is to check the consistency of the observed image optical flow. This approach is differential and can be applied to pairs of successive frames, but is more noise sensitive.

The results shown were processed with Matlab in batch mode, ongoing work is being done to have the system running online.

Future work will address the use of inertial dynamic data to improve the optical flow consistency check, without depending on any tracked feature for the translation, and on combining the two methods to improve robustness.

Figure 6.12 shows the output of the two methods for the same frame. The voxel background subtraction correctly identifies the independent motion. Due to this fact, in this work it also provided a ground truth to compare the optical flow consistency method. The optical flow consistency method also segments the independent motion, but with

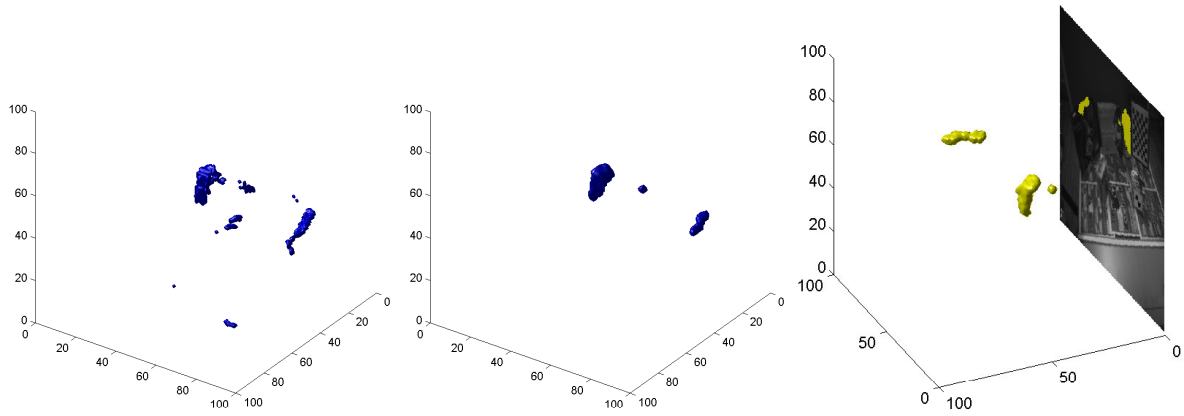


Figure 6.11: 3D voxels from moving object, based on difference between observed and expected optical flow assuming a static scene, and on the right segmented moving object voxels after thinning for at least 6 neighbours and subsequent smoothing, from two different views and mapped onto the image frame.

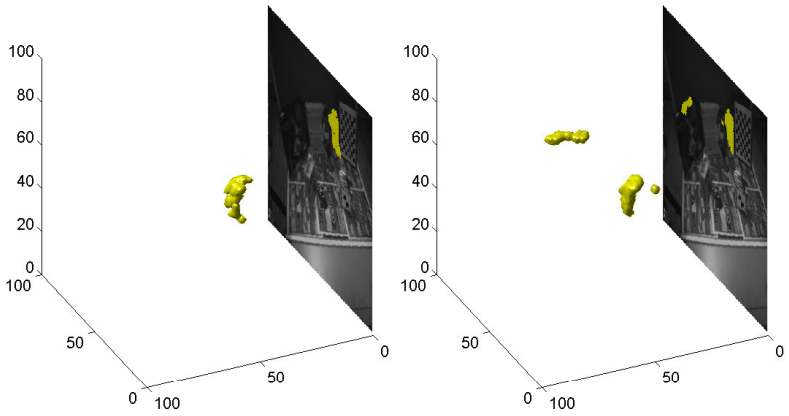


Figure 6.12: Output of the two methods for the same frame, voxel background subtraction on the left, and optical flow consistency on the right.

added false positives due to uncertainties in the optical flow computation and camera motion reconstruction.

On the other hand, voxel background subtraction requires a volumetric representation of the whole workspace, and also some past history statistics, which introduces a start-up lag of at least 10 frames, whereas optical flow consistency only needs the present and immediately preceding frames to function. Therefore, the latter can be improved on by retaining a short-term memory of 3D space occupancy, since the inertial data allows a

fast depth map registration, resulting in a hybrid method that combines the differential approach with occupancy statistics.

Chapter 7

Discussion and Future Work

This work sets a framework for inertial and visual sensor cooperation. From the studies on human vision it is clear that inertial cues play an important role in visual perception.

The unit sphere projection model used provides an intuitive representation of projective geometry, onto which inertial cues are easily integrated.

We explored the camera and inertial data relationship and proposed calibration methods suitable for the needs of robotics applications.

The inertial vertical reference provided by gravity enabled the detection and segmentation of planar and vertical 3D features and the simplification of stereo depth maps registration for moving observers.

Independent motion segmentation was accomplished by combining inertial and magnetic sensors to provide a rotation update and using image features, optical flow and stereo depth maps derived from the images.

We have tried to cover all aspects of the fusion, from a theoretical point of view, and shown practical applications. Although the implementations can be better engineered to improve performance, they prove their suitability in solving or simplifying some perceptual tasks for robotic applications. In some cases the edge given by the inertial sensors allows known computer vision techniques like simultaneous structure and motion to be applied with less demand on workable image features, as we presented in our results for depth maps registration used for independent motion segmentation.

Future work will address better modelling of sensor fusion under a Bayesian framework.

Studies have shown that in humans the solution to the gravito-inertial ambiguity can be modelled as a Bayesian inference with specific priors that explain the human perception and illusions [Laurens2006] We hope to extend this to the fusion of magnetic and inertial data to have a more robust perception of self rotation.

The dimensionality of the optical flow segmentation imposes limits on the applicability of full Bayesian inference. However we hope to build on previous work on Bayesian optical flow [Zelek2004] and explore the complementarity with the stereo depth maps and inertial data.

In our lab we are setting up a robotic football team for the small sized league [Simoes2005] [Rodrigues2005] [RAC2006]. The robots are small and the games are very dynamic, presenting an interesting test bed for fast visual and motion perception systems incorporating the ideas presented in this thesis.

Ongoing work is being done applying the proposed stereo depth map registration method to robotic airships that incorporate vision and inertial sensing [Mirisola2006]. The large scale fusion of depth maps presents problems when the magnetic field is not reliable for heading data, and the registration has to rely on more image features.

