University of Coimbra

Faculty of Science and Tecnology

Electrical and Computer Engineering Department

INERTIAL SENSOR DATA INTEGRATION IN COMPUTER VISION SYSTEMS

M.Sc. Thesis

Jorge Nuno de Almeida e Sousa Lobo

Electrical Engineer

Coimbra

February 2002

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 Master of Science.

This work was done under the supervision of Doctor Jorge Manuel Miranda Dias, assistant 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 girlfriend, Sílvia.

"A thesis is a very large work. It 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; JNICT[2] for the scholarship and project grant; DEEC[3] for employment and support for my work. I would also like to thank all my colleagues at ISR, Luis Almeida, Inácio, Nuno Cid, Arlindo, João Filipe and Paulo Menezes, amongst others. I would like to especially thank my parents and family for all their love and support, my father for proofreading the thesis and providing useful comments and sugestions, and Sílvia, for her love, support and inspiration.

[1]Institute of Systems and Robotics, Coimbra

[2]Junta Nacional de Investigação em Ciência e Tecnologia, now FCT-Fundação para a Ciência e Tecnologia, from the Portuguese Government. Financial support for this work was partially provided by FCT. Jorge Lobo received from JNICT a scholarship under the PRAXIS XXI program for the MSc. The project SIRMA, inertial system for autonomous mobile robots, was also financed by JNICT

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

Abstract

Advanced sensor systems, exploring high integration of multiple sensorial modalities, have been significantly increasing the capabilities of autonomous robots and enlarging the application potential of vision systems. In this work I explore the cooperation between image and inertial sensors, motivated by what happens with the vestibular system and vision in humans and animals. Visual and inertial sensing are two sensory modalities that can be explored to give robust solutions on image segmentation and recovery of three-dimensional structure.

In this work I overview the currently available low-cost inertial sensors. Using some of these sensors, I have built an inertial system prototype and coupled it to the vision system used in this work. The vision system has a set of stereo cameras with vergence. Using the information about the vision system's attitude in space, given by the inertial sensors, I obtained some interesting results. I use the inertial information about the vertical to infer one of the intrinsic parameters of the visual sensor - the focal distance. The process involves having at least one image vanishing point, and tracing an artificial horizon.

Based on the integration of inertial and visual information, I was able to detect three-dimensional world features such as the ground plane and vertical features. Relying on the known vertical reference, and a few system parameters, I was able to determine the ground plane geometric parameters and the stereo pair mapping of image points that belong to the ground plane. This enabled the segmentation and three-dimensional reconstruction of ground plane patches. It was also used to identify the three-dimensional vertical structures in a scene. Since the vertical reference does not give a heading, image vanishing points can be used as an external heading reference. These features can be used to build a *metric map* useful to improve mobile robot navigation and autonomy.

Contents

Chapter 1

Introduction

1.1 Motivation

The advances in automation and robotics are putting evermore demands on the technology. What some years ago was science fiction, is now feasible. 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. I think that other sensory modalities can aid 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 animals the vestibular system gives inertial information essential for spatial orientation, body posture, equilibrium, reflex behaviours such as eye movement coordination, and navigation. The vision, vestibular and proprioceptive (muscle, tendon, and joint receptors) systems are the key elements in spatial orientation and posture control. The robotic analogy would be the cameras, inertial sensors and joint encoders.

In humans, vision is by far the most important sensory modality for spatial orientation. Visual and vestibular orientation information are integrated at very basic neural levels, suggesting that inertial sensors can play an important role in artificial vision systems.

In the next sections I will briefly describe the human vestibular system, its main func-

tions and interactions with human vision. For a more detailed and medical description see [Gillingham and Previc, 1996], and for more on human senses and perception see [Coren et al., 1994] and [Carpenter, 1988]. [Berthoz, 2000] gives insight into neural interactions of human vision and vestibular system, and action-perception behaviours.

1.1.1 Human Vestibular System

Within the vestibule of inner ear we find the human inertial sensor, the vestibular system (see figure 1.1). 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.

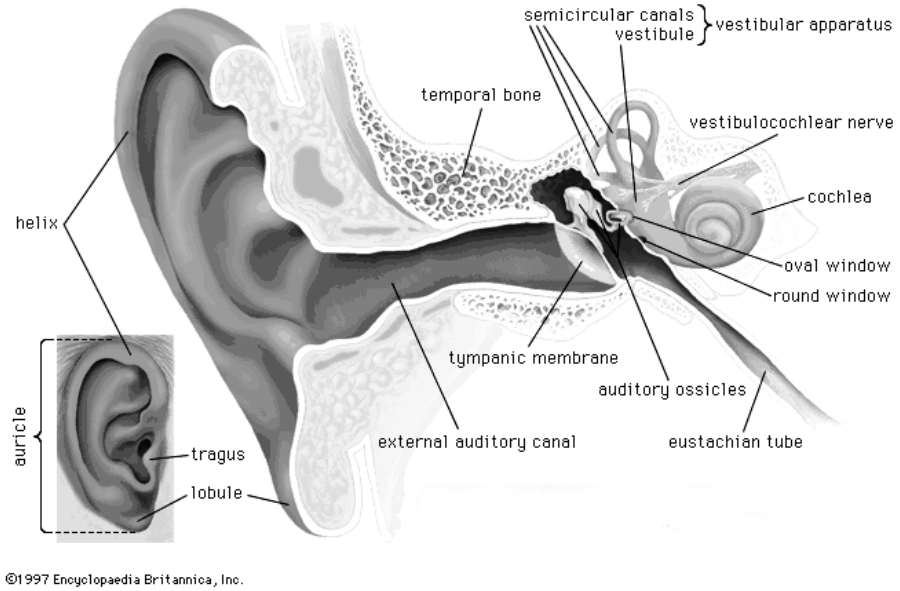


Figure 1.1: Human Ear (taken from [Encyclopaedia Britannica, 2001]).

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 1.1, 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 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.

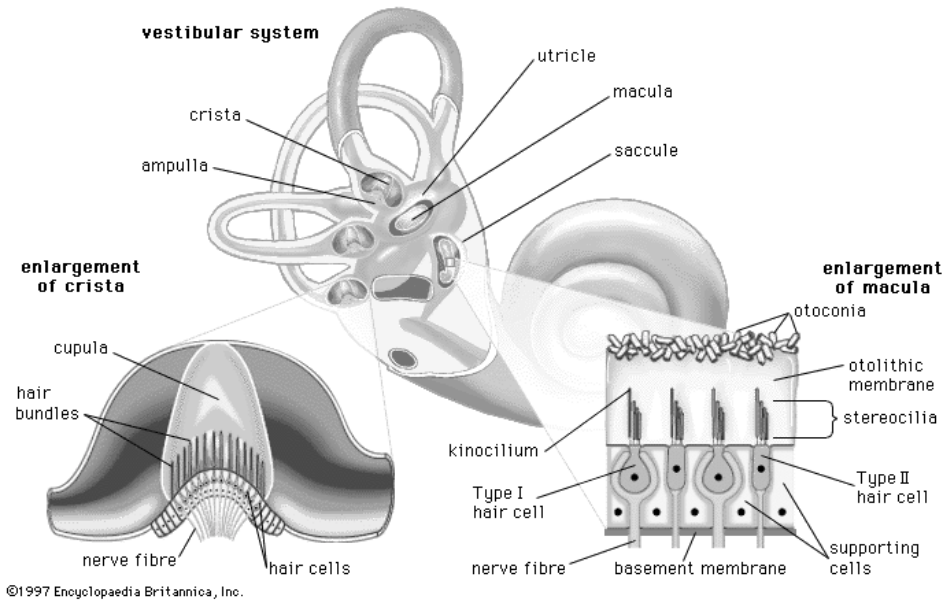


Figure 1.2: Human Vestibular System (taken from [Encyclopaedia Britannica, 2001]).

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 1.2. 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 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 otholitic 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.

1.1.2 Human Vision and Vestibular System

In humans, the retinal image is stabilized 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 stabilization. Holding the extended fingers half a metre 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 stabilization 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 stabilization 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 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 stabilization, 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 manovering fighter airplane, the resulting spatial disorientation is difficult or impossible to correct by higher-level neural processing.

1.1.3 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. The lowest reported threshold for perception of rotation is 0.035 $deg.s^{-2}$, but this degree of sensitivity is obtained only with continuous angular acceleration and long response latencies. Other observations put the perceptual threshold between roughly 0.1 and 2.0 $deg.s^{-2}$. Reasonable values are 0.14, 0.5 and 0.5 $deg.s^{-2}$ for yaw, roll, and pitch motions, respectively. Another way to describe the threshold for rotation of the vestibular system is in terms of the minimum angular acceleration-time product. This product, known as Mulder's constant, remains fairly constant for stimulus times of up to 5 seconds. A reasonable value for Mulder's constant is 2 $deg.s^{-1}$. The perceptual threshold of the otolith organs can be given as a minimum angle for which the change in the gravity force is perceived, or as a linear acceleration. A 1.5 *deg* change in direction of applied gravity force is perceptible under ideal conditions. The minimum perceptible intensity of linear acceleration has been reported by various authors to be between 0.001 and 0.03 g, depending on the direction of the acceleration and the experimental method used. Values of 0.01 g for vertical and 0.006 g for horizontal acceleration are appropriate representative thresholds. These are valid for sustained and relatively low frequency stimulus. As with rotation, the thresholds for linear accelerations applied for less than about 5 seconds is a more or less constant acceleration-time product of about 0.3 to 0.4 $m.s^{-1}$.

The currently available low cost inertial sensors are capable of similar performances. An overview these sensors is given in chapter 2. The inertial system prototype built for this work, using low cost sensors, has gyros with 0.1 $deg.s^{-1}$ resolution, and accelerometers with 0.005 g resolution. Notice that the gyros measure angular velocity and not angular acceleration.

1.2 This Work

In this work I explore the integration of inertial sensor data in artificial vision systems. Inertial sensors coupled to the vision system can provide valuable information to aid the image processing task. In humans and animals the vestibular system plays a similar role. This integration enables the solution to some problems which are complex, if not impossible, based on image processing alone. Visual and inertial sensing are two sensory modalities that can be explored to give robust solutions on image segmentation and recovering of 3D structure from images, increasing the capabilities of autonomous vehicles and enlarging the application potential of vision systems.

After reviewing the currently available inertial sensors, I built an inertial system prototype based on low cost sensors. This prototype was coupled to a stereo vision system. The inertial sensors can provide valuable data about the vision system's attitude relative the world by keeping track of the gravity vector. From the image data and inertial sensors I can recover the horizon, vertical, camera focal distance and pose, and ground plane orientation.

I begin by formalizing the underlying principles of combining and fusing the two sensing modalities. The unit sphere model I used provides a good framework for integration of scene recovered projective data and inertial references. The cameras focal distance is a key factor for scene metric perception and reconstruction. I developed a method for the estimation of camera focal distance based on vanishing points and inertial sensors. In an image, the horizon can be found by having two distinct vanishing points of levelled plane lines. With a suitable calibration target (e.g., a levelled square with well defined edges) the horizon can be determined. Knowing the vertical in the camera's referential and the focal distance, an artificial horizon can be traced. I show that using just one vanishing point, obtained from two parallel lines belonging to some levelled plane, and using the camera's attitude taken from the inertial sensors, the unknown scaling factor f in the camera's perspective projection can be estimated. The quality of the estimation of f depends on the numerical stability of the computed vanishing point, and the noise level in the accelerometer data. Nevertheless it provides a reasonable estimate for a completely uncalibrated camera. The advantage over using two vanishing points is that the

best (*i.e.*, more stable) vanishing point can be chosen, and that in indoors environment the vanishing point point can sometimes be obtained from the scene without placing any specific calibration target.

By performing visual fixation of a ground plane point, the system height can be estimated and the ground plane completely determined. The collienation, or homography, of image ground points on the left and right cameras can than be found. To segment the ground plane, image feature points below the horizon are parsed, and matches tested using the homography of image ground points. The identified ground plane points can be used to define a patch of ground plane along which the robot can move. By keeping track of the vertical in the stereo image pair, vertical lines can be detected. Using the homography of ground points for the *foot* of each segment, the lines can be matched and recovered in 3D, providing valuable visual cues in man made environments. In order to have a suitable world referential, a scene vanishing point can be used together with the inertial vertical reference. The mobile robot can than map the detected vertical features in a world map as it moves.

1.3 Related Work

Since I am exploring the integration of inertial sensors in artificial vision systems for robotic applications, it is interesting to see work that has previously used inertial sensors in robotics and computer vision.

Inertial sensors have typically been used for navigation in aerospace and naval applications [Collinson, 1996] [Pitman, 1962]. The electronic and silicon micromachining development, pushed by the needs of the automotive industry, brought about low cost, batch fabricated, silicon sensors. New micromachined sensors are being developed, aiming at having a single chip inertial system to be integrated in inertial aided GPS navigation systems [Allen et al., 1998]. This development has enabled many new applications for inertial sensors, namely in robotics and computer vision.

One application that has matured in consumer products is camera vibration compensation. Video camera image stabilization can be done using camera motion detection

followed by image correction [Luther, 1998]. Image processing over multiple frames can provide motion vectors, or image optical flow, from which camera motion can be inferred, but better results are achieved when the camera motion is directly detected using inertial sensors. The image correction can be done using a larger CCD and cropping it to the final output. For larger motions mechanical camera stabilization can be used, which might be just a simple suspension mechanism, or might involve inertial sensors in an active system for improved performance.

While simple tilt sensors had been used in many robotic applications, full inertial navigation systems were too expensive for most applications.

Viéville and Faugeras proposed the use of an inertial system based on low cost sensors for mobile robots [Viéville and Faugeras, 1989] and studied the cooperation of the inertial and visual systems in mobile robot navigation by using the vertical cue taken from the inertial sensors [Viéville and Faugeras, 1990] [Viéville et al., 1993b] [Viéville et al., 1993a]. In Viéville's book [Viéville, 1997] he describes the work done on the integration of inertial sensor data in vision systems to rectify images and improve self-motion estimation for 3D structure reconstruction.

An inertial sensor integrated optical flow technique was proposed by Bhanu *et al.* [Bhanu et al., 1990]. Inertial sensors were used to improve optical flow for obstacle detection. The inertial system was used to compensate ego motion of the vehicle, improving interest point selection, matching of the interest points, and the subsequent motion detection, tracking, and obstacle detection.

Vaganay, Aldon *et al.* used low cost inertial sensors to estimate the attitude of a mobile robot [Vaganay et al., 1993]. They fused the accelerometer and gyro data with an estended Kalman filter to obtain a better estimate for roll and pitch. Yaw was not estimated since, in close to planar conditions, the accelerometers cannot provide an estimate for yaw. Their technique relied on the good long term average attitude given by the accelerometers and the good short time attitude given by the gyros, overcoming the noisy instantaneous acceletometer data due to vibrations and vehicle movement, and the gyro drift due to integration over time.

Barshan and Durrant-Whyte developed a mobile robot navigation system using low cost inertial sensors [Barshan and Durrant-Whyte, 1994b] [Barshan and Durrant-Whyte,

1994a] [Barshan and Durrant-Whyte, 1995] [Nebot and Durrant-Whyte, 1992]. Error models for the sensors were estimated and used in an extended Kalman filter approach. Their experiments revealed that the system provided useful short-range data, but needed frequent periodic resets from some absolute positioning system.

Panerai and Sandini used a low cost gyroscope for gaze stabilization of a rotating camera, and compared the camera rotation estimate given by image optical flow with the gyro output [Panerai and Sandini, 1997] [Panerai and Sandini, 1998]. They also studied the integration of inertial and visual information in binocular vision systems [Panerai et al., 2000] and have built an artificial vestibular system using inertial sensors similar to the inertial system prototype built in my work.

More recently Kurazume and Hirose have used inertial sensors for image stabilization of remote legged robots and attitude estimation [Kurazume and Hirose, 2000].

A very interesting work is being developed by Coorg and his colleagues at MIT Media Lab [Coorg, 1998]. The purpose of the project is to automate the creation of virtual reality models of entire cities. By using *pose imagery*, *i.e.*, images with know orientation and position obtained by inertial sensors, they use mosaicing and other techniques to perform an automated three-dimensional modelling of the urban environment.

1.4 Overview of the Thesis

I will firstly present an overview of the current inertial sensor technology, focusing on low cost sensors suitable for robotic applications, and take a brief look at inertial navigation systems and the involved calculations, as well as some of the simplifications that can be made when considering mobile robotic systems. Chapter 3 introduces some computer vision background and the notation used. The next chapter presents the experimental setup used, namely the inertial sensors, data acquisition system, vision system and the computer system. The system's geometric model is described, as well as the method of estimating the systems attitude with the inertial sensors. Chapter 5 presents a method for camera focal distance calibration, using the inertial data and one vanishing point. Using the vanishing point and the known geometry, camera pose is also recovered. The next

chapter presents some world feature detection results. The first is the segmentation of the ground plane in the stereo image pair, using the inertial data and one initial fixated ground point. The second result is the detection of vertical lines in the stereo image pair, and 3D mapping of the features, using the vertical tracked by the inertial sensors and the ground plane homography.