Wherever there is motion and cameras, or relative pose is important, inertial sensing can aid perception. For instance gesture recognition takes into account observer pose relative to the fixed world, for which the inertial gravity vertical can provide an important cue [Rett2005]. It can also incorporate results from independent motion so that a moving robot can interpret gestures on the fly.

Although the techniques presented in this thesis are based on systems that attempt to recreate the *hardware* of biological visuo-vestibular systems, no attempt has yet been made to follow the internal biological models of perception.

The usefulness of introducing models which mimic biological systems of perception and the limitations of biological perception posed by the physiological characteristics of biological motion sensors, which in certain situations yield partial or ambiguous information, has been demonstrated in previous research (see, for example, work by Reymond et al. [Reymond2002]). Biological visuo-vestibular systems take into account ego-motion, and deal well with independent motion segmentation. In spite of this, however robust, biological perception estimation processes are prone to suffering from illusions, conflicts

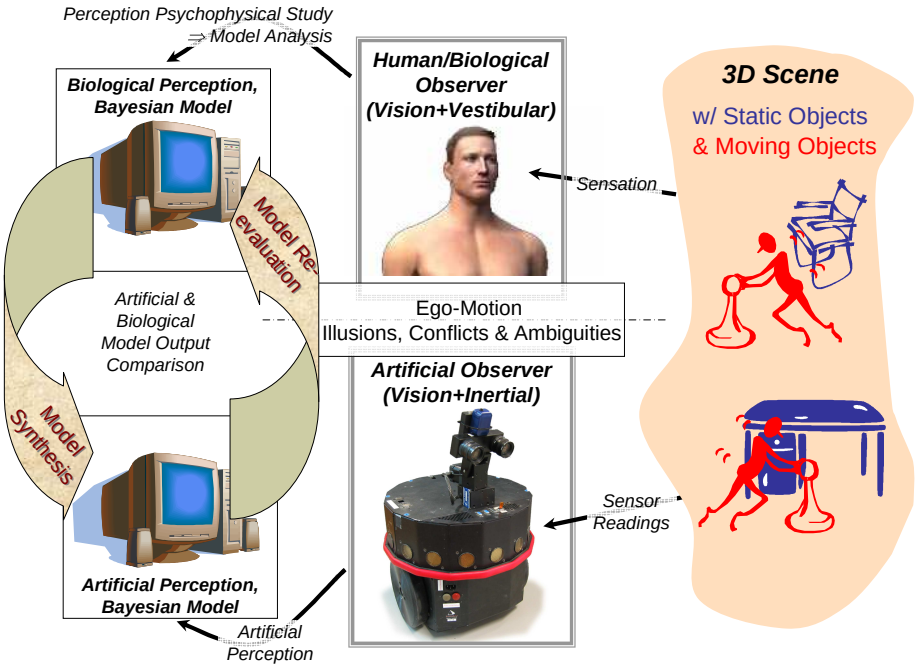


Figure 7.1: Biomimetic artificial perception research proposal schematic [Lobo2006ICVW] (human observer image courtesy of 3DScience.com).

and ambiguities.

We have thus reached a point in which the next step will be to take artificial perception to the next level: *from bioinspired to biomimetic* — see figure 7.1.

We therefore intend in future work to perform psychophysical studies, such as in [Reymond2002], of human visuo-vestibular models under a Bayesian framework, to implement these models as closely as possible using the presented technology in a robotic-based artificial perception system, to tackle 3D structure perception (specifically independent motion segmentation in the presence of self-motion), and to test the possibilities opened by the robustness of artificial sensor technology as opposed to biological sensory solutions on extreme perception tasks (see Figure 7.1) [Lobo2006ICVW].

Further studies in the field, as well as bio-inspired robotic applications, will enable a better understanding of the underlying principles, with possible application for bio-

implants of artificial vision and vestibular system in patients.

Refining and Combining Independent Motion Segmentation Methods

In the case of independent motion segmentation, we will address the use of inertial dynamic data to improve the optical flow consistency check, without depending on any tracked feature for the translation, and on combining the two methods to improve robustness.

The comparative analysis of the two approaches showed that background subtraction yields more robust results, but requires thorough coverage of scene. On the other hand, optical flow consistency does not require memory or priors regarding the scene, however is much more prone to noise.

Therefore we think that a hybrid approach is a more adequate solution, where priors gathered from past states of the workspace being perceived would be combined in a probabilistic Bayesian framework with fast low-level processing of image optical flow and inertial information.

Instead of the voxel voting scheme, a probabilistic 3D map would be more appropriate to fuse both methods. Figure 7.2 shows a probabilistic map proposed by [Rocha2006] to represent occupancy.

The fraction of the voxel volume that is actually occupied is modeled by a continuous random variable C_l, taking values $c_l \in [0, 1]$. A probability density function defined by two parameters is used to model the occupancy.

In our case we want to represent the probability of the voxel being static background and not free space, or of being part of a moving object at some time. Initially nothing is known about the observed scene, so both have flat pdfs. The probabilistic map for static background is unique and accumulates all information from previous frames. Notice that by free space we imply the complement of the static background, that might at some time be occupied by some moving object.

The non-uniform error in the depth maps due to the stereo geometry [Matthies1987] can also be taken into account using a probabilistic representation. Indication of occupancy over several frames, i.e. points in the depth maps within the same voxel, increases

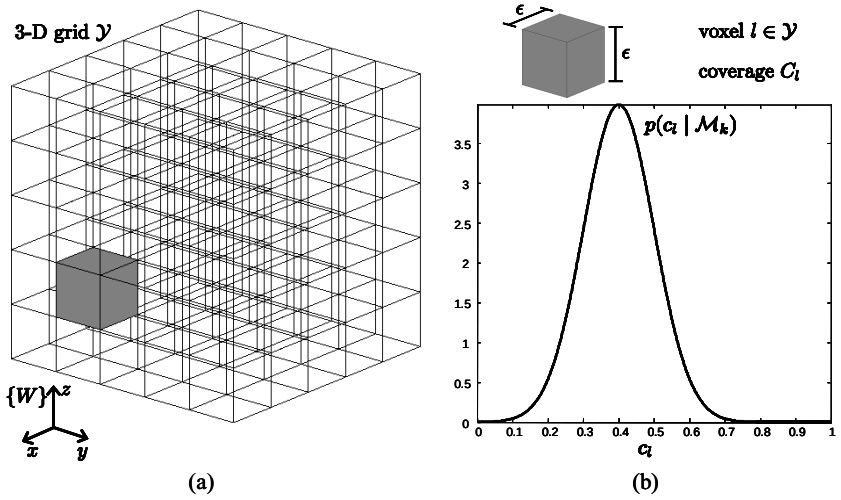


Figure 7.2: 3D probabilistic map: a) workspace divided into discrete voxels; b) the occupancy C_l of a voxel l, given the sequence of measurements $\mathcal{M}_k$, is modeled by a probability density function (pdf) $p(c_l|\mathcal{M}_k)$, in this example a normal pdf $N(\mu_l = 0.4,\ \sigma_l = 0.1)$ (taken with permission from [Rocha2006]).

the probability of being fixed background. Indication of free space, derived from space carving [Kutulakos2000] that takes into account the line of sight for reconstructed points, decreases the probability of being fixed background.

For moving objects we can initially consider probabilistic maps for each frame. A voxel that corresponds to an image point with inconsistent opticalflow has increased probability of belonging to a moving object. Indication of occupancy over just a few frames also indicates a moving object, and can be obtained by subtracting the accumulated static background map from the observed depth map. To better represent the moving objects, clustering and tracking over time has also to be addressed.

Apendix A

Notation

The following table summarises the notation used in this thesis.

Table A.1: Summary of Mathematical Notation

symbol	description
a, b, α	scalars
$\mathbf{v}$, $\mathbf{p}$, ω	vectors
$\mathbf{v} \cdot \mathbf{p}$	vector dot product
$\mathbf{v} \times \mathbf{p}$	vector cross product
$\mathbf{x} = (x_1, x_2, x_3)^\mathsf{T}$	column vector (transpose)
$\dot{\mathbf{v}}$	vector 1^{st} derivative
$\ddot{\mathbf{v}}$	vector 2^{nd} derivative
$\mathbf{A}$, $\mathbf{M}$	matrices
$\mathbf{A}^\mathsf{T}$	matrix transpose
$\mathring{\mathbf{q}}$, $\mathring{\mathbf{p}}$	quaternions
$\mathring{\mathbf{q}} = q_0 + \mathbf{q}$	quaternion scalar and vector part
$\mathring{\mathbf{q}}^*$	quaternion conjugate
$\mathring{\mathbf{p}}\mathring{\mathbf{q}}$	quaternion product
$\mathring{\mathbf{p}} \cdot \mathring{\mathbf{q}}$	quaternion dot product
$\dot{\mathring{\mathbf{q}}}$	quaternion 1^{st} derivative
$\ddot{\mathring{\mathbf{q}}}$	quaternion 2^{nd} derivative
$\{\mathcal{I}\}$, $\{\mathcal{C}\}$, $\{\mathcal{N}\}$	frames of reference
${}^{\mathcal{I}}\mathbf{T}_{\mathcal{C}}$	transformation matrix from $\{\mathcal{I}\}$ to $\{\mathcal{C}\}$
${}^{\mathcal{C}}\boldsymbol{\omega}$	vector $\boldsymbol{\omega}$ in $\{\mathcal{C}\}$ frame of ref

Scalars are represented by simple italic font, vectors by boldface non-italic roman font, quaternions are similar but with a small circle on top, and matrices by boldface roman capitals. Frames of reference use capital calligraphy font.

Apendix B

InerVis WebIndex

We created the *InerVis WebIndex* as reference web site for research work in the field. It provides several topics, such as a bibliography list, InerVis workshops, software and links for camera and inertial sensors manufactures.

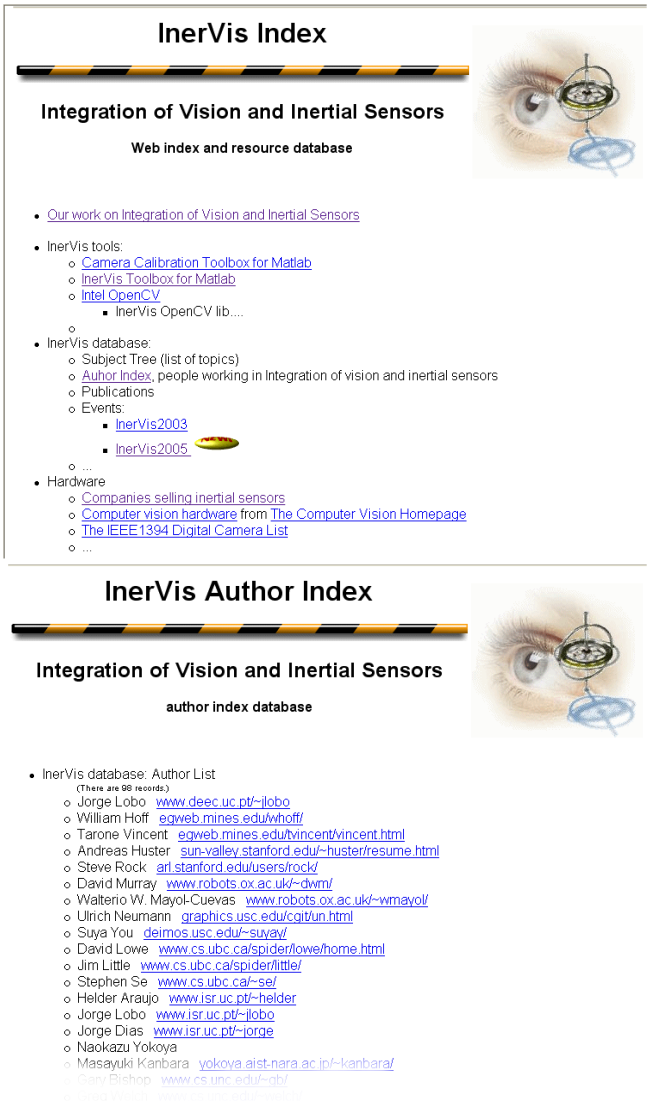


Figure B.1: InerVis Webindex (`:http://www.deec.uc.pt/~jlobo/InerVis_WebIndex/`) and InerVis Author Index (`:http://www.deec.uc.pt/~jlobo/InerVis_WebIndex/InerVis_Author.php`)

Apendix C

InerVis Matlab Toolbox

Within the *InerVis WebIndex* the *InerVis Matlab Toolbox* page shares our calibration methods with the community, as a Matlab toolbox complete with examples.