The results are discussed in the last chapter, drawing some conclusions and proposing future work.

Chapter 2

Inertial Sensor Technology and Systems

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 gravity vector. In the following sections the current inertial sensor technology and underlying principles, as well as some of these sensors will be presented, focusing on low cost sensors suitable for robotic applications. [Slater, 1962] and [Kuritsky and Goldstein, 1990] present some of the older inertial sensor technology, and [Lawrence, 1998] an overview of modern inertial sensor technology. An updated list of manufacturers and companies selling inertial sensors can by found at my web site [Lobo, 2001].

2.1 Accelerometers

2.1.1 Accelerometer Basics

A basic accelerometer may be conceived as a mass-spring system as shown in figure 2.1. A proof mass is suspended by an elastic spring (*i.e.*, obeying Hooke's law), the damper is included to control ringing. Upon acceleration of the base frame the spring must supply

a force to make the proof mass keep up, and spring deflection is taken as measure of acceleration. The device is thus a force-measuring instruments which solves the equation

$$\boldsymbol{F} = m\boldsymbol{a} \tag{2.1}$$

where $\boldsymbol{F}$ is the force, m is the mass and $\boldsymbol{a}$ acceleration of the sensor, including gravity.

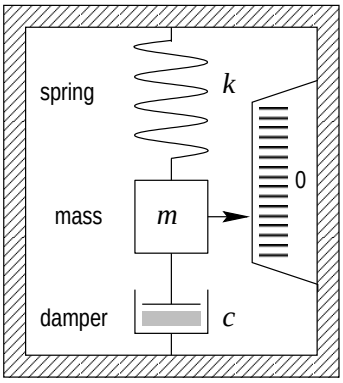


Figure 2.1: Basic accelerometer.

This damped mass-spring system with applied force constitutes a classical second order mechanical system. Considering the forces acting upon the proof mass to produce acceleration $\ddot{\boldsymbol{x}}$ we have

$$\boldsymbol{F}_{applied} - \boldsymbol{F}_{damping} - \boldsymbol{F}_{spring} = m\,\ddot{\boldsymbol{x}} \tag{2.2}$$

$$\boldsymbol{F}_{applied} = m\,\ddot{\boldsymbol{x}} + c\,\dot{\boldsymbol{x}} + k\boldsymbol{x} \tag{2.3}$$

where c is the damping coefficient, k the spring constant and $\boldsymbol{x}$ the proof mass displacement relative to the sensor frame and $\dot{\boldsymbol{x}}$ denotes the first derivative over time, and $\ddot{\boldsymbol{x}}$ the second derivative over time. If we consider the applied acceleration $\boldsymbol{a}$ as the system's input and the proof mass displacement $\boldsymbol{x}$ as the output, we can rearrange 2.3 to obtain the transfer function

$$\boldsymbol{F}_{applied} = m\boldsymbol{a} = m\,\ddot{\boldsymbol{x}} + c\,\dot{\boldsymbol{x}} + k\boldsymbol{x} \tag{2.4}$$

$$\boldsymbol{a} = \ddot{\boldsymbol{x}} + \frac{c}{m}\,\dot{\boldsymbol{x}} + \frac{k}{m}\boldsymbol{x} \tag{2.5}$$

Assuming initial rest conditions, with $\boldsymbol{x}_0 = 0$ and $\dot{\boldsymbol{x}}_0 = 0$, manipulation of the Laplace transformation of 2.5 gives the mechanical transfer function

$$G_m(s) = \frac{\boldsymbol{x}(s)}{\ddot{\boldsymbol{x}}(s)} = S_m \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{2.6}$$

where $\omega_n = \sqrt{k/m}$ is the resonance frequency of the structure, $\xi = \frac{1}{2}c\sqrt{k/m}$ the damping ratio, and $S_m = m/k$ the mechanical sensitivity.

The system's response will be determined by amount of damping, ξ. If under-damped, overshooting and oscillation will occur. The system will have the shortest rise time without overshoot when it is critically damped. When the system is over-damped there will be no overshoot but the rise time will be slow [Ausman, 1962].

Practical accelerometers vary in design and technology, but all mechanise the equation $\boldsymbol{F} = m\boldsymbol{a}$ in some way. They can be electromagnetic, vibrating string, gyro-pendulum, optical, piezoresistive, piezoelectric, capacitive, amongst others.

2.1.2 Silicon Accelerometers

In recent years, micromachined accelerometers have become widely available, largely due to the ability to produce them at a low cost. The needs of the automotive industry, namely for airbag deployment systems, encouraged silicon sensor development, enabling the batch-fabrication of integrated accelerometer sensors. The current commercially available accelerometers are developed in silicon and incorporate amplification, signal conditioning and temperature compensation.

There are presently three main types of micromachined low cost accelerometers. These are the capacitive, piezoelectric and piezoresistive types. The piezoelectric sensors have 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. These changes are used to generate a signal proportional to the acceleration applied to the sensor. Some are open loop sensors, others have a force balancing feedback loop that keeps the sensing element at its central position, gaining better response and improved linearity. These devices are built so as to have a sensing axis and reduced off-axis sensitivity. These sensors present a wide range of measurement ranges from $\pm\, 2\ g$ up to $\pm\, 500\ g$.

Typical applications of such devices in the automotive industry include frontal impact airbag systems, suspension control, braking control and crash testing. They also find applications in industrial vibration monitoring, transportation shock monitoring, motion control. This big market will push the development of the technology further and better and lower cost sensors are to be expected. New micromachined sensors are being developed, aiming at having a single chip inertial system to be integrated in inertial aided GPS navigation systems [Allen et al., 1998].

A silicon accelerometer typically has a silicon spring and a silicon mass. In open loop configurations the acceleration is computed by measuring the displacement of the mass. Typical errors include: non-linearity of the spring; off-axis sensitivity; hysteresis due to the springs or hinges; rotation-induced errors (*i.e.*, when body rotation adds rotational acceleration to the linear acceleration we intend to measure); and accelerometer signal noise.

For higher precision, force balancing closed loop configurations are implemented. Forces are applied to the mass to make it track the frame motion perfectly, and thus zero-balance the mass. Typical restoring forces used in silicon accelerometers include magnetic, piezoelectric and electrostatic. The sensor output will be given by the amount of force necessary to zero-balance the mass. By zero-balancing the mass, errors due to distortions and spring non-linearity are minimised. The input dynamic range and bandwidth is increased. Weaker hinges can be used, reducing hysteresis effects, and mechanical fatigue is minimised. No damping fluid is required, allowing operation in vacuum, and mechanical resonance avoided. Improved precision is thus accomplished.

In order to sense the proof mass displacement, either to directly give the output signal or control the zero-balancing loop, a number of sensing techniques are available. These include piezoresistive, piezoelectric, capacitive and optical. The piezoelectric accelerom-

eters rely on the deposition of a piezoelectric layer onto the silicon springs. They have a high output at relatively low current, but have high impedance and no DC response. Optical silicon accelerometers rely on the changing characteristics of an optical cavity, due to mass displacement. Infra-red light (IR) penetrating the cavity is bandpass dependent of the mass displacement. This technology has been used in high-resolution, but rather high cost, pressure sensors [Allen et al., 1989]. Piezoresistive and capacitive both have DC response and relatively low cost, making them suitable for low-grade inertial navigation systems. In the following sections these sensors are described in more detail.

Tables 2.1 and 2.2 present a comparative chart of some of the commercially available accelerometer sensors described. It is by no means complete, due to the vast number of sensors available. An updated list of manufacturers and companies selling inertial sensors, as well as sensor technical specifications, can by found at my web site [Lobo, 2001].

Piezoresistive Accelerometers

The piezoresistive accelerometers rely on the deposition of a piezoresistive layer onto the silicon springs. Changes in resistance provide a measure of proof mass displacement. The first silicon accelerometer prototype was built in 1976 [Allen et al., 1989]. This device had a single cantilever structure, was fragile, and had to be damped with a liquid. Despite its limitations, it represented a significant step from the attachment of silicon strain sensors onto metal diaphragms to having the resistor diffused onto single-crystal silicon. The basic design structures that have evolved are shown in figure 2.2.

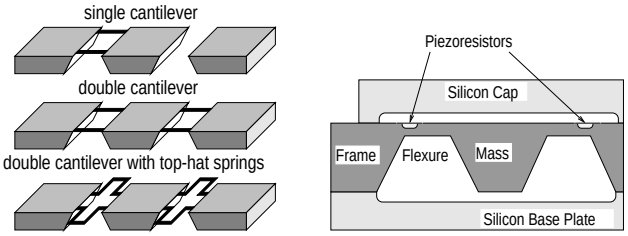


Figure 2.2: Design structures for the piezoresistive accelerometer and cross-section of double cantilever sensor (adapted from [ICSensors, 1988]).

The single cantilever has, in theory, the highest sensitivity, but has more off-axis errors,

sensor	manufactor	obs	technology	range	axis	resolution	bandwidth	sensitivity	linearity error	transverse sensitivity	output noise	noise density
ADXL202	Analog Devices	low power digital output	capacitive	$\pm 2\ g$	X,Y	$5\ mg$	$6\ kHz$					
ADXL105	Analog Devices	precision $\pm 5g$ replacement for XL05	capacitive	$\pm 5\ g$	X	$2\ mg$	$10\ kHz$					
ADXL210	Analog Devices	low power digital output	capacitive	$\pm 10\ g$	X,Y	$10\ mg$	$6\ kHz$					
ADXL250	Analog Devices	high resolution	capacitive	$\pm 50\ g$	X,Y	$10\ mg$	$1\ kHz$					
ADXL150	Analog Devices	high resolution replacement for XL50	capacitive	$\pm 50\ g$	X	$10\ mg$	$1\ kHz$					
ADXL190	Analog Devices	(pre-release)	capacitive	$\pm 100\ g$	X	$40\ mg$	$400\ Hz$					
ADXL50	Analog Devices	(undergoing obsolescence)	capacitive	$\pm 50\ g$	X	$50\ mg$	$4\ kHz$					
ADXL05	Analog Devices	(obsolete)	capacitive	$\pm 5\ g$	X	$5\ mg$	$6\ kHz$					
CXL04M1 CXL04M1Z CXL04M3	CrossBow	packaged sensor general purpose factory calibrated	capacitive	$\pm 4\ g$	X Z X,Y,Z		$\mathrm{DC} - 100\ Hz$	$500\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 3.5\ \%\ FS$	$5\ mg\ rms$	$500\ \mu g/\sqrt{Hz}$
CXL25M1 CXL25M1Z CXL25M3	CrossBow	packaged sensor general purpose factory calibrated	capacitive	$\pm 25\ g$	X Z X,Y,Z		$\mathrm{DC} - 100\ Hz$	$80\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 3.5\ \%\ FS$	$50\ mg\ rms$	$5000\ \mu g/\sqrt{Hz}$
CXL10LP1 CXL10LP1Z CXL10LP3	CrossBow	packaged sensor general purpose low power factory calibrated	capacitive	$\pm 10\ g$	X Z X,Y,Z		$\mathrm{DC} - 100\ Hz$	$200\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 3.5\ \%\ FS$	$10\ mg\ rms$	
CXL25LP1 CXL25LP1Z CXL25LP3	CrossBow	packaged sensor general purpose low power factory calibrated	capacitive	$\pm 25\ g$	X Z X,Y,Z		$\mathrm{DC} - 100\ Hz$	$80\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 3.5\ \%\ FS$	$10\ mg\ rms$	
CXL50LP1 CXL50LP1Z CXL50LP3	CrossBow	packaged sensor general purpose low power factory calibrated	capacitive	$\pm 50\ g$	X Z X,Y,Z		$\mathrm{DC} - 100\ Hz$	$40\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 3.5\ \%\ FS$	$10\ mg\ rms$	
CXL100LP1Z	CrossBow	packaged sensor general purpose low power factory calibrated	capacitive	$\pm 100\ g$	Z		$\mathrm{DC} - 400\ Hz$	$20\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 3.5\ \%\ FS$	$80\ mg\ rms$	
CXL01LF1 CXL01LF1Z CXL01LF3	CrossBow	packaged sensor precision high stability factory calibrated	capacitive	$\pm 1\ g$	X Z X,Y,Z		$\mathrm{DC} - 125\ Hz$	$2\ V.g^{-1}$	$\pm 0.3\ \%\ FS$	$\pm 3.5\ \%\ FS$	$1.2\ mg\ rms$	$100\ \mu g/\sqrt{Hz}$
CXL02LF1 CXL02LF1Z CXL02LF3	CrossBow	packaged sensor precision high stability factory calibrated	capacitive	$\pm 2\ g$	X Z X,Y,Z		$\mathrm{DC} - 125\ Hz$	$1\ V.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 3.5\ \%\ FS$	$1.5\ mg\ rms$	$130\ \mu g/\sqrt{Hz}$

Table 2.1: Technical specifications of some commercial accelerometer sensors

sensor	manufactor	obs	technology	range	axis	resolution	bandwidth	sensitivity	linearity error	transverse sensitivity	output noise	noise density
CXL25TA1Z	CrossBow	packaged sensor instrumentation amplified output factory calibrated		$\pm 25\ g$	Z		DC $- 1000\ Hz$	$80\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 1\ \%\ FS$	$20\ mV\ pp$	
CXL50TA1Z	CrossBow	packaged sensor instrumentation amplified output factory calibrated		$\pm 50\ g$	Z		DC $- 1000\ Hz$	$40\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 1\ \%\ FS$	$10\ mV\ pp$	
CXL50TB1Z	CrossBow	packaged sensor instrumentation low noise unamplified bridge output factory calibrated	piezoresistive	$\pm 50\ g$	Z		DC $- 1000\ Hz$	$10\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 1\ \%\ FS$	$20\ \mu V\ rms$	
CXL100TB1Z	CrossBow	packaged sensor instrumentation low noise unamplified bridge output factory calibrated	piezoresistive	$\pm 100\ g$	Z		DC $- 1000\ Hz$	$5\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 1\ \%\ FS$	$10\ \mu V\ rms$	
23200A	Summit Instruments	packaged sensor signal conditioning temperature sensor factory calibrated	capacitive	$\pm 10\ g$ to $\pm 75\ g$	X,Y		DC $- 5000\ Hz$	$8\ mV.g^{-1}$ to $60\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 0.25\ \%$	$66\ mg\ rms$	$6.6\ mg/\sqrt{Hz}$
23203A	Summit Instruments	packaged sensor signal conditioning temperature sensor factory calibrated	capacitive	$\pm 1\ g$ to $\pm 7.5\ g$	X,Y		DC $- 2500\ Hz$	$84\ mV.g^{-1}$ to $630\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 0.25\ \%$	$5\ mg\ rms$	$1\ mg/\sqrt{Hz}$
34100A	Summit Instruments	packaged sensor signal conditioning Precision Aligned factory calibrated	capacitive	$\pm 10\ g$ to $\pm 75\ g$	X,Y,Z		DC $- 5000\ Hz$	$8\ mV.g^{-1}$ to $60\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 0.25\ \%$	$66\ mg\ rms$	$6.6\ mg/\sqrt{Hz}$
34103A	Summit Instruments	packaged sensor signal conditioning Precision Aligned factory calibrated	capacitive	$\pm 1\ g$ to $\pm 7.5\ g$	X,Y,Z		DC $- 2500\ Hz$	$84\ mV.g^{-1}$ to $630\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 0.25\ \%$	$5\ mg\ rms$	$1\ mg/\sqrt{Hz}$
C3A-02	BASE & SPP	packaged sensor signal conditioning low noise factory calibrated	capacitive	$\pm 2\ g$	X,Y,Z	$1\ mg$	DC $- 10\ Hz$		X,Y $\pm 0.5\ \%\ FS$ Z $\pm 1.5\ \%\ FS$	$3\ \%$	$1\ mg\ rms$	
3031	EG&G IC Sensors	OEM sensor low cost surface mount package	piezoresistive	$\pm 2\ g$ to $\pm 500\ g$	Z		DC $- 350\ Hz$ to DC $- 2400\ Hz$	$2.4\ mV.g^{-1}$ to $0.1\ mV.g^{-1}$	$\pm 0.2\ \%\ FS$	$\pm 1\ \%\ FS$	$1\ \mu V\ pp$	

Table 2.2: Technical specifications of some commercial accelerometer sensors

and is rather fragile. The double cantilever provides good off-axis cancellation, and is more robust. The folded springs of the top-hat configuration allow for large displacements in a smaller chip area, thus reducing the cost of the sensor.