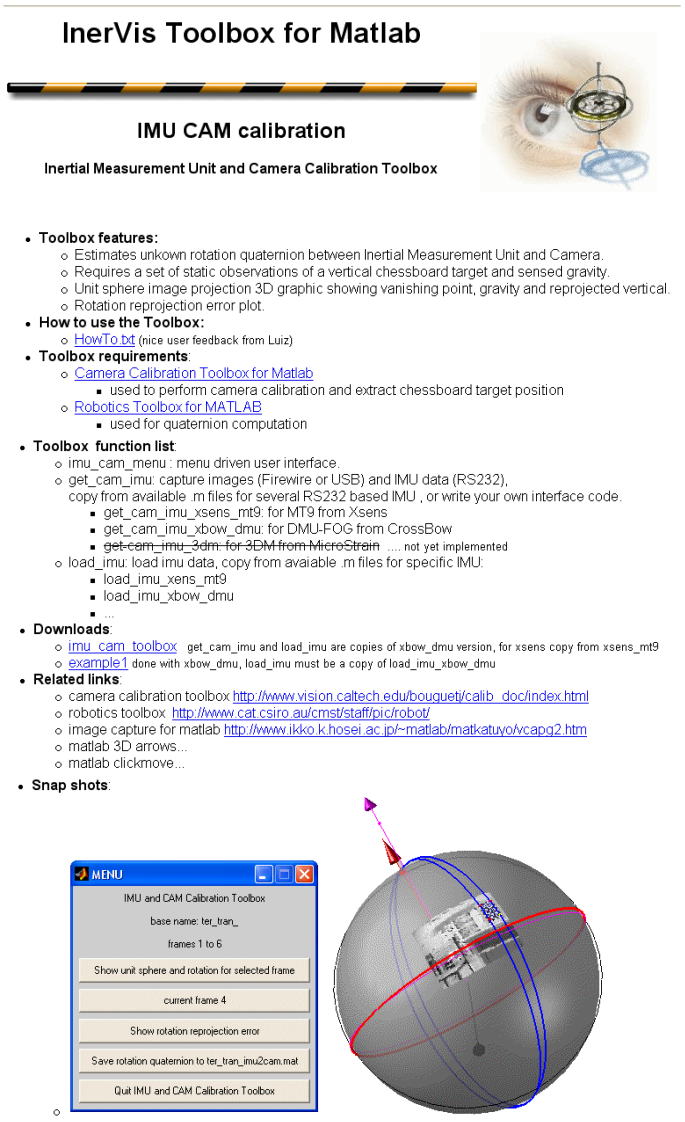


Figure C.1: InerVis Author Index (`:http://www.deec.uc.pt/~jlobo/InerVis_WebIndex/InerVis_Toolbox.html`)

References

[Alenya2004JRS] Guillem Alenya, Elisa Martnez, and Carme Torras. Fusing visual and inertial sensing to recover robot ego-motion. *Journal of Robotic Systems*, 21(1):23–32, January 2004.
URL PDF ... 12

[Allen1998] J.J. Allen, R.D. Kinney, J. Sarsfield, M.R. Daily, J.R. Ellis, J.H. Smith, S. Montague, R.T. Howe, B.E. Boser, R. Horowitz, A.P. Pisano, M.A. Lemkin, W.A. Clark, and T. Juneau. Integrated Micro-Electro-Mechanical Sensor Development for Inertial Applications. *IEEE Aearospace and Electronic Systems Magazine*, 13:36–40, November 1998.
URL PDF ... 16

[Alves2003] João Alves, Jorge Lobo, and Jorge Dias. Camera-inertial sensor modeling and alignment for visual navigation. *Machine Intelligence and Robotic Control*, 5(3):103–112, September 2003.
URL PDF ... 54

[Alves2003ICAR] João Alves, Jorge Lobo, and Jorge Dias. Camera-Inertial Sensor modelling and alignment for Visual Navigation. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 1693–1698, Coimbra, Portugal, July 2003.
PDF ... 54

[Ames1997] A. L. Ames, D. R. Nadeau, and J. L. Moreland. *VRML 2.0 Sourcebook*. John Wiley and Sons, 2nd edition, 1997. ISBN 0-471-16507-7.
URL ... 94

[AnalogDevices] Analog Devices. Analog devices, mems and sensors. http://www.analog.com.
URL ... 17, 85

[Angelaki1999] D E Angelaki, M Q McHenry, J D Dickman, S D Newlands, and B J Hess. Computation of inertial motion: neural strategies to resolve ambiguous otolith information. *The Journal Of Neuroscience: The Official Journal Of The Society For Neuroscience*, 19(1):316–327, January 1999.
URL PDF ... 10

[Azuma1999] R. Azuma, B. Hoff, H. Neely, and R. Sarfaty. A motion-stabilized outdoor augmented reality system. In *Proceedings of IEEE Virtual Reality*, pages 252–259, March 1999.
URL PDF ... 12

[Barbour1999] N. Barbour and G. Schmidt. Inertial sensor technology trends. In *The Draper Technology Digest*, volume 3, pages 5–13. Draper Laboratory, 1999.
URL PDF ... 17

[Barbour2001] N. Barbour and G. Schmidt. Inertial sensor technology trends. *Sensors Journal, IEEE*, 1(4):332–339, December 2001.
URL PDF ... 11, 17

[Barron1994] J.L. Barron, D.J. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.
PDF ... 34

[Beer2002] Jeremy Beer, Colin Blakemore, Fred Previc, and Mario Liotti. Areas of the human brain activated by ambient visual motion, indicating three kinds of self-movement. *Experimental Brain Research*, 143(1):78–88, March 2002.
URL PDF ... 10

[Berthoz2000] Alain Berthoz. *The Brain's Sense of Movement*. Havard University Press, 2000. ISBN: 0-674-80109-1.
... 1, 3

[Bouguet2006] Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab.
http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2006.
URL ... 45, 59, 63, 72, 79

[Brillault1991] B. Brillault and O'Mahony. New Method for Vanishing Point Detection. *CVGIP: Image Understanding*, 54(2):289–300, 1991.
URL ... 77

[Britannica2001] Encyclopaedia Britannica. www.britannica.com, 2001.
URL ... 4, 5

[Caprile1990] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, 4(2):127–140, 1990.
URL ... 77

[Carpenter1988] H. Carpenter. *Movements of the Eyes*. London Pion Limited, 2nd edition, 1988. ISBN 0-85086-109-8.
... 1, 3

[Caruso1998] Michael J. Caruso, Tamra Bratland, Carl H. Smith, and Robert Schneider. A New Perspective on Magnetic Field Sensing. Technical report, Honeywell, Inc, 1998.
URL PDF ... 85, 116

[Chae2005] Junseok Chae, H. Kulah, and K Najafi. A monolithic three-axis micro-g micromachined silicon capacitive accelerometer. *Microelectromechanical Systems, Journal of*, 14(2):235–242, April 2005.
URL PDF ... 17

[Chai2002] Lin Chai, William A. Hoff, and Tyrone Vincent. Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality. *Presence: Teleoper. Virtual Environ.*, 11(5):474–492, October 2002.
URL PDF ... 12

[Chalimbaud2005] P. Chalimbaud, F. Berry, F. Marmoiton, and S. Alizon. Design of a hybrid visuo-inertial smart sensor. In *ICRA 2005 Workshop on Integration of Vision and Inertial Sensors (InerVis2005)*, April 2005.
URL PDF ... 11

[Chen2004] J. Chen and A. Pinz. Structure and motion by fusion of inertial and vision-based tracking. In W. Burger and J. Scharinger, editors, *Digital Imaging in Media and Education*, volume 179 of *Schriftenreihe*, pages 55–62. OCG, 2004. Proceedings of the 28^{th} ÖAGM/AAPR Conference.
URL PDF ... 12

[Chroust2004JRS] S. G. Chroust and M. Vincze. Fusion of vision and inertial data for motion and structure estimation. *Journal of Robotic Systems*, 21(2):73–83, February 2004.
URL PDF ... 12

[Coorg1998] Satyan R. Coorg. *Pose Imagery and Automated Three-Dimensional Modeling of Urban Environments*. PhD thesis, Massachusetts Institute of Technology, September 1998.
URL PDF ... 81

[Coren1994] Stanley Coren, Lawrence M. Ward, and James T. Enns. *Sensation and Perception*. Harcourt Brace & Company, fourth edition edition, 1994. ISBN 0-15-500103-5.
... 3

[Corke2004JRS] Peter Corke. An inertial and visual sensing system for a small autonomous helicopter. *Journal of Robotic Systems*, 21(2):43–51, February 2004.
URL PDF ... 12

[Daniilidis1999] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotic Research*, 18(3):286–298, March 1999.
URL PDF ... 44, 66, 81

[Dias1995] Jorge Dias, Carlos Paredes, Inácio Fonseca, and A. T. de Almeida. Simulating Pursuit with Machines. Experiments with robots and artificial vision. In *Proceedings of the 1995 IEEE Conference on Robotics and Automation*, volume 1, pages 472–477, Nagoya, Japan, May 1995.
URL PDF ... 90

[Dias1998] Jorge Dias, Carlos Paredes, Inacio Fonseca, Helder Araujo, Jorge Baptista, and Anibal Traca de Almeida. Simulating Pursuit with Machine Experiments with Robots and Artificial Vision. *IEEE Transactions on Robotics and Automation*, 14(1):1–18, February 1998.
URL PDF ... 90

[Dickmanns1998] Ernst D. Dickmanns. Vehicles capable of dynamic vision: a new breed of technical beings? *Artificial Intelligence*, 103(1-2):49–76, August 1998.
URL PDF ... 12

[Diel2005] David D. Diel, Paul DeBitetto, and Seth Teller. Epipolar constraints for vision-aided inertial navigation. *Proc. IEEE Motion and Video Computing*, pages 221–228, January 2005.
URL PDF ... 12

[Dorst2005] Leo Dorst. First order error propagation of the procrustes method for 3d attitude estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):221–229, February 2005.
URL PDF ... 44

[Eason1992] R. O. Eason and R. C. Gonzalez. Least-Squares Fusion of Multisensory Data. In Mongi A. Abidi and Rafael C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, chapter 9. Academic Press, 1992.
... 2

[Eveland1998] Christopher Eveland, Kurt Konolige, and Robert C. Bolles. Background modeling for segmentation of video-rate stereo sequences. In *Conference on Vision and Pattern Recognition*, Santa Barbara, CA, USA, June 1998.
URL PDF ... 115

[Foxlin2003VR] Eric Foxlin and Leonid Naimark. Vis-tracker: A wearable vision-inertial self-tracker. In *Proceedings of the IEEE Virtual Reality 2003*, page 199. IEEE Computer Society, 2003.
URL PDF ... 12

[Foxlin2004] E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington. Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision. In *Proceedings of Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 212–221, November 2004.
URL PDF ... 12

[Fukushima1997] Kikuro Fukushima. Corticovestibular interactions: anatomy, electrophysiology, and functional considerations. *Experimental Brain Research*, 117(1):1–16, October 1997.
URL PDF ... 10

[Gillingham1996] Kent K. Gillingham and Fred. H. Previc. *Spatial Orientation in Flight*, chapter 11. Williams and Wilkins, second edition, 1996.
... 1, 3, 7

[Gluckman1998] Joshua Gluckman and Shree K. Nayar. Ego-Motion and Omnidirectional Cameras . In *Proceedings of International Conference on Computer Vision (ICCV'98)*, pages 999–1005, Bombay, India, January 1998.
URL PDF ... 33, 35

[Goedeme2004JRS] Toon Goedem, Marnix Nuttin, Tinne Tuytelaars, and Luc Van Gool. Vision based intelligent wheel chair control: The role of vision and inertial sensing in topological navigation. *Journal of Robotic Systems*, 21(2):85–94, February 2004.
URL PDF ... 12

[Goldbeck2000] J. Goldbeck, B. Huertgen, S. Ernst, and L. Kelch. Lane following combining vision and dgps. *Image and Vision Computing*, 18(5):425–433, April 2000.
URL PDF ... 12

[Graovac2004JRS] Stevica Graovac. Principles of fusion of inertial navigation and dynamic vision. *Journal of Robotic Systems*, 21(1):13–22, January 2004.
URL PDF ... 12

[Grimm2004] M. Grimm and R.-R. Grigat. Real-time hybrid pose estimation from vision and inertial data. In *Proceedings of First Canadian Conference on Computer and Robot Vision*, pages 480–486, 2004.
URL PDF ... 11