An example of a double cantilever silicon accelerometer is the 3145 model from ICSensors [ICSensors, 2001]. This integrated accelerometer incorporates amplification, signal conditioning, temperature compensation and internal voltage regulation, all in a small package. The sensor is available with ranges from ± 2 g up to ± 200 g. Piezoresistors are diffused onto the flexures of the bridge supporting the sensing mass as seen in figure 2.2. The resultant electrical structure is a Wheatstone bridge where two resistors increase with downward acceleration and two decrease with the same force. This provides a true DC response, unlike the piezoelectric accelerometers that only have AC response. The full bridge design substantially reduces off-axis sensitivity. Since sideways accelerations cause symmetric disturbances on each side of the double cantilever structure, their effect is cancelled out, provided the piezoresistors are well matched [ICSensors, 1988]. Some sensors include an electrode on the cap to provide a means to electrostatically deflect the mass for a self-test function. ICSensors has a wide range of accelerometers based on the same technology including a two chip sensor with signal conditioning that is surface mounted directly to the circuit board [ICSensors, 1995].

Many manufacturers, such as Lucas [Lucas, 2001] and Entran Devices [Entran, 2001], amongst others also make piezoresistive accelerometers with the same technology.

Capacitive Accelerometers

In capacitive accelerometers, proof mass displacement alters the geometry of capacitive plates. An output signal is derived from the changing capacitance.

One design of capacitive silicon accelerometers uses a main beam that constitutes the proof mass, with springs at each end. The beam has multiple centre plates at right angles to the main beam that interleave with fixed plates attached to the frame on each side, forming a comb-like symmetric structure. This design allows sensing of positive and negative acceleration along the axis of the main beam in the sensor plane.

Each of the centre plates fits between two adjacent fixed plates, forming a capacitive

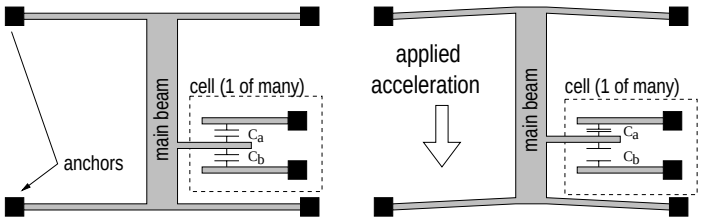


Figure 2.3: Capacitive comb finger array accelerometer working principle (adapted from [Summit Instruments, 1994]).

divider, as shown in figure 2.3. The two fixed plates are driven with an equal amplitude but opposite polarity square wave signals, typically 1 *MHz*.

With no acceleration, the two capacitances are approximately equal and the centre plate will be at approximately zero volts. Any applied acceleration causes a mismatch in plate separation which results in greater capacitive coupling from the closer fixed plate; a voltage output can thus be detected on the centre plate. The acceleration signal is contained in the phase relative to the driving signal, and a synchronous demodulator technique is used to extract the relatively low frequency acceleration signal.

The resulting acceleration signal is used in a feedback loop to force balance the sensor, impeding the deflection and servoing the sensor back to its zero acceleration position. The balancing force is obtained electrostatically, caused by driving the centre plates to a voltage proportional to the acceleration signal. The force balancing servo loop response has to be fast enough and flat enough to track fast level changes, keeping the sensor nearly motionless, minimising the errors.

One example of a capacitive micromachined accelerometer is the ADXL05 from Analog Devices [Analog Devices, 2001], a $\pm$ 5 g version of the ADXL50 $\pm$ 50 g accelerometer currently used in airbag deployment systems [Analog Devices, 1996]. The sensor's range and resolution of 0.005 g with a typical noise floor of $\pm 500\ \mu g/\sqrt{Hz}$ makes it suitable for robotic applications. The accelerometer is contained in a 10-pin TO100 metal case and consists of a sensor, an oscillator, a demodulator, pre-amplifier, voltage reference, and a buffer amplifier. The structure consists of a main beam with 46 centre plates as described above. The main beam however is only anchored at two points, being suspended by folded

springs at each end.

Many companies sell packaged sensors based on Analog Devices' accelerometers. The CXLM series of accelerometers from Crossbow [Crossbow, 2001] uses the ADXL05 sensor. Jewell Instruments [Jewell Instruments, 2001] has sensors based on the accelerometers from Analog Devices. The 34100 series of sensors from Summit Instruments [Summit Instruments, 2001] are based on the same accelerometers. These have three precision aligned capacitive accelerometers, with built-in conditioning and amplification, and are available with ranges from $\pm$ 1 g to $\pm$ 75 g. The complete sensor is a compact one inch cube.

The sensor employs three polysilicon surface micromachined sensors each capable of measuring positive and negative acceleration along one axis. Each sensor consists of a main beam tethered at 4 points with 42 centre plates at right angles to the main beam as described above [Summit Instruments, 1994].

Other designs have also been implemented, namely the C3A-02 model from British Aerospace Systems and Equipment working together with Sumitomo Precision Products [Base&SPP, 2001]. The three-axial accelerometer uses a single proof mass and capacitive sensing. This micromachined sensor is not flat like the ones described above, and consists of a proof mass with multiple capacitive sensing elements, as shown in figure 2.4.

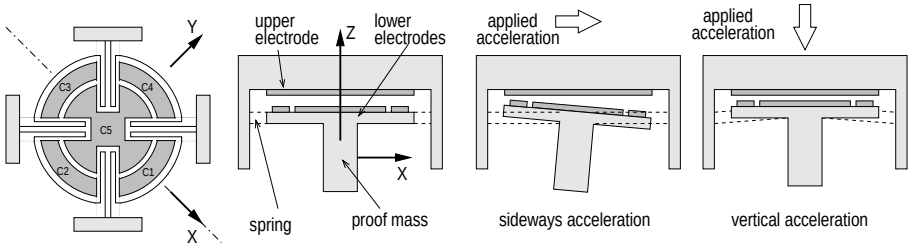


Figure 2.4: Diagram of BASE C3A-02 sensing element (adapted from [Base&SPP, 1996]).

The difference between C1 and C3 capacitance values gives the x-axis acceleration, likewise C2 and C4 provide the y-axis acceleration and change in the capacitance value of C5 gives the z-axis acceleration [Base&SPP, 1996].

In [Lemkin et al., 1997] a single chip three-axis surface micromachined sigma-delta

capacitive accelerometer is presented. The sensor has three separate proof masses, each with its own feedback and interface circuitry, co-ordinated by an on-chip master clock. The x- and y-axis sense elements are implemented with a differential comb finger array as described above, but the z-axis sense element consists of a flat parallel plate capacitor.

2.2 Inclinometers

Inclinometers (also known as clinometers, tilt sensors or level sensors) measure the orientation of the gravity vector. When accelerations other than gravity are present, the sensors measure the resultant acceleration vector acting upon them. The tilt with respect to level ground can only be measured when the sensor is at rest or with constant velocity. The AccuStar electronic capacitive inclinometer, from Lucas [Lucas, 2001], is an example of such a device. This tilt sensor has a linear range of $\pm$ 45 *deg* with a resolution of 0.001 *deg*. Cross-axis inclinations up to 45 *deg* introduce an error of less than 1 %.

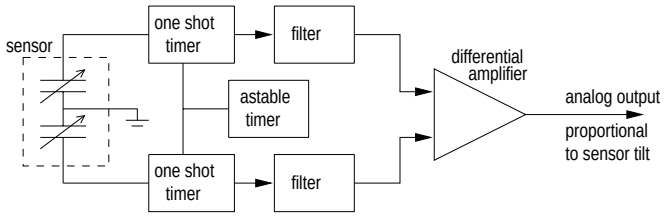


Figure 2.5: AccuStar inclinometer block diagram.

The concept of the sensor is based on a dielectric fluid, with an air bubble, inside a capacitive sensor. When you tilt the sensor, the bubble moves under the force of gravity, changing the capacitance of the sensor elements. The resulting differential generates an output signal which reflects the relative tilt in the sensing axis as shown in figure 2.5. Due to the fluids inertia and settling time, and sometimes the measurement method, inclinometers tend to have a delayed response. For this particular sensor the delay has a time constant of 0.3 *s*.

An interesting design is the dual axis inclinometer AccuStar II. The concept of the

sensor is based on a dielectric fluid with an air bubble inside a dome shaped capacitive sensor. The sensing dome is divided into four quadrants. When you tilt the sensor, the bubble changes the capacitance of the sensor elements in each quadrant. The resulting differential generates an output signal which reflects the relative tilt of the device in either x- or y-axis. The sensor has a linear range of $\pm$ 20 *deg* with a resolution of 0.01 *deg*.

Other designs, still using the principle of the spirit level, measure resistance to obtain the tilt. The Electrolevel-ELH46 inclinometer manufactured by TILT Measurement [Tilt Measurement , 2001] is an example of a resistive inclinometer. The sensor has a suitably curved tube, with an electrically conducting liquid and gas bubble inside, and three electrodes. When the sensor is tilted the bubble's position relative to the electrodes changes, causing a difference in the electrical resistance between electrodes proportional to the tilt. Cline Labs [Cline Labs, 2001] manufactures some inclinometers also based on resistance measurements with specs similar to the AccuStar.

When using inclinometers, care should be taken when accelerations other than gravity are present, since the tilt will be measured relative to the resultant vector. If the sensor is tilted by an angle α to the horizontal and is subject to an acceleration a in a direction normal to the sensor's measuring axis in the horizontal plane, the tilt sensor will not measure α. The measured angle will be

$$\alpha_{measured} = \alpha + \tan^{-1}\left(\frac{a}{g}\right) \tag{2.7}$$

where g is the modulus of the gravity vector.

2.3 Gyroscopes

The mechanical gyroscope [1], a well known and reliable rotation sensor, based on the inertial properties of a rapidly spinning rotor, has been around since the early 1800s. The spinning rotor or flywheel type of gyroscope uses the fundamental characteristic of the angular momentum of the rotor to resist changing its direction to either provide a spatial reference or to measure the rate of angular rotation [Collinson, 1996]. Many

[1] from the Greek word *gyros* meaning rotation and *skopein* meaning view.

different designs have been built, and different methods used to suspend the spinning wheel. [Slater, 1962] and [Lawrence, 1998] have some examples of such devices.

Optical gyroscopes measure angular rate of rotation by sensing the resulting difference in the transit times for laser light waves travelling around a closed path in opposite directions. Since light always travels at the same velocity, the difference in transit time is proportional to the input rotation rate, and the effect is known as the *Sagnac effect*, after the French physicist G. Sagnac. In fact Sagnac demonstrated that rotation rate could be sensed optically with the Sagnac interferometer as long ago as 1913 [Collinson, 1996].

Several significant engineering challenges had to be overcome, and it was not until after the invention of the laser in 1960 that the first practical optical gyroscopes were built [Everett, 1995]. The first ones were ring-laser gyroscopes and later came the optical fibre gyroscope. Ring-laser gyros require tight-tolerance machining and clean-room assembly. The telecommunication industry has made optical fibres increasingly available, enabling the construction of low-cost fibre optic gyroscopes. These devices, named FOG (Fiber Optic Gyroscope) or OFG (Optical Fibre Gyroscope) for short, use multiple loops of optical fibre to construct the closed loop path, and semiconductor laser diodes for the light source.

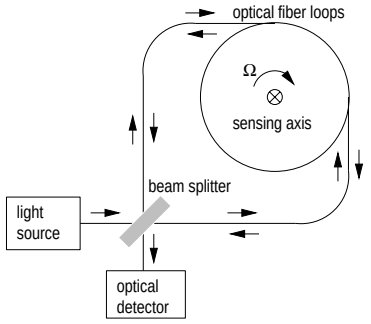


Figure 2.6: Simplified diagram of optical fibre gyroscope (adapted from [Komoriya and Oyama, 1994]).

A simplified diagram is shown in figure 2.6. The beam splitter divides the laser beam into two coherent components. The difference of travelling time between the two beams, caused by the difference in optical path lengths, is detected as the interference between

the two beams by an optical detector. Hitachi Cable [Hitachi Cable, 2001] pioneered the production of these sensors with its low-cost optical fibre gyroscopes. Crossbow [Crossbow, 2001], amongst others, incorporate optical fibre gyroscopes in their inertial measurement units.

But even lower cost, and becoming increasingly compact, are the vibrating structure gyroscopes. These use the Coriolis effect, whereby an object with linear motion in a rotating frame of reference, relative to inertial space, will experience a so called Coriolis acceleration given by

$$\boldsymbol{a}_{coriolis} = 2\boldsymbol{\varpi} \times \boldsymbol{v} \tag{2.8}$$

where $\boldsymbol{\varpi}$ is the angular velocity of the rotating frame and the object's velocity $\boldsymbol{v}$ is given in the rotating frame of reference.

Imagine a billiard ball rolling across a rotating table. An outside observer would see it moving along a straight line. But an observer on the table would see the ball following a non-linear trajectory, as if a mysterious force was driving it. This apparent force is called the Coriolis force. We can see from equation 2.8, that the Coriolis force will be perpendicular to both the rotation axis and the object's linear motion.

2.3.1 Vibrating Structure Gyroscopes

The basic principle of 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 cancelling out of unwanted accelerations.

The common house fly, in fact, uses a miniature vibrating structure gyro to control its flight. A pair of small stalks with a swelling at their ends constitute radially oscillating masses that will be subject to Coriolis forces when yaw is experienced. These forces will generate muscular signals that assist the acrobatic fly in its flight [Everett, 1995].

The Vibrating Prism Gyroscope

The Gyrostar ENV-011D, built by Murata [Murata Electronics, 2001], is a piezoelectric vibrating prism sensor for angular velocity. The device' s output is voltage proportional to the angular velocity. The principle of the sensor is outlined in figure 2.7.

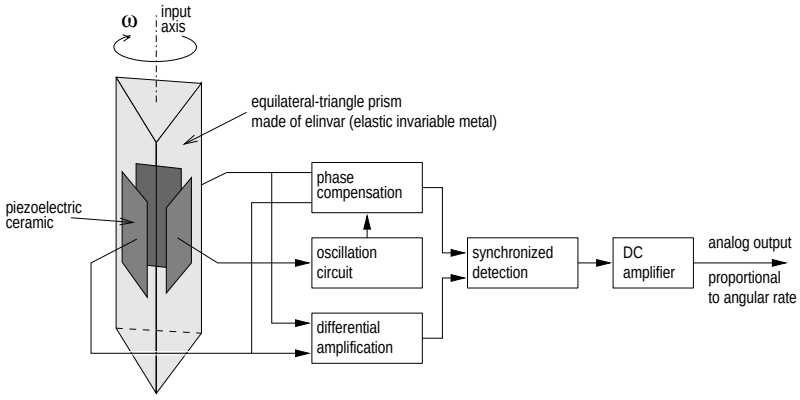


Figure 2.7: Piezoelectric vibrating prism gyroscope (adapted from [Murata, 1991]).

Inside the device there is an equilateral triangular prism, made from *elinvar* (elastic invariable metal), which is fixed at two points. Three piezoelectric ceramic elements are attached to the faces of prism, one on each side. The prism is forced to vibrate at about 7 kHz by two of the piezoelectric elements, whilst the other is used for feedback to the drive oscillator. These two elements are also used for detection. When there is no rotation they detect equally large signals. When the prism is turned, Coriolis forces will affect the prism vibration and the sensing piezoelectric elements will receive different signals. The difference between the signals is processed by the internal analogue circuits to provide an output voltage proportional to the angular velocity [Murata, 1991].

The Gyrostar ENV-011D has an angular velocity range of $\pm$ 90 $deg.s^{-1}$, is linear to within 0.5 % of maximum angular velocity, and has no hysteresis. The output scale factor is 22.2 $mV/deg.s^{-1}$ and the noise level is 10 mV rms maximum. Simpler models of the gyroscope built by Murata are known to have self-heating problems that cause offset drift.

The Tuning Fork Gyroscope

The GyroChip sensor built by Systron Donner Inertial Division [Systron Donner, 2001] uses a micro-miniature double-ended piezoelectric quartz tuning fork element to sense angular velocity. The sensor element and supporting structure are fabricated chemically from a single wafer of monocrystalline piezoelectric quartz.

The drive tines, being the active portion of the sensor, are driven by a high frequency oscillator circuit at a precise amplitude, producing the radial oscillation of the tines along the sensor plane, as shown in figure 2.8.