[Hague2000] T. Hague, J. A. Marchant, and N. D. Tillett. Ground based sensing systems for autonomous agricultural vehicles. *Computers and Electronics in Agriculture*, 25(1-2):11–28, January 2000.
URL PDF ... 12

[Harris2000] Laurence R. Harris, Michael Jenkin, and Daniel C. Zikovitz. Visual and non-visual cues in the perception of linear self motion. *Experimental Brain Research*, 135(1):12–21, October 2000.
URL PDF ... 10

[Hartley2000] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univercity Press, 2000. ISBN: 0521623049.
URL ... 92

[Hoff1996] William A. Hoff, Khoi Nguyen, and Torsten Lyon. Computer vision-based registration techniques for augmented reality. In *Proceedings of Intelligent Robots and Computer Vision*, pages 538–548, November 1996.
URL PDF ... 12

[Hogue2004] A. Hogue, M.R. Jenkin, and R.S. Allison. An optical-inertial tracking system for fully-enclosed vr displays. In *Proceedings of the First Canadian Conference on Computer and Robot Vision*, pages 22–29, May 2004.
URL PDF ... 12

[Horn1987] B.K.P Horn. Closed-Form Solution of Absolute Orientation Using Unit Quaternions. *Journal of the Optical Society of America*, 4(4):629–462, April 1987.
URL PDF ... 44, 55, 57

[Hurley2001] Susan Hurley. Perception and action: Alternative views. *Synthese*, 129(1):3–40, October 2001.
URL PDF ... 10

[Intel2006] Intel. Intel Open Source Computer Vision Library. http://www.intel.com/technology/computing/opencv/index.htm, 2006.
URL ... 45

[Jahne1997] Bernd Jahne. *Digital Image Processing*. Springer-Verlag, 1997. ISBN 3-540-62724-3.
... 97

[Jiang2004] B. Jiang, U. Neumann, and Suya You. A robust hybrid tracking system for outdoor augmented reality. In *Proceedings of the IEEE Virtual Reality*, pages 3–275, 2004.
URL PDF ... 12

[Jung2001] S.-H. Jung and C.J. Taylor. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–732–II–737, 2001.
PDF ... 12

[Kanatani1993] Kenichi Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993. ISBN 0-19-856385-X.
... 22, 77

[Kelly2005] Jonathan W. Kelly, Jack M. Loomis, and Andrew C. Beall. The importance of perceived relative motion in the control of posture. *Experimental Brain Research*, 161(3):285–292, March 2005.
URL PDF ... 10

[Klein2004] G. S. W. Klein and T. W. Drummond. Tightly integrated sensor fusion for robust visual tracking. *Image and Vision Computing*, 22(10):769–776, September 2004.
URL PDF ... 12

[Konolige1997] Kurt Konolige. Small vision systems: Hardware and implementation. In *Eighth International Symposium on Robotics Research*, Hayama, Japan, October 1997.
URL PDF ... 106, 121

[Kurazume2000] R. Kurazume and S. Hirose. Development of image stabilization system for remote operation of. In *Proceedings. ICRA '00. IEEE International Conference on Robotics and Automation*, volume 2, pages 1856–186, April 2000.
URL PDF ... 11

[Kutulakos2000] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.
URL PDF ... 133

[Lang2002] P. Lang, A. Kusej, A. Pinz, and G. Brasseur. Inertial tracking for mobile augmented reality. In *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference*, volume 2, pages 1583–1587 vol.2, May 2002.
URL PDF ... 12

[Lang2005] Lang P. and Pinz A. Calibration of hybrid vision / inertial tracking systems. In 2^{nd} *InverVis 2005: Workshop on Integration of Vision and Inertial Systems*, Barcelona, Spain, April 2005.
URL PDF ... 44

[Laurens2006] Jean Laurens and Jacques Droulez. Bayesian processing of vestibular information. *Biological Cybernetics*, December 2006. (Published online: 5th December 2006).
URL PDF ... 130

[Lawrence1998] Anthony Lawrence. *Modern Inertial Technology: Navigation, Guidance, and Control*. Mechanical Engineering Series. Springer, 2nd edition edition, December 1998. ISBN 0-387-98507-7.
URL ... 15

[Leone1998] Gilles Leone. The effect of gravity on human recognition of disoriented objects. *Brain Research Reviews*, 28(1-2):203–214, November 1998.
URL PDF ... 10

[Li1994] M. Li. Camera calibration of the kth head-eye system. In *ECCV94*, pages A:543–554, 1994.
URL PDF ... 77

[Li2005] Rui Li and Stan Sclaroff. Multi-scale 3d scene flow from binocular stereo sequences. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)*, volume 2, pages 147 – 153, January 2005.
URL PDF ... 115

[Lobo2001MFI] Jorge Lobo and Jorge Dias. Fusing of image and inertial sensing for camera calibration. In *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI 2001*, pages 103–108, Baden-Baden, Germany, August 2001.
URL PDF ... 80

[Lobo2001SIRS] Jorge Lobo, Carlos Queiroz, and Jorge Dias. Vertical world feature detection and mapping using stereo vision and accelerometers . In *Proceedings of the 9th International Symposium on Intelligent Robotic Systems - SIRS'01*, pages 229–238, Toulouse, France, July 2001.
URL PDF ... 102

[Lobo2002MSc] Jorge Lobo. Inertial Sensor Data Integration in Computer Vision Systems. Master's thesis, University of Coimbra, April 2002.
URL PDF ... 8, 17, 79, 85, 94

[Lobo2003JRAS] Jorge Lobo, Carlos Queiroz, and Jorge Dias. World feature detection and mapping using stereovision and inertial sensors. *Robotics and Autonomous Systems*, 44(1):69–81, July 2003.
URL PDF ... 8, 100, 102

[Lobo2003PAMI] Jorge Lobo and Jorge Dias. Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1597–

1608, December 2003.
URL PDF ... 8, 115

[Lobo2004JRS] Jorge Lobo and Jorge Dias. Inertial sensed ego-motion for 3d vision. *Journal of Robotic Systems*, 21(1):3–12, January 2004.
URL PDF ... 8, 115

[Lobo2005InerVis] Jorge Lobo and Jorge Dias. Relative pose calibration between visual and inertial sensors. In *ICRA 2005 Workshop on Integration of Vision and Inertial Sensors (InerVis2005)*, Barcelona, Spain, April 2005.
PDF ... 8

[Lobo2006] Jorge Lobo. InerVis Toolbox for Matlab.
http://www.deec.uc.pt/ jlobo/InerVis_WebIndex/, 2006.
URL ... 9, 63, 76

[Lobo2006ICVW] Jorge Lobo, Joo Filipe Ferreira, and Jorge Dias. Bioinspired visuovestibular artificial perception system for independent motion segmentation. In *Second Inernational Cognitive Vision Workshop, ECCV 9th European Conference on Computer Vision*, Graz, Austria, May 2006.
URL PDF ... 9, 131

[Lucas1981] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision . In *Proceedings of Imaging Understanding Workshop*, pages 674–679, 1981.
PDF ... 35, 124

[Matthies1987] L. Matthies and Steven Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):239 – 250, June 1987.
URL PDF ... 132

[Mirisola2006] Luiz G. B. Mirisola, Jorge Lobo, and Jorge Dias. Stereo vision 3d map registration for airships using vision-inertial sensing. In *The 12th IASTED Int. Conf. on Robotics and Applications*, Honolulu, USA, August 2006.
URL PDF ... 130

[Mukai2000] Toshiharu Mukai and Noboru Ohnishi. Object shape and camera motion recovery using sensor fusion of a video camera and a gyro sensor. *Information Fusion*, 1(1):45–53, July 2000.
URL PDF ... 11

[Muratet2005] Laurent Muratet, Stephane Doncieux, Yves Briere, and Jean-Arcady Meyer. A contribution to vision-based autonomous helicopter flight in urban environments. *Robotics and Autonomous*

Systems, 50(4):195–209, March 2005.
URL PDF ... 12

[Naimark2002] Leonid Naimark and Eric Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 27. IEEE Computer Society, 2002.
URL ... 12

[Nayak2005] Jagannath Nayak and V.K Saraswat. Studies on micro opto electro mechanical (moem) inertial sensors for future inertial navigation systems. In *International Conference on Smart Materials Structures and Systems*, volume SE, pages 28–35, Bangalore, India, July 2005.
URL PDF ... 17

[Nebot1997] E. Nebot and H. Durrant-Whyte. Initial calibration and alignment of an inertial navigation system. In *Proceedings of the 4th Annual Conference on Mechatronics and Machine Vision in Practice*, page 175. IEEE Computer Society, 1997.
URL ... 46

[Neumann2003] U. Neumann, Suya You, Jinhui Hu, Bolan Jiang, and JongWeon Lee. Augmented virtual environments (ave): dynamic fusion of imagery and 3d models. In *Proceedings of the IEEE Virtual Reality*, pages 61–67, 2003.
URL PDF ... 12

[Nygards2004JRS] Jonas Nygards, Per Skoglar, Morgan Ulvklo, and Tomas Hgstrm. Navigation aided image processing in uav surveillance: Preliminary results and design of an airborne experimental system. *Journal of Robotic Systems*, 21(2):63–72, February 2004.
URL PDF ... 12

[Panerai1998] Francesco Panerai and Giulio Sandini. Oculo-motor stabilization reflexes: integration of inertial and visual information. *Neural Networks*, 11(7-8):1191–1204, October 1998.
URL PDF ... 11

[Panerai2000] Francesco Panerai, Giorgio Metta, and Giulio Sandini. Visuo-inertial stabilization in space-variant binocular systems. *Robotics and Autonomous Systems*, 30(1-2):195–214, January 2000.
URL PDF ... 11

[Panerai2002] F. Panerai, G. Metta, and G. Sandini. Learning visual stabilization reflexes in robots with moving eyes. *Neurocomputing*, 48(1-4):323–337, October 2002.
URL PDF ... 11

[Previc2000] Fred H. Previc, Mario Liotti, Colin Blakemore, Jeremy Beer, and Peter Fox. Functional imaging of brain areas involved in the processing of coherent and incoherent wide field-of-view visual motion. *Experimental Brain Research*, 131(4):393–405, April 2000.
URL PDF ... 10

[Qian2001] G. Qian, R. Chellappa, and Q. Zheng. Robust structure from motion estimation using inertial data. *Journal Optical Society of America A*, 18(12):2982–2997, December 2001.
URL PDF ... 12

[Qian2002] Gang Qian, R. Chellappa, and Qinfen Zheng. Bayesian structure from motion using inertial information. In *Proceedings of the International Conference on Image Processing*, volume 3, pages III–425–III–428, 2002.
PDF ... 12

[RAC2006] Jorge Lobo, Rui Rocha, and Jorge Dias. RAC Robtica Acadmica de Coimbra. http://www.deec.uc.pt/ jlobo/RAC/, 2006.
URL ... 130

[Rehbinder2003] H. Rehbinder and B.K. Ghosh. Pose estimation using line-based dynamic vision and inertial sensors. *Automatic Control, IEEE Transactions on*, 48(2):186–199, February 2003.
URL PDF ... 11

[Rett2005] Joerg Rett and Jorge Dias. Gesture recognition based on visual-inertial data - registering gravity in the gesture plane. In *Proceedings of the Colloquium of Automation*, 2005.
URL ... 130

[Reymond2002] Gilles Reymond, Jacques Droulez, and Andras Kemeny. Visuovestibular perception of self-motion modeled as a dynamic optimization process. *Biological Cybernetics*, 87(4):301–314, October 2002.
URL PDF ... 10, 130, 131

[Ribo2004JRS] Miguel Ribo, Markus Brandner, and Axel Pinz. A flexible software architecture for hybrid tracking. *Journal of Robotic Systems*, 21(2):53–62, February 2004.
URL PDF ... 12

[Rocha2006] Rui Rocha. *Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing: a Distributed Control Approach based on Entropy.* PhD thesis, Faculty of Engineering of University of Porto, Portugal, May 2006.
PDF ... 132, 133

[Rodrigues2005] João Rodrigues, Ségio Brandão, Jorge Lobo, Rui Rocha, and Jorge Dias. Rac robotic soccer small-size team: Omnidirectional drive modelling and robot construction. In *Proceedings of Robótica 2005 Encontro Científico*, Coimbra, April 2005.
PDF ... 130

[Roetenberg2003] D. Roetenberg, H. Luinge, and P. Veltink. Inertial and magnetic sensing of human movement near ferromagnetic materials. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 268–269, October 2003.
PDF ... 116