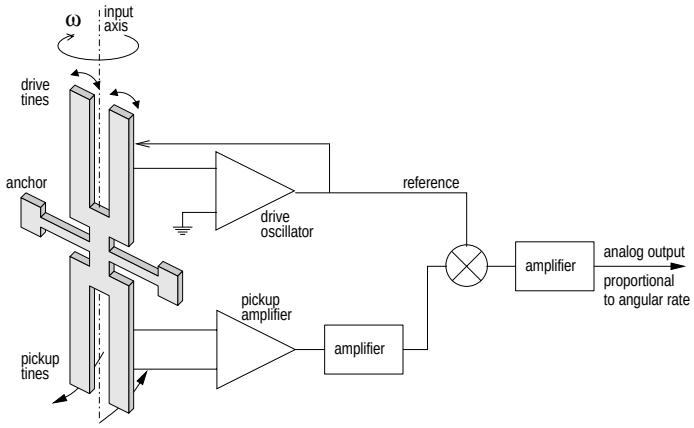


Figure 2.8: Systron Donner Inertial Division tuning fork gyroscope (adapted from [Systron Donner, 1995]).

A rotational motion about the sensor's longitudinal axis produces a DC voltage proportional to the rate of rotation due to the Coriolis forces acting on the sensing tines. Each tine will have a Coriolis force, $\boldsymbol{F}$, acting on it

$$\boldsymbol{F} = 2m\boldsymbol{\omega_i} \times \boldsymbol{v_r} \tag{2.9}$$

where m is the tine mass, $\boldsymbol{v_r}$ the instantaneous radial velocity and $\boldsymbol{\omega_i}$ the input rate. This force is perpendicular to both the input rate and the instantaneous radial velocity.

The two drive tines move in opposite directions, and the resultant Coriolis forces are normal to the plane of the fork assembly, and also in opposite directions. This produces

a torque which is proportional to the input rotational rate. Since the radial velocity is sinusoidal, the torque produced is also sinusoidal at the same frequency of the drive tines, and in-phase with the radial velocity of the tine.

The pickup tines respond to the oscillating torque by moving in and out of plane, producing a signal at the pickup amplifier. The sensed pickup signal is then synchronously demodulated to get the output signal proportional to the angular velocity along the sensor input axis.

Besides the GyroChip sensor and the enhanced GyroChip II, Systron Donner has other products using the same sensor, like the MontionPak that incorporates three miniature vibrating quartz gyros and three accelerometers in one package. The GyroChip II sensor has an angular velocity range of $\pm 100\ deg.s^{-1}$ and the output scale factor varies from 2.5 to 50 $mV/deg.s^{-1}$, depending on factory set range [Systron Donner, 1995].

Other Vibrating Structure Gyroscope Designs

Another interesting design of vibrating structure gyroscopes is the vibrating ring design used by British Aerospace Systems and Equipment working together with Sumitomo Precision Products [Base&SPP, 2001] in their angular rate sensors. The diagram in Figure 2.9 illustrates the working principle. The sensor uses a micromachined silicon ring placed over a permanent magnet. The ring is suspended by springs that allow it to vibrate and carry conducting wires to the ring.

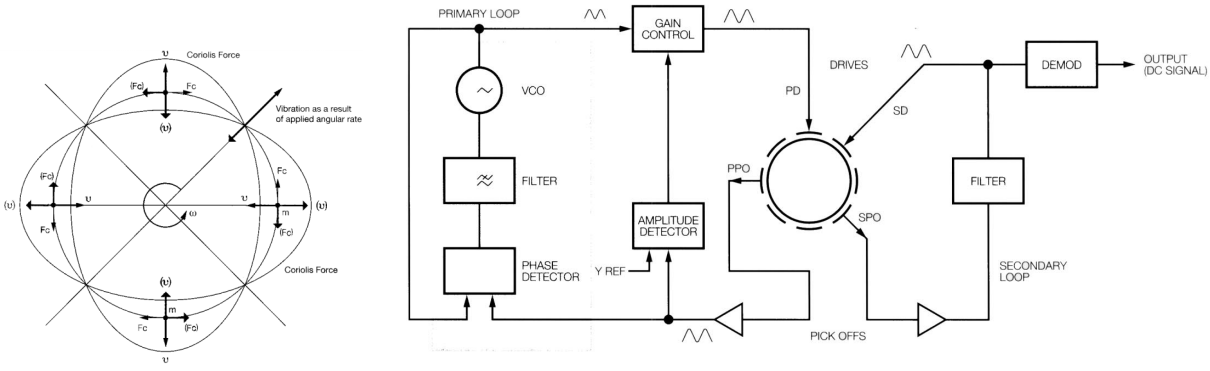


Figure 2.9: BASE and SPP vibrating ring gyroscope (adapted from [Base&SPP, 1996]).

By applying an alternating current to the conductors on the ring, the microminiature ring is made to vibrate. By conveniently placing the driving and sensing conductors on the ring, the sensing ones will be subject to movement due to Coriolis forces. The sensing conductors in the ring, moving in a magnetic filed, produce an output proportional to that movement. The amplitude of this movement is determined by the sensor rotation rate. To increase linearity and dynamic range of the sensor, a closed loop feedback nulling voltage is applied to the sensing conductors. The ring is forced to its *rest* position and the nulling voltage demodulated to obtain a DC voltage proportional to the rotation rate of the sensor around it's sensing axis [Base&SPP, 1996].

There are enumerate possible variations of the vibrating structure gyroscope and new micromachined gyros are being developed, [Clark et al., 1996] being an example.

2.4 Inertial Navigation Systems (INS)

Einstein's generalised relativity states 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 we can only determine an estimate for linear accelerations and angular velocities. Linear velocity and position, and angular position, can be obtained by integration [Pitman, 1962].

Inertial navigation systems (INS) implement this process of obtaining velocity and position information from inertial sensor measurements. The basic principle employed in inertial navigation is deduced reckoning, since the present location is determined by advancing some previous position through known course and velocity information over time. A set of three accelerometers is used to measure acceleration along three orthogonal axes, and their outputs are integrated twice to determine position. To compensate body rotation, three gyroscopes are used to measure rotation rates about three orthogonal axis. In gimballed systems the accelerometers are kept on a gyro-stabilised platform. In strap-down systems all sensors are rigidly fixed to the vehicle, and the gyro data is used to transform the accelerometer data to the navigation frame of reference. This can be

seen as computationally stabilised accelerometer platforms, as opposed to the physically stabilised platforms used in gimballed systems.

Early versions of INS were used by the Peenumüde group in Germany, during World War II, to guide the V2 rockets. This was the first primitive example of inertial guidance, relying on a gyro assembly to control the missile's attitude and an integrating accelerometer to sense accelerations along the thrust axis. The system was enabled during the powered ascent, and controlled the shut down of the rocket engines, after which the vehicle's trajectory became ballistic and therefore predictable. The design of complete systems , including gyro-stabilised accelerometers, integrators and computers for gravitational acceleration evolved in the late 1940's in the USA [Slater, 1962].

INS have since become widespread used in avionics and naval applications. But these systems are typically high-cost, since they require high-grade sensors to overcome the severe drift problems due to the double integration of acceleration measurements when estimating position. Although their cost has lowered due to technological developments, it is still rather high for robotic applications.

High-grade INS were usually gimballed systems, relying on expensive inertial grade mechanical components. With new sensor development and more computation power, strap-down systems are becoming more accurate and suitable for high-end applications. They provide high performance and reliability at a lower cost, consume less power and are more compact and lightweight [Collinson, 1996].

2.4.1 INS Calculations

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

$$\boldsymbol{\Delta}_{position} = \boldsymbol{x} = \int \dot{\boldsymbol{x}}\, dt = \iint \ddot{\boldsymbol{x}}\, dt = \int \boldsymbol{a}_{sensed} dt \tag{2.10}$$

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 stabilized platform. In strapdown systems the gyros measure the body rotation rate, and the sensed accelerations are computationally converted to the navigation frame of reference.

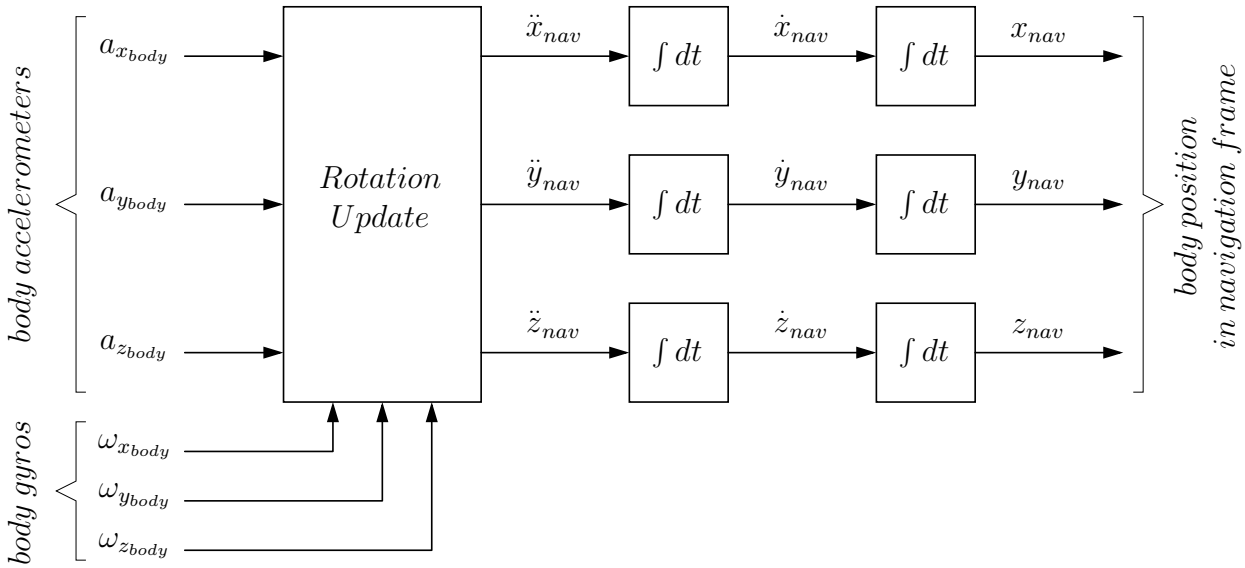


Figure 2.10: Rotation update in strapdown system mechanization

The mechanization of this rigid body angular motion has to account for the non-commutativity of finite rotations, mathematical singularities and numerical instability. Bortz [Bortz, 1971] presented a new orientation vector formulation suitable for strapdown inertial systems. Shuster [Shuster, 1993] discusses the various derivations for the rotation vector. The mechanization presented here, using quaternions, follows Savage's paper on strapdown inertial systems [Savage, 1984].

Quaternion algebra was developed by W.R.Hamilton in the nineteenth century as an extension of imaginary numbers to higher dimensions. A quaternion q can be written as

$$q = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k} = (q_0, \boldsymbol{v}) \tag{2.11}$$

where q_1, q_2 and q_3 are the components of the imaginary or vector part $\boldsymbol{v}$ of the quaternion,

$\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ 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 $\boldsymbol{i}$ in complex numbers, are all square roots of -1.
The magnitude of a quaternion is defined as

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{2.12}$$

The complex conjugate q^* of quaternion q is given by

$$q^* = q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k} = (q_0, -\boldsymbol{v}) \tag{2.13}$$

and the inverse q^{-1}

$$q^{-1} = \frac{1}{q} = \frac{q^*}{qq^*} \tag{2.14}$$

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

Vectors can by represented by purely imaginary quaternions. A point in space given by the vector $\boldsymbol{p}$ can be represented by the quaternion $p = (0, \boldsymbol{p})$. In my notation, when multiplying vectors with quaternions, I refer to the corresponding imaginary quaternion.

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

$$q = cos\frac{\phi}{2} + sin\frac{\phi}{2}\boldsymbol{u} \tag{2.15}$$

And the rotation update for a space point $\boldsymbol{p}$ is given by

$$\boldsymbol{p_{rotated}} = q\boldsymbol{p}q^{-1} = q\boldsymbol{p}q^* \tag{2.16}$$

If the quaternion $q\,(k)$ 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

$$\boldsymbol{a_{nav}} = q\,(k)\,\boldsymbol{a_{body}} q\,(k)^* \tag{2.17}$$

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

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathsf{T}} \tag{2.18}$$

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

$$\Delta q = cos\left(\frac{\omega \Delta t}{2}\right) - sin\left(\frac{\omega \Delta t}{2}\right) \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \tag{2.19}$$

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

$$q(k+1) = q(k)\,\Delta q \tag{2.20}$$

and using equation 2.17 the measured body accelerations are converted to the navigation frame and integrated to give body velocity and position in the navigation frame.

But the mechanization presented in figure 2.10 is only valid in the absence of a gravitational field. If the INS is being used in a ground vehicle or airplane, one has to take into account the earth's gravitational field. If we want the navigation frame of reference to be earth fixed, we have to take into account the fact that the earth is itself rotating. A complete mechanization of INS is given in [Savage, 1984], [Kelly, 1994] and [Collinson, 1996].

However, when considering ground robotic vehicles and small trajectories, some simplifications can be made. We can consider the local level ground plane as a reference and a local uniform gravity field. In this case, only the uniform gravity field has to be accounted for, by subtracting it from the sensed acceleration. By knowing the initial attitude and subsequent rotation updates, this can be done in the navigation frame of reference.

2.4.2 Mobile Robot INS

Recent development in accelerometers and gyroscopes has lead to some new low-cost sensors, as described in the previous section. Strap-down systems based on these low-cost inertial sensors offer low performance, namely in accumulated drift over time, making them unsuitable for high performance inertial systems, but can still be useful in some mobile robotic applications. The inertial system can be used to provide short-term accurate relative positioning, which combined with some other external reference absolute positioning system, to limit the INS absolute position drift error, will provide a suitable navigation system.

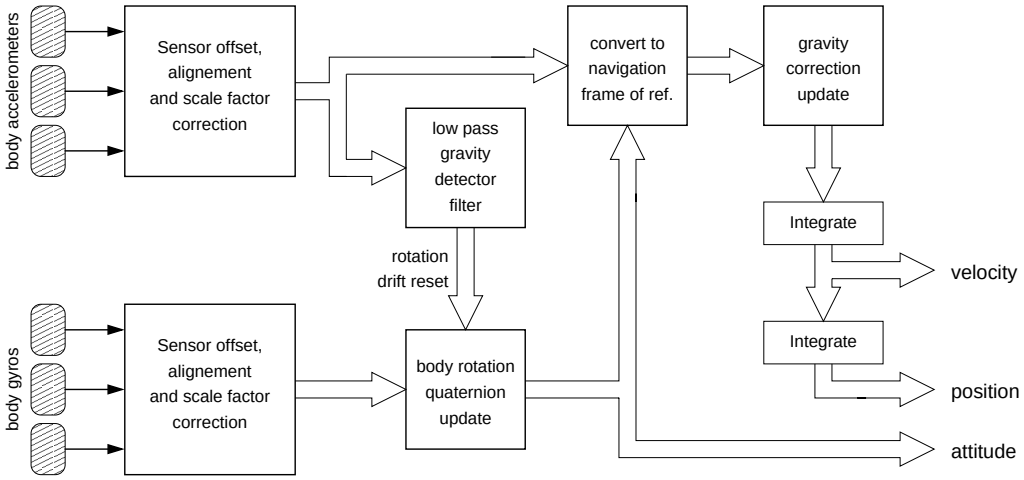


Figure 2.11: Simplified Strapdown Inertial Navigation System

To cope with the accumulated drift, some assumptions can be made on the vehicle'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 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 vehicle never accelerates or rotates below a certain value, preventing the error accumulation in the rotation update and position integration.

2.5 Summary

An overview was made of the currently available low-cost inertial sensors. The underlying principles and computations performed by INS were also described. The particular case of mobile robotic systems was considered, where some assumptions and simplifications can be made.

In this work I am only using the inertial sensors to provide a vertical reference, and not the full dynamic data and computed attitude and position from the INS. With the sensors used, only short term valid data could be obtained, due to the accumulated drift.

Chapter 3

Vision Notation and Background

To provide a background, upon which later results will be drawn, the following sections provide a summary introduction to some aspects of vision system geometry and image processing.

3.1 Camera and Image Geometry

Computer vision image acquisition has many aspects. The scene being viewed is influenced by lighting. The camera optics may introduce distortions or colour aberrations. Images constitute a spatial distribution of the irradiance at a plane, but the sensing element does a discrete sampling of this distribution. The sensing elements might be arranged in a rectangular grid, or in some other grid. Depending on the application and equipment being used, simplified models can be used.

3.1.1 Camera Model

The pinhole camera, or *camera obscura*, limits the incidence of light rays onto a surface, enabling the formation of an inverted image of the world. A small opening (the pinhole) is the centre of projection of the image, all rays of light forming the image pass through this point, and if it is small enough, a crisp but faint image is formed. To increase the image intensity a lens is used. A larger opening is used and the lens deflects the rays of

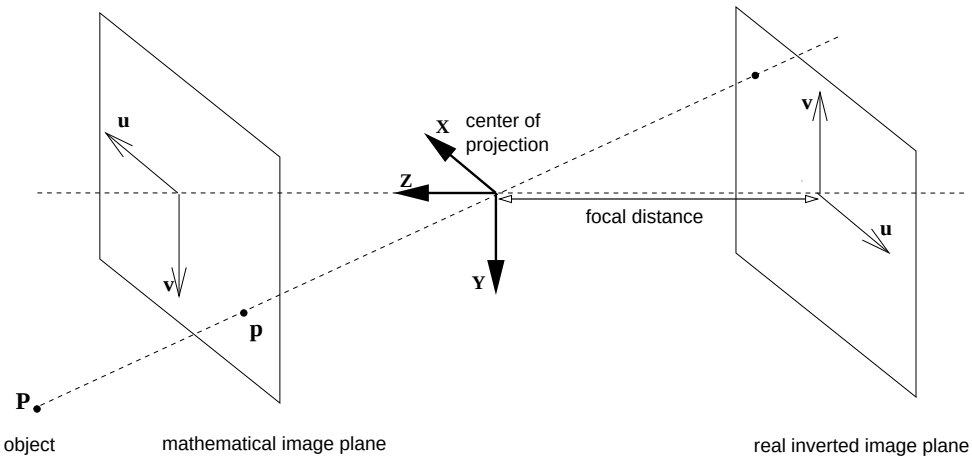


Figure 3.1: Camera perspective projection

light so that they form an inverted image at a particular distance. The distance from the image plane to the centre of projection is called the focal distance f. The optical axis is normal to the image plane and contains the centre of projection, defining the viewing direction of the camera. The image centre is the intersection of the optical axis with the image plane.

The images are therefore formed by mapping the 3D world onto a surface, usually a plane. The mapping will lose information about the depth of objects. Small objects near the camera appear to be the same size as larger objects placed further away from the camera. This mapping can be modelled by a perspective transformation, whereby each 3D point is projected onto the image plane. To avoid image inversion, the image plane can be considered to be in front of the centre of projection, forming a non-inverted mathematical image plane.

A projection point $\boldsymbol{p}_i = (u, v)^\mathsf{T}$ in the camera image is related with a 3D point $\boldsymbol{P} = (X, Y, Z)^\mathsf{T}$ by the perspective relations

$$u = S_u f \frac{X}{Z} + u_0 \qquad v = S_v f \frac{Y}{Z} + v_0 \tag{3.1}$$

where u and v are the pixel coordinates as show in figure 3.1, $(u_0, v_0)^\mathsf{T}$ the image centre, 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 $\boldsymbol{P}$ is expressed in the camera frame of reference. 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

$$u = f\frac{X}{Z} \qquad v = f\frac{Y}{Z} \tag{3.2}$$

where u and v are the pixel coordinates with origin at the image centre 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

$$s\boldsymbol{p}_i = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \boldsymbol{C}\begin{bmatrix} \boldsymbol{I} & 0 \end{bmatrix}\boldsymbol{P} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.3}$$

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

3.1.2 Projection onto Unit Sphere

The previous model derives from the cameras' geometry, but the projection need not be onto a plane. Consider a unit sphere around the optical centre, with the images being formed on its surface. The image plane can be seen as a plane tangent to a sphere of radius f concentric with the unit sphere. The image plane touches the sphere at point

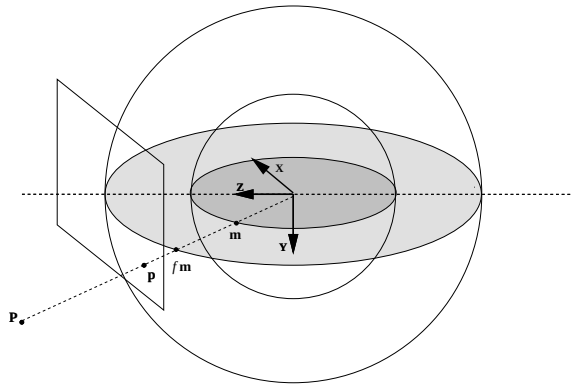


Figure 3.2: Projection onto Unit Sphere.

that defines the image centre and the sphere's equator. Using the unit sphere gives a more general model for central perspective, and has advantages when considering points at infinity.

Consider the unit sphere where every world point $\boldsymbol{P}$ in the world is projected, forming an image on its surface. The image points on this surface can be represented by unit vectors $\boldsymbol{m}$ placed at the sphere's centre, the optical centre of the camera.

$$\boldsymbol{P} \rightarrow \boldsymbol{m} = \frac{\boldsymbol{P}}{||\boldsymbol{P}||} \tag{3.4}$$

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

$$(u, v)^\mathsf{T} = \left(f\frac{m_1}{m_3}, f\frac{m_2}{m_3} \right)^\mathsf{T} \tag{3.5}$$

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.

For a given image point $\boldsymbol{p}_i = (u, v)^\mathsf{T}$, which expressed as a 3D point is $\boldsymbol{P}_i = (u, v, f)^\mathsf{T}$, the projection onto the unit sphere is given by

$$\boldsymbol{m} = \frac{1}{\sqrt{u^2+v^2+f^2}} \begin{bmatrix} u \\ v \\ f \end{bmatrix} = \pm N \left(\begin{bmatrix} u \\ v \\ f \end{bmatrix} \right) = \pm N\left(\boldsymbol{P}_i\right) = \pm N\left(\boldsymbol{P}\right) \tag{3.6}$$

where $N[\,]$ is the normalization operator used by Kanatani in his book [Kanatani, 1993], *i.e.*,

$$N\left[\boldsymbol{P}\right] = \frac{\boldsymbol{P}}{||\boldsymbol{P}||} \tag{3.7}$$

This operator performs the necessary transformation of an image point $\boldsymbol{P_i}$, or world point $\boldsymbol{P}$, onto the unit sphere.

Image lines can also be represented in a similar way. Any image line defines a plane with the centre of projection. A vector normal to this plane uniquely defines the image line. A normalized vector, normal to the plane defined by the image line and the optical centre, can be used to represent the line.

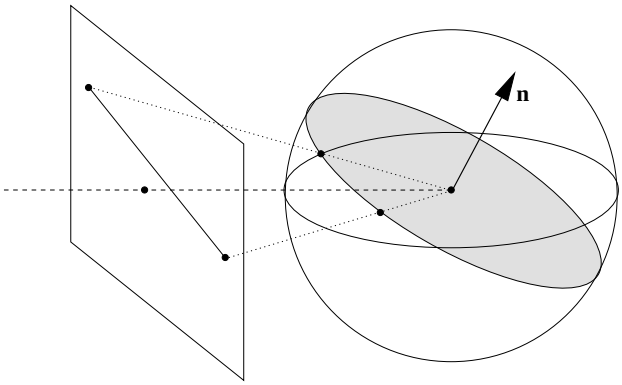


Figure 3.3: Line projection onto Unit Sphere.

For a given image line with equation $au+bv+c=0$, the unit sphere vector that represents the line is given by

$$\boldsymbol{n} = \frac{1}{\sqrt{a^2+b^2+(c/f)^2}} \begin{bmatrix} a \\ b \\ c/f \end{bmatrix} = \pm N \left(\begin{bmatrix} a \\ b \\ c/f \end{bmatrix} \right) \tag{3.8}$$

The normal to the plane defined by the line and the centre of projection, $\boldsymbol{n}$, is projected onto the image plane as $(a,b)^{\mathsf{T}}$.If we consider that $(a,b)^{\mathsf{T}}$ is a 2D vector normal to the line in the image plane, and c is the distance to the image origin. Since the image plane is placed at distance f away from the origin along the Z-axis, it follows by simple triangulation that $(a,b,c/f)^{\mathsf{T}}$ will be normal to the plane defined by the image line and the centre of projection.

The image line equation can be written as

$$au + bv + c = 0 = \begin{bmatrix} u \\ v \\ f \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} a \\ b \\ c/f \end{bmatrix} = \boldsymbol{P_i}.\boldsymbol{n} = N\left[\boldsymbol{P_i}\right].\boldsymbol{n} = 0 \tag{3.9}$$

The unit sphere model is based on Kanatani's book [Kanatani, 1993], where computational and statistical aspects of computer vision are studied in a projective framework. Stolfi presents in his book [Stolfi, 1991] an oriented projective geometry based on the unit sphere model.

In the following sections, unless otherwise stated, image lines and points will be represented by their corresponding unit sphere vectors.

Duality of Points and Lines

Consider the intersection of two image lines $\boldsymbol{n}$ and $\boldsymbol{n}'$. Their intersection point $\boldsymbol{m}$ must be orthogonal to both $\boldsymbol{n}$ and $\boldsymbol{n}'$ (*i.e.*, belong to both lines), and using the external vector product we have

$$\boldsymbol{m} = \pm N\left[\boldsymbol{n} \times \boldsymbol{n}'\right] \tag{3.10}$$

Consider a line passing through two image points $\boldsymbol{m}$ and $\boldsymbol{m}'$. The line passing through them, $\boldsymbol{n}$, which is orthogonal to both points $\boldsymbol{m}$ and $\boldsymbol{m}'$, is given by

$$\boldsymbol{n} = \pm N\left[\boldsymbol{m} \times \boldsymbol{m}'\right] \tag{3.11}$$

Considering these two equations we can see that the role of image points and image lines are interchangeable. This duality is one of the most fundamental characteristics of

projective geometry, here made evident by the unit sphere model.

Collinearity of Points and Concurrency of Lines

A given set of image plane points are collinear if their corresponding unit sphere vectors, $\boldsymbol{m_i}$, all lie within the same plane, touching the unit sphere along a great circle. All $\boldsymbol{m_i}$ are orthogonal to the unit vector $\boldsymbol{n}$ that defines the line passing through them all, *i.e.*,

$$\boldsymbol{n}.\boldsymbol{m_i} = 0 \tag{3.12}$$

From the duality of points and lines it follows that a given set of image lines are concurrent if their vectors, $\boldsymbol{n_i}$, all lie within the same plane, all $\boldsymbol{n_i}$ are orthogonal to the unit vector $\boldsymbol{m}$ that defines their common point, *i.e.*,

$$\boldsymbol{m}.\boldsymbol{n_i} = 0 \tag{3.13}$$

For both cases, the mentioned plane is defined by the unit sphere centre (centre of projection) and the normal $\boldsymbol{n}$.

Orthogonality and Conjugate Points

Image points $\boldsymbol{m}$ and $\boldsymbol{m}'$ are said to be conjugate to each other if

$$\boldsymbol{m}.\boldsymbol{m}' = 0 \tag{3.14}$$

and image lines $\boldsymbol{n}$ and $\boldsymbol{n}'$ are conjugate to each other if

$$\boldsymbol{n}.\boldsymbol{n}' = 0 \tag{3.15}$$

In image coordinates we have that image points $(u, v)^\mathsf{T}$ and $(u', v')^\mathsf{T}$ are conjugate to each other if

$$uu' + vv' + f^2 = 0 \tag{3.16}$$

and the projective lines passing through each point and the centre of projection are orthogonal.

Image lines $ax + by + c = 0$ and $a'x + a'y + c' = 0$ are conjugate to each other if

$$aa' + bb' + \frac{cc'}{f^2} = 0 \tag{3.17}$$

and the projective planes passing through each line and the centre of projection are orthogonal.

3.1.3 Vanishing Points and Vanishing Lines

Since the perspective projection maps a 3D world onto a plane or planar surface, phenomena that only occurs *at infinity* will project to very finite locations in the image. Parallel lines only meet at infinity, but as seen in figure 3.4, the point where they meet can be quite visible and is called the *vanishing point* of that set of parallel lines.

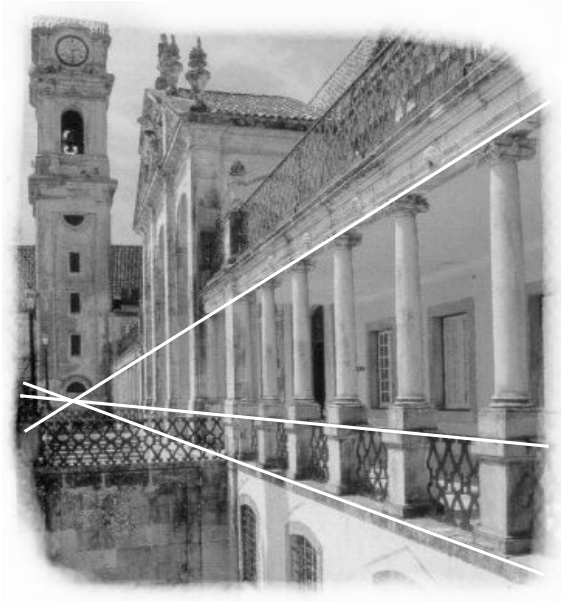


Figure 3.4: Picture of *Via Latina* at Coimbra University showing a vanishing point.

A space line with the orientation of an unit vector $\boldsymbol{m}$ has, when projected, a *vanishing point* with unit sphere vector $\pm\boldsymbol{m}$. 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.

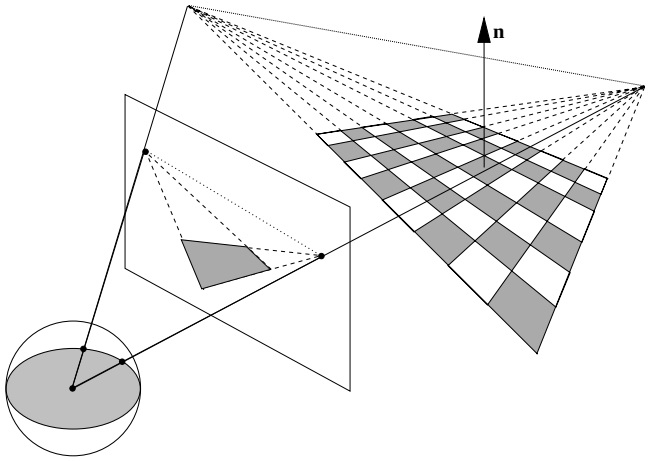


Figure 3.5: Vanishing line of a planar surface.

A planar surface with a unit normal vector $\boldsymbol{n}$, not parallel to the image plane has, when projected, a *vanishing line* with unit sphere vector $\pm\boldsymbol{n}$. Since the vanishing line is determined alone by the orientation of the planar surface, then the projections of planar surfaces parallel in the scene define a common vanishing line. A vanishing line is a set of all vanishing points corresponding to the lines that belong to the set of parallel planes defining the vanishing line.

In an image the horizon can be found by having two distinct vanishing points as seen in figure 3.5. With a suitable calibration target (e.g. a levelled square with well defined edges) the horizon can be determined.

The effective focal distance f of a camera can be estimated using vanishing points. If the vanishing points, $(u, v)^\mathsf{T}$ and $(u', v')^\mathsf{T}$, correspond to orthogonal sets of parallel lines, they are conjugate to each other and from (3.16) we have

$$f = \sqrt{-uu' - vv'} \tag{3.18}$$

If the points are given in unit sphere projection vectors $\boldsymbol{m}$ and $\boldsymbol{m}'$, obtained using an initial estimate f', the correct estimate of f is given by

$$f = f'\sqrt{-\frac{m_1 m_1' + m_2 m_2'}{m_3 m_3'}} \tag{3.19}$$

3.2 Image Processing

In the previous section I was concerned with the geometry aspects of vision, and the mapping between world and image points. I will now discuss image processing, *i.e.*, methods to extract information from the images. Features most commonly used in computer vision tasks are edges, line segments, regions and interest points. In the next sections I will present the image processing methods used for my work. Gradient-based edge detection was used as an initial step for image line segmentation. Several edge detection methods are compared to evaluate the quality of the edge direction estimate, a crucial factor for the subsequent vanishing point estimation. The detected edge points are than grouped into lines using model-based segmentation. An interest point detector is also presented, since it will be used in this work to obtain feature points across the stereo image pair. For my work I was interested in having fast image processing algorithms suitable for real-time implementation.

3.2.1 Gradient-Based Edge Detection

One important feature to be extracted from an image are the edges. The task of *edge detectors* requires neighbourhood operators that are sensitive to changes in image gray values. Edge detection is mostly based on differentiation in some form or other. In discrete images, differentiation is replaced by discrete differences which only approximate to differentiation. This approximation leads to anisotropic edge detection, where edges are not detected equally well in all directions, and erroneous estimation of the direction of the edges. The principal problem with all types of edge detectors is that a derivative operator can only be approximated on a discrete grid, which leads to a wide variety of solutions for edge detectors, that balance a trade off between complexity and errors.