[Roetenberg2005] D. Roetenberg, H.J. Luinge, C.T.M. Baten, and P.H. Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Rehabilitation Engineering]*, 13(3):395–405, September 2005.
URL PDF ... 116

[Roumeliotis2002] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, page 4326, May 2002.
URL PDF ... 12

[Rourke1993] Joseph O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1993. ISBN 0-512-22592-2.
URL ... 94

[Savage1984] Paul G. Savage. *Strapdown System Algorithms, Advances in strapdown inertial systems.*, chapter 3, pages 3.1–3.30. Lecture Series 133. AGARD, Advisory Group for Aerospace Research and Development, 1984.
... 19

[Shuster1993] M.D. Shuster. The kinematic equation for the rotation vector. *Aerospace and Electronic Systems, IEEE Transactions on*, 29(1):263–267, January 1993.
URL PDF ... 19

[Simoes2005] José Rui Simões, Rui Rocha, Jorge Lobo, and Jorge Dias. Rac robotic soccer small-size team: Control architecture and global vision. In *Proceedings of Robótica 2005 Encontro Científico*, Coimbra, April 2005.
PDF ... 130

[Smith1997] Stephen M. Smith and J. Michael Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.
URL PDF ... 94

[Stolfi1991] Jorge Stolfi. *Oriented projective geometry, a framework for geometric computations* . Boston Academic Press, 1991.
... 22

[Stratmann2004JRS] Irem Stratmann and Erik Solda. Omnidirectional vision and inertial clues for robot navigation. *Journal of Robotic Systems*, 21(1):33–39, January 2004.
URL PDF ... 12

[Strelow2002] Dennis Strelow and Sanjiv Singh. Optimal motion estimation from visual and inertial measurements. In *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, page 314. IEEE Computer Society, 2002.
URL PDF ... 12

[Strelow2003] Dennis Strelow and Sanjiv Singh. Optimal motion estimation from visual and inertial measurements. In *Proceedings of the Workshop on Integration of Vision and Inertial Sensors (INERVIS 2003)*, June 2003.
URL PDF ... 12

[SummitInstruments] Summit Instruments. http://www.summitinstruments.com/.
... 85

[Tsai1989] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Trans. Robot. Automat.*, 5(3):345358, June 1989.
URL PDF ... 44, 69, 74, 81

[Vedula1999] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729, 1999.
URL PDF ... 115

[Vedula2005] S. Vedula, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):475–480, March 2005.
URL PDF ... 115

[Videre] Videre. Videre Design. http://www.videredesign.com/.
URL ... 106, 121

[Vieville1989] T. Viéville and O.D. Faugeras. Computation of Inertial Information on a Robot. In Hirofumi Miura and Suguru Arimoto, editors, *Fifth International Symposium on Robotics Research*, pages 57–65. MIT-Press, 1989.
PDF ... 10, 46, 47

[Vieville1990] T. Viéville and O.D. Faugeras. Cooperation of the Inertial and Visual Systems. In Thomas C. Henderson, editor, *Traditional and NonTraditional Robotic Sensors*, volume F 63 of *NATO ASI*, pages 339–350. SpringerVerlag Berlin Heidelberg, 1990.
... 10

[Vieville1993ICCV] T. Viéville, E. Clergue, and P.E.D. Facao. Computation of ego-motion and structure from visual an inertial sensor using the vertical cue. In *Proceedings of the Fourth International Conference on Computer Vision.*, pages 591–598, May 1993.
URL PDF ... 10

[Vieville1993IROS] Thierry Viéville, François Romann, Bernard Hotz, Hervé Mathieu, Michel Buffa, Luc Robert, P.E.D.S. Facao, Olivier Faugeras, and J.T. Audren. Autonomous navigation of a mobile robot using inertial and visual cues. In M. Kikode, T. Sato, and K. Tatsuno, editors, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 360 – 367, Yokohama, Japan, July 1993.
URL PDF ... 10

[Vieville1995] T. Viéville, E. Clergue, and P. E. Dos Santos Facao. Computation of ego motion using the vertical cue. *Mach. Vision Appl.*, 8(1):41–52, 1995.
PDF ... 10

[Vieville1997] Thierry Viéville. *A Few Steps Towards 3D Active Vision.* Springer-Verlag, 1997. ISBN=3540631062.
URL ... 10

[Viollet2005] Stephane Viollet and Nicolas Franceschini. A high speed gaze control system based on the vestibulo-ocular reflex. *Robotics and Autonomous Systems*, 50(4):147–161, March 2005.
URL PDF ... 11

[Wall2003] C. Wall, D.M. Merfelda, S.D. Raucha, and F.O. Blackf. Vestibular prostheses: The engineering and biomedical issues. *Journal of Vestibular Research*, (12):95113, 2003.
URL PDF ... 11

[Wang1991] Ling-Ling Wang and Wen-Hsiang Tsai. Camera Calibration by Vanishing Lines for 3-D Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370–

376, April 1991.
URL PDF ... 77

[Willson1994] Reg G. Willson and Steven A. Shafer. What is the center of the image. *Journal of the Optical Society of America A*, 11(11):2946–2955, 1994.
URL PDF ... 77

[Xsens] Xsens. Xsens Technologies. http://www.xsens.com/.
URL ... 121

[Yang2005] W. Yang, K. Ngan, J. Lim, and K. Sohn. Joint motion and disparity fields estimation for stereoscopic video sequences. *Signal Processing: Image Communication*, 20(3):265–276, March 2005.
URL PDF ... 115

[Yazdi1998] N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. *Proceedings of the IEEE*, 86(8):1640–1659, August 1998.
URL PDF ... 11, 16, 17

[You2001] Suya You and Ulrich Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *Proceedings of the IEEE Virtual Reality*, page 71. IEEE Computer Society, 2001.
URL ... 11

[Zelek2004] John S. Zelek. Towards bayesian real-time optical flow. *Image and Vision Computing*, 22(12):1051–1069, October 2004.
URL PDF ... 130

[Zhang1999] Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, volume 1, pages 666–673, Kerkyra, Greece, September 1999.
URL PDF ... 45

[Zhang2003] Ye Zhang and C Kambhamettu. On 3-d scene flow and structure recovery from multiview image sequences. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 33(4):592–606, August 2003.
URL PDF ... 115