From the spatial derivatives in the image coordinate directions, the gradient vector operator $\boldsymbol{\nabla}$ can be expressed as

$$\boldsymbol{\nabla} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \tag{3.20}$$

where $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ are the X and Y spatial derivative operators. If $I(x, y)$ is the image intensity function, the gradient is given by $\boldsymbol{\nabla} I$ where

$$\boldsymbol{\nabla} I = \begin{bmatrix} \frac{\partial}{\partial x} I \\ \frac{\partial}{\partial y} I \end{bmatrix} \tag{3.21}$$

First-derivative methods rely on gradient magnitude and orientation to detect image edges. The magnitude of the gradient vector $\boldsymbol{\nabla} I$ provides a measure of the edge strength, and can be expressed as

$$||\boldsymbol{\nabla} I|| = \sqrt{\left(\frac{\partial}{\partial x} I\right)^2 + \left(\frac{\partial}{\partial y} I\right)^2} \tag{3.22}$$

and the image edges can be segmented by thresholding the magnitude of the gradient. To overcome the computational cost involved in (3.22), the magnitude is sometimes approximated by

$$||\boldsymbol{\nabla} I|| \approx \left|\frac{\partial}{\partial x} I\right| + \left|\frac{\partial}{\partial y} I\right| \tag{3.23}$$

but a detector based on this gradient estimation is $\sqrt{2}$ times more sensitive to edges along the diagonals, and therefore anisotropic in the edge strength measure. Depending on the application, and thresholding method, this can be an acceptable error. More elaborate edge segmentation methods can also be used, such as thresholding the gradient magnitude with hysteresis based on already detected neighbouring edge pixels.

The direction of the gradient vector $\boldsymbol{\nabla} I$ provides a measure of the direction of greatest change, which presumably is the direction across the edge (i.e., edge normal). It can be expressed as

$$\phi(\boldsymbol{\nabla} I) = \arctan\left(\frac{\frac{\partial}{\partial y} I}{\frac{\partial}{\partial x} I}\right) \tag{3.24}$$

where $\arctan()$ is the arctangent which results in angles in $\left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$.

Second-derivative methods directly extract the peaks in gradient magnitude by detecting zero-crossings in the second derivative of the gradient vector. These methods, however, are extremely sensivive to noise and produce thick edges.

First-derivative methods rely on convolution to calculate the gradient, and vary in the convolution kernel used to estimate each of the spatial derivatives.

With K_x and K_y being the convolution kernels for the estimation of $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$, the gradient is estimated by

$$\boldsymbol{\nabla} I = \begin{bmatrix} \frac{\partial}{\partial x} I \\ \frac{\partial}{\partial y} I \end{bmatrix} \approx \begin{bmatrix} I * K_x \\ I * K_y \end{bmatrix} \tag{3.25}$$

and

$$||\boldsymbol{\nabla} I|| \approx \sqrt{(K_x * I)^2 + (K_y * I)^2} \tag{3.26}$$

$$\phi(\boldsymbol{\nabla} I) \approx \arctan\left(\frac{K_y * I}{K_x * I}\right) \tag{3.27}$$

Gradient based edge detectors vary in the kernels used, as seen in the following sections.

First-Order Central Differences

One dimensional central difference kernels such as

$$K_x = \tfrac{1}{2}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \qquad K_y = \tfrac{1}{2}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \tag{3.28}$$

can be used to estimate the gradient, but will predominantly detect edges that lie along the coordinate axes. A three pixel wide mask is used, since having a two pixel wide mask, a central pixel would have to be chosen, and would introduce a shift of half a pixel.

Roberts Edge Detectors

The Roberts edge detectors use the smallest possible difference filters that have a common central point, but estimates the gradient along diagonal directions. The kernels used are

$$K_{d1} = \tfrac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad K_{d2} = \tfrac{1}{2}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{3.29}$$

The gradient is estimated as above, but using a different coordinate system, with axis along the diagonals of the image. The estimated gradient vector is therefore rotated by 45°.

Prewitt Edge Detectors

The Prewitt kernels are based on central differences, but to reduce noise sensitivity, average in the direction perpendicular to the differentiation. The Prewitt edge detector uses the following kernels:

$$K_x = \frac{1}{6}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad K_y = \frac{1}{6}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \tag{3.30}$$

Sobel Edge Detectors

The Sobel edge detectors are also based on central differences and average the image in the direction perpendicular to the differentiation. The central pixels are given a greater weight leading to the following kernels:

$$K_x = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad K_y = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3.31}$$

The Sobel kernels are equivalent to applying a 3×3 approximation to a Gaussian, equivalent to blurring the image, followed by central difference derivative operators.

A modified Sobel kernel has been proposed (optimised Sobel in [Jahne, 1997]) that estimates the gradient with the following kernels

$$K_x = \frac{1}{32}\begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \qquad K_y = \frac{1}{32}\begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \tag{3.32}$$

The modified Sobel filter has a lower angle error than the standard Sobel filter, as shown in the following section. In figure 3.6 we see the result of the modified Sobel edge detector

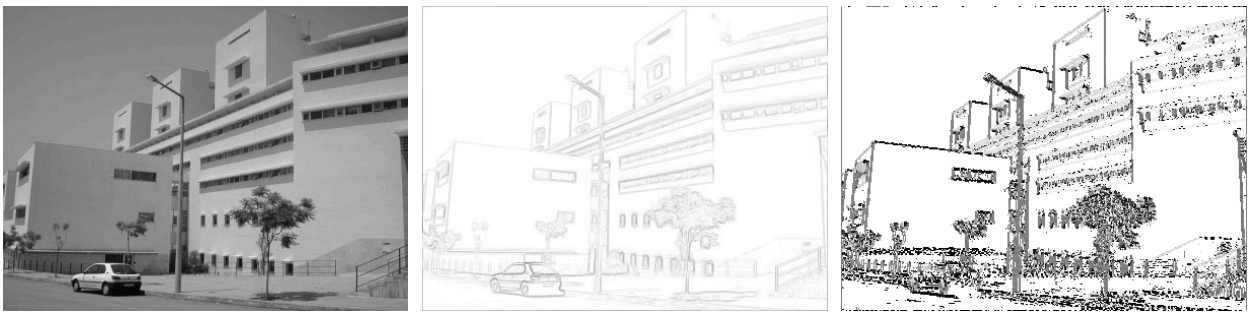


Figure 3.6: Original image, gradient magnitude and gradient phase at edges, obtained with modified Sobel edge detector

on an image.

Edge Direction Estimation Error

For this work I am mainly interested in having edge detectors that provide a good estimate of edge direction given by the phase of the gradient. Since detected image lines will be used to find vanishing points, their direction should be correctly determined. When using the model-based segmentation presented in the nest section, using the estimated edge direction can speed up the segmentation process.

To evaluate the quality of the gradient phase provided by the different edge detectors, a ring pattern is used. The ring pattern used has a set of smooth concentric rings of decreasing edge thickness. The ring pattern image is generated by

$$I(r) = \begin{cases} \frac{1}{2} + \frac{1}{2}\sin\left(2\pi r \frac{n}{16} e^{\left(\frac{2r}{n}\right)^2}\right) & , \; r < \frac{n}{2} \\ \frac{1}{2} & , \; r \geq \frac{n}{2} \end{cases} \tag{3.33}$$

where $I(r)$ is the $n \times n$ square image intensity at distance r from the image centre, admitting gray pixel values in the range $[0;\ 1]$ where 0 is black and 1 is white.

In figure 3.7 the image given by (3.33) is shown, and the expected gradient direction with angles from $[-90°;\ 90°]$ mapped to 256 gray scale levels.

To compare the edge direction estimated by edge detectors, their estimated gradient phase image was subtracted from the reference image presented above. This error image

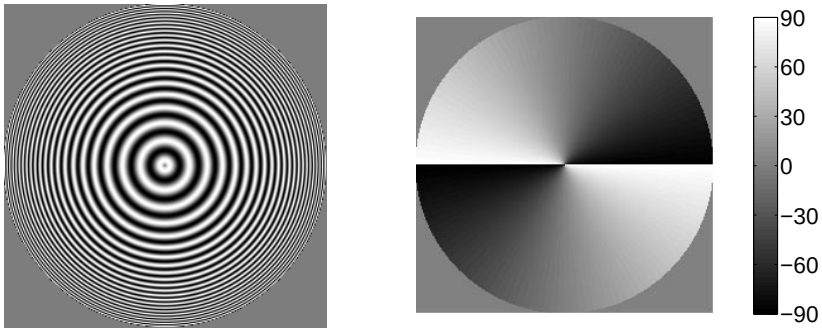


Figure 3.7: Ring pattern image and gradient phase image

was then sampled along two concentric circles with radius $\frac{3n}{8}$ and $\frac{7n}{16}$, were the ring pattern edges approach one pixel thickness, as show in figure 3.8.

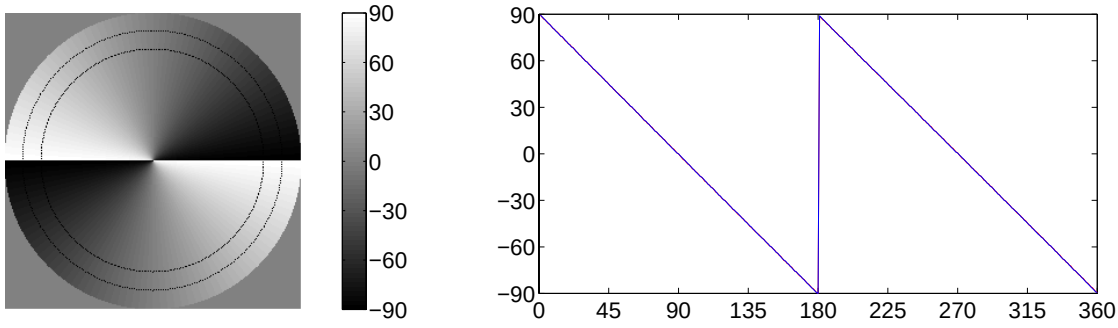


Figure 3.8: Gradient phase image and gradient phase sampled along two concentric circles.

In figure 3.9 these error images are shown, as well as the error along the two circles shown in figure 3.8. The higher error plot corresponds to the outer circle, where the edges are thinner. The Roberts edge detector was rotated 45° to get the correct gradient direction.

It is clear that the modified Sobel detector has the lowest angle error amongst the tested edge detectors. The first-order central differences and the Sobel detectors have angle errors in opposite directions. The modified Sobel is the result of the combination of both to reduce the gradient direction angle error [Jahne, 1997].

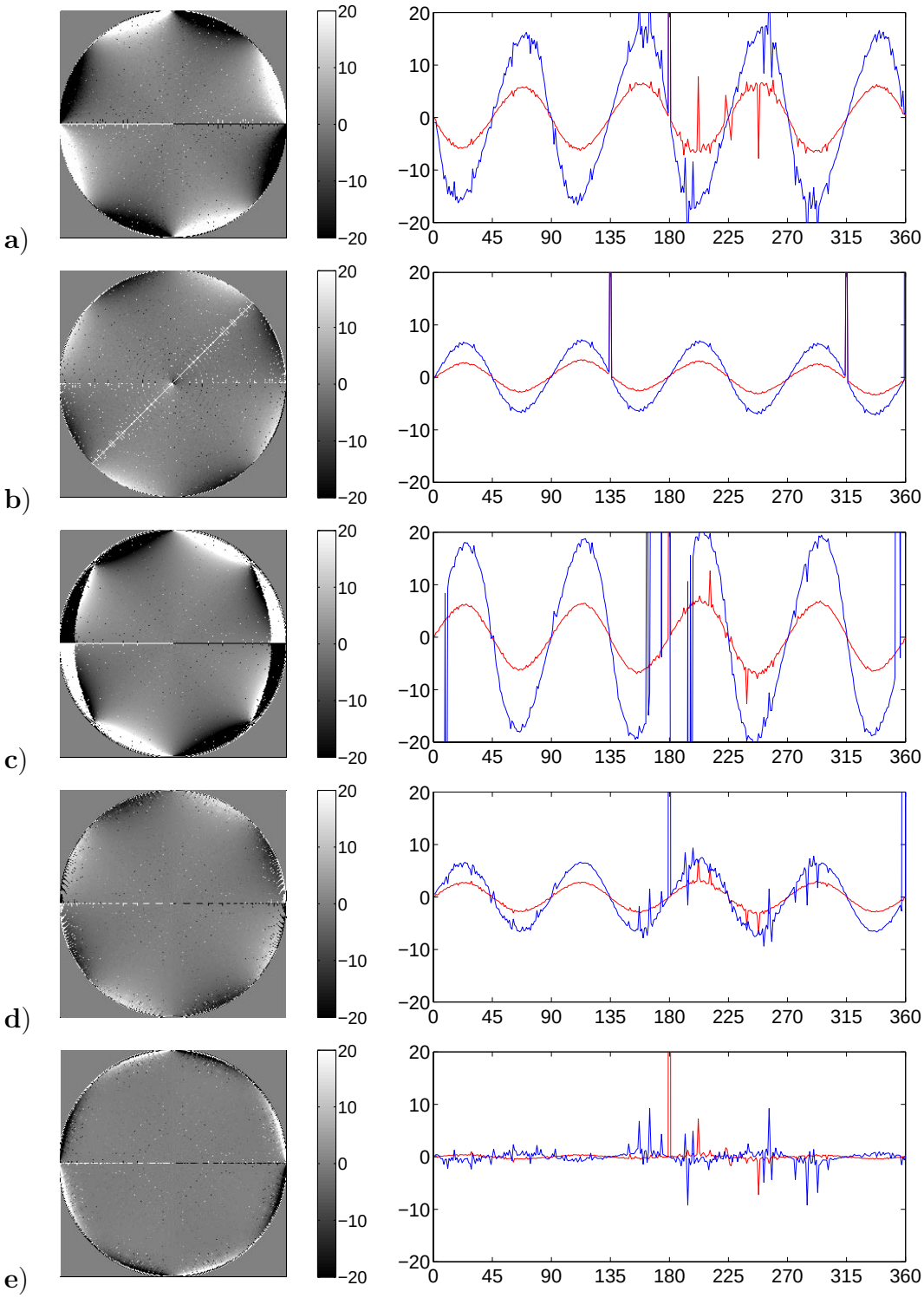


Figure 3.9: Comparison of edge direction estimation errors: **a**) first-order central differences; **b**) Roberts; **c**) Prewitt; **d**) Sobel; **e**) modified Sobel.

3.2.2 Model-Based Segmentation

Edge detectors perform low-level image processing, and rely only on local information. In order to extract higher level information a model based approach can be used. If we know the geometrical model of the features we are looking for in the image, we can compare it with the local information to segment the wanted features. This can be achieved by transforming the image into a parameter space consistent with the geometrical features. The feature detection problem is mapped to a simple peak detection problem in parameter space.

The Hough Transform

The Hough transform is used to detect image lines. In order to group the edge points into lines, image points are mapped to parameter space lines, transforming a line detection into a line intersection problem.

The parameter space can be built using line parameters (d, θ), where d is the distance of the line from the origin of the coordinate system and θ the angle with the x axis, as shown in figure 3.10. Using the standard line slope and offset parametrisation would be impractical in a discrete model, with problems when the slope approached infinity and uneven representation of line direction. With the (d, θ) parametrisation image points are not transformed into straight lines in parameter space, but into sinusoidal lines.

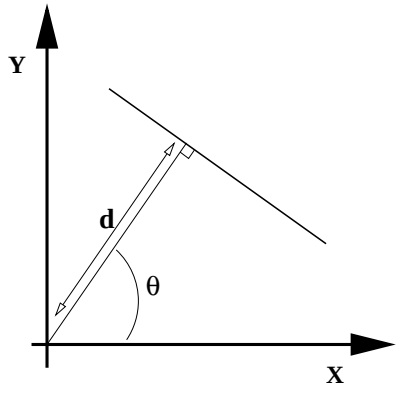


Figure 3.10: Line parametrisation used in Hough transform.

The Hough transform maps image points $\boldsymbol{p}_i = (u_i, v_i)^\mathsf{T}$to parameter space $= (d, \theta)^\mathsf{T}$

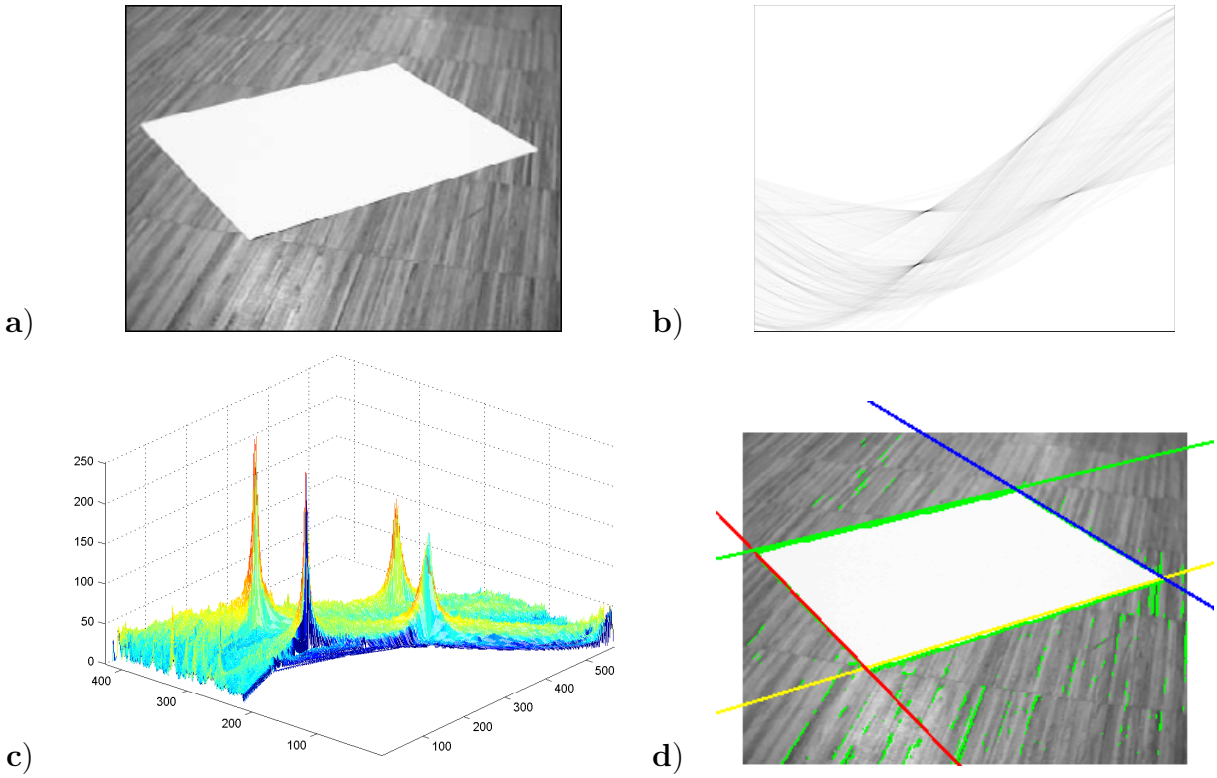


Figure 3.11: **a**) Original image; **b**) Hough transform parameter space; **c**) 3D view of Hough transform parameter space; **d**) Original image with segmented lines.

where

$$u_i \cos\theta + v_i \sin\theta = d \tag{3.34}$$

So for each image point, the complete range of θ is parsed and the corresponding sinusoidal line plotted in the parameter space. In order to detect line intersections in parameter space, a voting scheme is used. The parameter space accumulates the voting of all the image edge points. The highest peaks found in parameter space correspond to the image dominant lines.

The parameter space is than parsed to find the highest peak. Having identified the line, a neighbourhood of this peak is zeroed so that the second line can be found, and the process is iterated to find more lines. In figure 3.11 we can see the result of applying the Hough transform to an image with four distinct dominant lines.

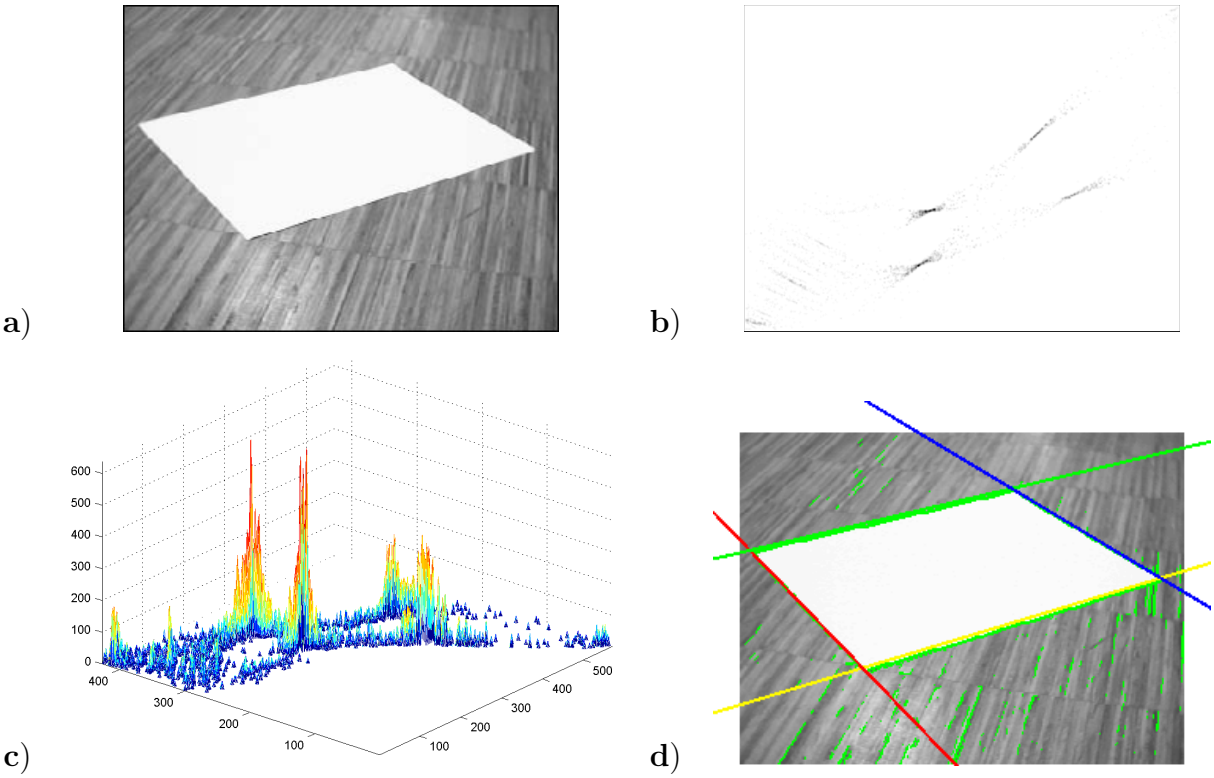


Figure 3.12: **a**) Original image; **b**) fast Hough transform parameter space; **c**) 3D view of fast Hough transform parameter space; **d**) Original image with segmented lines.

Orientation-Based Fast Hough Transform

The Hough transform is used to group the edge points into lines. Since the Sobel filter provides a local edge orientation estimate, we can use an orientation-based fast Hough transform that avoids the high computational effort of the parameter space tranform [Jahne, 1997].

The fast Hough transform uses the edge orientation given by the Sobel filter. Since the local edge orientation is noisy, a small neighbourhood mask is used when voting in the transform space. This vote spread can be done in several ways, but reasonable results were obtained using a simple square Gaussian mask in parameter space. The magnitude of the gradient is also used so that stronger edges have a higher weighting factor in the accumulator parameter space. In figure 3.12 we can see the result of applying the fast Hough transform to the same image as above, where the image lines are equally detected.

The fast Hough transform is therefore capable of replacing the full Hough transform, although the full Hough transform is more robust. Improving the voting spread in parameter space can improve the robustness of the fast Hough transform, but makes it slower. The full Hough transform corresponds to spreading the vote to the full sinusoidal line corresponding to all possible image lines going though that point.

My emphasis is on fast methods, and Hough transform based methods such as presented by Palmer [Palmer et al., 1993], and the cascaded Hough transform used by Lutton [Lutton et al., 1994] and Tuytelaars [Tuytelaars et al., 1998], were not used in this work.

3.2.3 Interest Point Detectors

Interest points can be defined as locations in the image where the signal changes two-dimensionally. Examples include edge corners and edge junctions or intersections, as well as locations where the texture varies significantly. Edge detectors can be based on contours, where previously detected edges are parsed to find maximal curvature or inflections points as well as junctions and intersections. Signal based methods compute the measure which indicates the presence of an interest point directly from the signal. Methods based on template fitting are used to obtain sub-pixel accuracy. They fit the image signal to a parametric model of a specific type of interest point, such as corners or vertices, and are not applicable in a general context [Schmid et al., 1998].

Many interest points detectors have been already proposed in the literature, [Deriche and Goraudon, 1993] and [Smith and Brady, 1997] include extensive surveys. The lack of a precise concept of interest point leads to this large set of proposed detectors. Based on the application, and typical image content, some are more suitable than others.

For my work I needed to detect feature points across a stereo image pair. Instead of using random or regularly sampled points, feature points need to be detected do that repeatability can be achieved across images. These are very similar images with a different viewing angle. The SUSAN 2D feature detector was chosen since it provided reasonable results with small computation time. In [Schmid et al., 1998] the Precise Harris is shown to be better than others in dealing with perturbations such as image rotations, scale changes, variations of illumination. In [Bres and Jolion, 1999] the improved Harris and

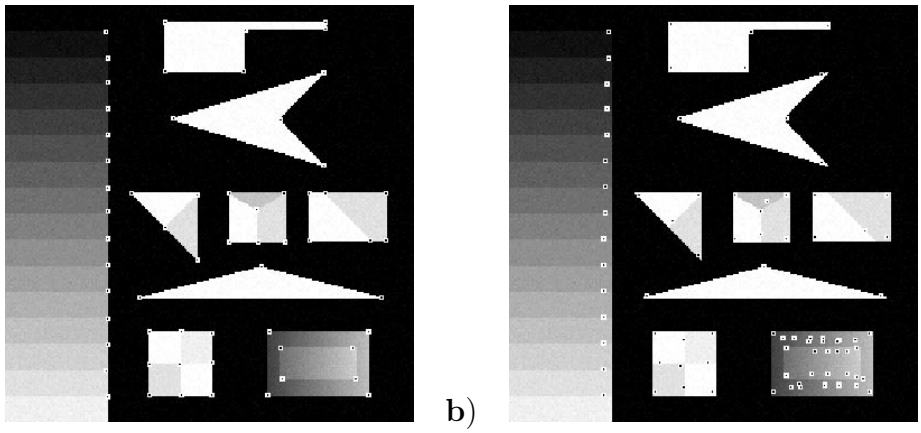


Figure 3.13: **a**) Output of Susan feature point detector given the test image with Gaussian noise added.; **b**) Output of Harris feature point detector with the same test image. (taken from [Smith and Brady, 1997])

SUSAN detectors are compared with a new method for detection of interest points for image indexation. The strong points in favour of the SUSAN detector are good localization accuracy of the detected features, noise immunity, and good coverage of features other than corners.

The Susan Feature Point Detector

The SUSAN principle is based on the concept that each image point has an associated local area of similar brightness, the USAN (univalue segment assimilating nucleus). From the sizes, centroid and second moments of the USAN two dimensional features and edges can be detected [Smith and Brady, 1997]. No image derivatives are used. This feature point detector, besides detecting corners can also detect many other types of localized structures that arise, such as T, Y and X junctions, as well as less rigidly defined structures.

Figure 3.13 compares the Susan detector with the Harris detector. In [Smith and Brady, 1997] it is named Plessey, since the Harris detector is a slightly modified version of the Plessey corner detector [Deriche and Goraudon, 1993], but in [Schmid et al., 1998] and [Bres and Jolion, 1999] the same method is named Harris. The Harris algorithm is known for its detection reliability, but the SUSAN detector finds the corners with much greater accuracy.

3.3 Summary

The framework used for computer vision modelling was presented. The unit sphere model provides a good representation for central projection, providing a suitable representations for vanishing points and lines. The geometry leading to vanishing points and lines was analysed, and is the basis for the results presented in the following chapters.

Some image processing algorithms needed for this work were presented. Comparison of several edge detectors revealed that a modified version of the Sobel edge detector had the lower edge direction estimation error. This was important since the model-based segmentation can use edge direction to speed up the computation, as shown in the fast Hough transform used in this work.

Chapter 4

System Modelling

4.1 Experimental Setup

As with any work in applied robotics, I needed the hardware to carry out my experiments. The experimental setup here described is the one being used now. It represents a compromise between performance, flexibility and cost. Some painstaking work was involved in solutions that were later abandoned, either for lack of time to fully develop them, or just because they proved inadequate. The work on ground plane segmentation was initially done using an existing robotic head to which I attached my inertial system prototype, as described in [Lobo and Dias, 1997].

4.1.1 Inertial System Prototype

Although complete inertial sensor packages based on low-cost sensors are now commercially available (e.g. [Crossbow, 2001] [Summit Instruments, 2001]), I chose to build my own inertial system prototype. This allowed me to tailor the system to my specific needs, gain a better insight into the technology, and work towards a low-cost inertial and vision system for robotic applications, within the available budget. The sensors used in the prototype system include a three-axial accelerometer, three gyroscopes and a dual-axis inclinometer. A temperature sensor is also included to enable future implementation of temperature error compensation.

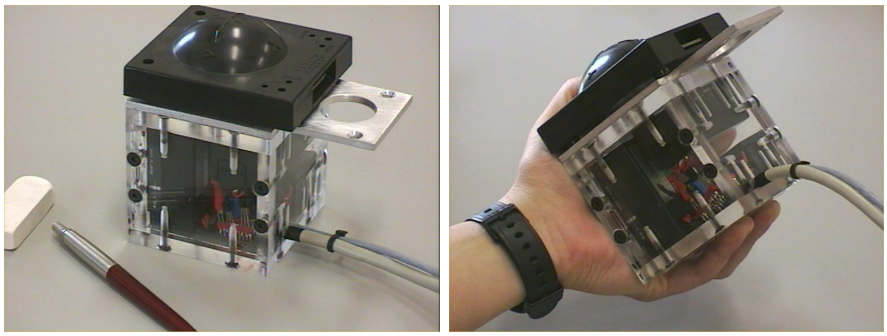


Figure 4.1: The inertial system prototype, built for the present work.

The three-axial accelerometer chosen for the system, while minimising eventual alignment problems, did not add much to the equivalent cost of three separate single-axis sensors. The device used was Summit Instruments' 34103A three-axial capacitive accelerometer [Summit Instruments, 2001]. In order to keep track of rotation on the x-, y- and z-axis, three gyroscopes were used. The piezoelectric vibrating prism gyroscope Gyrostar ENV-011D built by Murata was chosen [Murata Electronics, 2001]. Initially tilt about the x and y-axis was measured with a dual axis AccuStar electronic inclinometer, built by Lucas Sensing Systems [Lucas, 2001], details given in [Lobo and Dias, 1997]. The inertial sensors were mounted inside an acrylic cube, enabling the correct alignment of the gyros, inclinometer (mounted on the outside) and accelerometer, as can be seen in figure 4.1. This inertial system can measure angular velocity with 0.1 $deg.s^{-1}$ resolution, and linear acceleration with 0.005 g resolution.

For now, I am only extracting the camera's attitude from the accelerometer data. This is done by keeping track of the gravity vector $\boldsymbol{g}$. The current implementation only estimates the camera's attitude when it is motionless. The gyros are intended to track all rotations so that the attitude can be known when acceleration other than gravity is present, as explained in chapter two.

4.1.2 Vision System

Although some initial work was done with an existing robotic head developed at my lab [Lobo and Dias, 1997], as seen in figure 4.2, the current system was built using two

commercial pan and tilt units. The robotic head was controlled by a PC running MS-DOS with a TMS320C40 DSP [Texas Instruments, 2001] based board for image aquisition and processing. Programming was hard, the operating system proved limited, and the vergence resolution was poor. The only degree of freedom in the active vision system that I needed for my tests was vergence. Therefore I could do away with the robotic head that, besides having vergence on a common baseline, had neck pan and tilt.

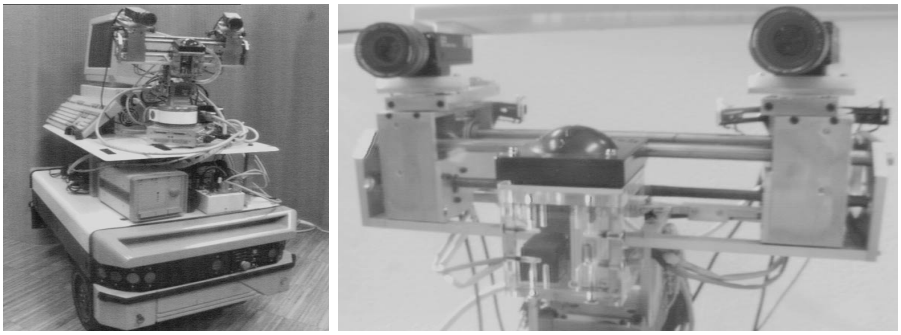


Figure 4.2: The old vision system and mobile robot where the inertial system prototype was initially used.

The new system I built uses two commercial pan and tilt stepper motor units, built by Directed Perception [Directed Perception, 2001], onto which two Sony XC-999 NTSC cameras, with 12 mm lens, were mounted. The pan and tilt units are used only for panning, enabling the system to verge with a common base line. So, apart from an initial reset and alignment, only the pan will be used. The cameras are positioned with their optical centre over the panning axis to ensure a constant common baseline.

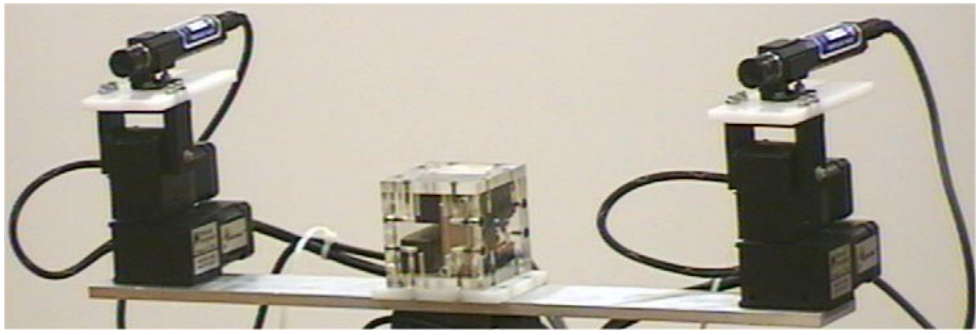


Figure 4.3: Vision system with inertial sensors.

The inertial system prototype was mounted onto the vision system as seen in figure 4.3. A large baseline of 0.5 m had to be used due to the initialisation on the pan and tilt units, that forces a full swing motion of the motors. By having a large baseline, the robustness of the stereo algorithms is also improved. The vergence resolution is 0.05413 degrees per step.

4.1.3 Processing System

The whole systems runs on a PC running a public domain unix operating system, the FreeBSD [FreeBSD, 2001]. An Advantech PCL818HD card [Advantech, 2001] was included to handle the inertial sensor data acquisition. The Matrox Meteor framegrabber [Matrox, 2001] was used to synchronously capture the images from both cameras. The FreeBSD driver for the Meteor was reprogrammed to handle the Advantech board and enable sensor data acquisition synchronous with the captured frames. The pan and tilt units connect to the PC's two RS232 ports. The system I built, while not enabling top performances, is very flexible and reconfigurable.

Figure 4.4: Complete system with host PC, hardware module, and stereo rig with inertial sensors mounted on a tripod

To study the integration of inertial information and vision in artificial autonomous mobile systems, the system was mounted onto a mobile robot platform, a Scout II from Nomadic Technologies [Nomadic Technologies, 2001]. Figure 4.5 shows the complete system and architecture.

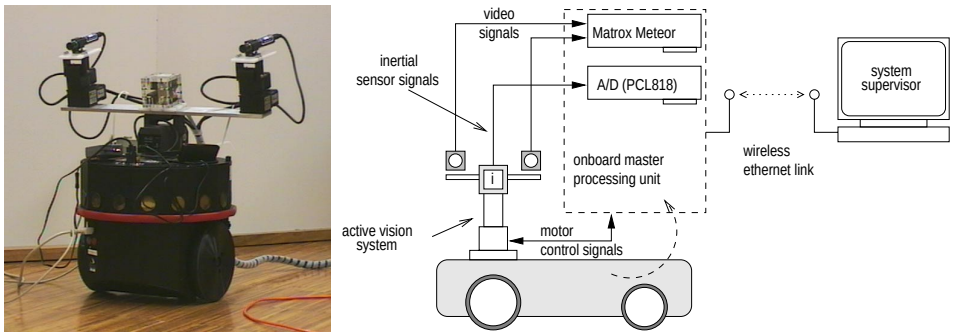


Figure 4.5: System on mobile robot, and system architecture.

Image and Sensor Data Synchronization

Since none was available, I had to write a special device for the PCL818HD data acquisition board. Later this device driver was merged with the matrox meteor frame-grabber driver. This enabled a proper synchronization between the acquired images and sensor data. With this setup, I was able to have a sequence of images, each with a time-stamp and inertial sensor data set.

In order to have data from the inertial sensor data synchronous with the grabbed images, some care had to be taken in implementing the software drivers. The NTSC standard has a nominal frame rate of 30 Hz, but the exact figure is actually 59.94 Hz [Poynton, 1996]. On the other hand, the PCL818HD data acquisition board had a top acquisition rate of 100 kHz, but had to multiplex the 8 channels in time. The PCL818HD has an internal 1024 word FIFO. I aimed at having the FIFO emptied at each frame grab, and append it directly to the image buffer. To accomplish this I merged my PCL818HD device driver with the meteor driver developed by Jim Lowe and Mark Tinguely for the FreeBSD system.

The system was set to run at 10 frames per second, or to be more precise at 9.99 Hz. This rate allowed for some real-time processing and also for data logging. The PCL818HD channel multiplexing limited the data acquisition rates to 12.5 kHz per channel. In order to have no more than one full FIFO for each frame, the chosen rate was 1.25 kHz. Due to internal timers limitations, the 8 channels were sampled at 20 kHz bursts, and not at the top rate.

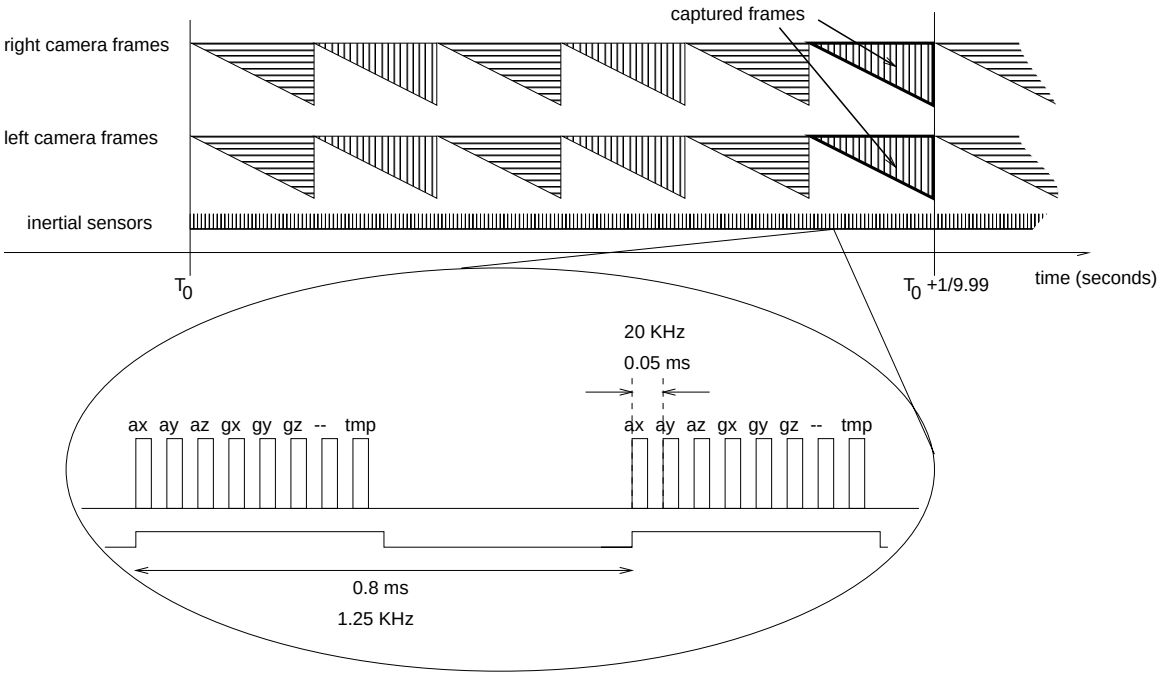


Figure 4.6: Timing of image and sensor data acquisition

Figure 4.6 shows the timing used by my software driver. Each stereo image pair would have a set of 125 samples from each sensor if the frame rate was 10 Hz. But since the true frame rate is 9.99 Hz, only 124 samples are taken from each sensor, and the PCL818HD timers are reset at each frame to deal with the time gap left behind.

Inertial Calculations

The inertial sensors connected were 3 accelerometers and 3 gyros. A temperature sensor was also placed there to enable future development of temperature compensated sensor error models. The accelerometer sensor by Summit Instruments comes with a factory calibration table, providing sensitivity, offset and alignment data. Since the offsets tend to drift with environmental conditions, they are reset at the beginning of each session. An initial rest position is assumed and the data from the first frame is averaged to provide the offsets. The same was done for the gyro sensors. For my tests I did not require the complete update of position and attitude, and with the sensors used, the position data

would only be within an acceptable error for a short time. The attitude is recovered using only the accelerometer data. This imposes the restriction of having only static poses, so that the only measured acceleration is gravity. So, for each frame the accelerometer data is averaged and the result provides the vertical reference for the images.

4.2 System Model and Inertial Sensor Data

Chapter 3 introduced some generic modelling for computer vision geometry. Having described the experimental setup in the previous section, I must now apply these concepts to my specific model. I have two cameras, instead of just one, but their relative pose is known from the pan and tilt units. The inertial sensors, by measuring the gravity vector, provide vertical reference, an image horizon, and ground plane orientation, from which a robot navigation frame of reference, and the collineation of ground plane points can be obtained.

4.2.1 System Geometry

The system used has an inertial unit at the middle of the stereo camera baseline, as seen in figure 4.3. The cameras' pan is controlled so as to have a symmetric verge angle. The system's coordinate frame referential, $\{\mathcal{C}\}$, is defined as having the origin at the centre of the baseline of the stereo cameras, as seen in figure 4.7. This referential was named *Cyclop*, after the mythological one-eyed giant.

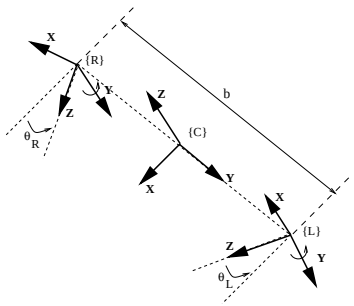


Figure 4.7: System Geometry

Each camera has its own referential, $\{\mathcal{R}\}$ and $\{\mathcal{L}\}$. Besides being translated from $\{\mathcal{C}\}$ along the baseline (represented by the homogeneous coordinate matrixes $\boldsymbol{T_r}$ and $\boldsymbol{T_l}$), and rotated θ_R and θ_L along the $\{\mathcal{C}\}$ Z-axis ($\boldsymbol{R_r}$ and $\boldsymbol{R_l}$), their axis are also swapped to comply with the typical referential convention used for camera images ($\boldsymbol{S_r}$ and $\boldsymbol{S_l}$) - see figure 4.7. Notice that for this work, I only used symmetric vergence, *i.e.*, $\theta = \theta_R = -\theta_L$. Using these conventions we can express a world point $\boldsymbol{P}$ in the Cyclop referential $\{\mathcal{C}\}$, given its coordinates in the camera referential $\{\mathcal{R}\}$ or $\{\mathcal{L}\}$ by

$$ {}^{\mathcal{C}}\boldsymbol{P} = \boldsymbol{T_r}.\boldsymbol{R_r}.\boldsymbol{S_r}.{}^{\mathcal{R}}\boldsymbol{P} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{R}}.{}^{\mathcal{R}}\boldsymbol{P} \qquad {}^{\mathcal{C}}\boldsymbol{P} = \boldsymbol{T_l}.\boldsymbol{R_l}.\boldsymbol{S_l}.{}^{\mathcal{L}}\boldsymbol{P} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{L}}.{}^{\mathcal{L}}\boldsymbol{P} \tag{4.1} $$

where

$$ \boldsymbol{S_r} = \boldsymbol{S_L} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{T_r} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{b}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{T_l} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{b}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2} $$

and

$$ \boldsymbol{R_r} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{R_l} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3} $$

and b is the baseline distance.

Modelling with VRML

In order to study the geometry to be implemented, I built a VRML world (Virtual Reality Modelling Language [Ames et al., 1997]). VRML is intended to be a standard for describing 3D scenes on the Internet. The cameras and pan and tilt units were modelled to scale, as seen in figures 4.8.

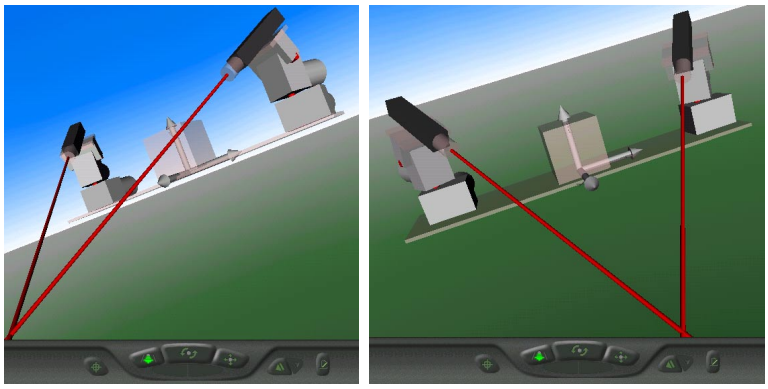


Figure 4.8: VRML model of vision system with inertial sensors.

A web browser with the appropriate plug-in lets the user see the images from the cameras viewpoint, or change the viewpoint and *fly* around the 3D world viewed by the vision system. The pan and tilt units can be moved by dragging the mouse. This model was later used to map detected world features.

4.2.2 Vertical Reference from Inertial Sensor Data

Both the inclinometer and the accelerometer data can be used to determine the system's attitude. When the system is motionless or subject to constant speed both inclinometer and accelerometers give the direction of the gravity vector $\boldsymbol{g}$ relative to the Cyclop referential $\{\mathcal{C}\}$. We can only determine the vertical unit vector normal to a ground levelled plane and not system pose, since rotations within the horizontal plane are not sensed. With α_x and α_y being the inclinometer sensed angles along the x and y-axis, the vertical unit vector will be

$$\hat{n} = -\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|} = \frac{1}{\sqrt{1 - \sin^2 \alpha_x - \sin^2 \alpha_y}} \begin{bmatrix} -\cos \alpha_x \sin \alpha_y \\ \cos \alpha_y \sin \alpha_x \\ \cos \alpha_y \cos \alpha_x \end{bmatrix} \tag{4.4}$$

given in the Cyclop frame of reference. The difference between inclinometer and accelerometer is mainly in their dynamic response. Initial work was done using the inclinometer data alone. Later, when the complete data acquisition system was operational, the accelerometer data was used instead.

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

$$\boldsymbol{a} = \boldsymbol{g} + \boldsymbol{a}_b \tag{4.5}$$

Assuming the system is motionless, then $\boldsymbol{a}_b = 0$ and the measured acceleration $\boldsymbol{a} = \boldsymbol{g}$ gives the gravity vector in the system's referential. So, with a_x,a_y and a_z being the accelerometer filtered measurements along each axis, the vertical unit vector will be given by

$$\hat{\boldsymbol{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = -\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|} = \frac{1}{\sqrt{a_x^2+a_y^2+a_z^2}} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad \text{and} \quad {}^{\mathcal{C}}\hat{\boldsymbol{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ 1 \end{bmatrix} \tag{4.6}$$

where ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$ is the homogeneous normal vector given in the Cyclop frame of reference.

Notice that if the assumption of the system being motionless or subject to constant speed is correct, than in the above equation

$$\sqrt{a_x^2 + a_y^2 + a_z^2} \cong 9.8 \; m.s^{-2} \tag{4.7}$$

and this condition can be tested and monitored by the system. As explained in the chapter 2, by performing the rotation update using the INS gyro data, gravity can be separated from the sensed acceleration. In this case ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$ is given by the rotation update, but must be monitored using the low pass filtered accelerometer signals, for which (4.6) still holds, to reset the accumulated drift.

In equation (4.6) the vertical unit vector is given in the Cyclop referential. The vertical for each of the cameras is given by

$${}^{\mathcal{R}}\hat{\boldsymbol{n}} = \boldsymbol{S}_{\boldsymbol{r}}^{-1}.\boldsymbol{R}_{\boldsymbol{r}}^{-1}.{}^{\mathcal{C}}\hat{\boldsymbol{n}} \quad \text{and} \quad {}^{\mathcal{L}}\hat{\boldsymbol{n}} = \boldsymbol{S}_{\boldsymbol{l}}^{-1}.\boldsymbol{R}_{\boldsymbol{l}}^{-1}.{}^{\mathcal{C}}\hat{\boldsymbol{n}} \tag{4.8}$$

This vertical reference will be used together with image cues to calibrate the vision sensor and detect world features such as vertical lines and the ground plane, as presented in the next chapters.

4.3 Summary

The experimental setup built and used for this work was described. The geometric model of the system was presented, incorporating the vertical reference provided by the inertial sensors. In some fixed setups this vertical reference might be known *apriori*, but the aim of this work is to explore the use of the reference provided by the inertial sensors. The need for inertial sensors becomes more obvious for walking or flying robots, but even with wheeled mobile robots it is advantageous. In the tests carried out with the mobile robot it became clear the importance of having inertial sensors close to the cameras. Whenever the robot moved the system's tilt varied slightly or had a low frequency oscillation due to some compliance of the mechanical mount and robot suspension and wheels.

Chapter 5

Estimation of Camera Parameters

Camera calibration using vanishing points has been widely explored, [Kanatani, 1993] [Wang and Tsai, 1991] [Caprile and Torre, 1990] [Brillault and O'Mahony, 1991] [Li, 1994] amongst others. The novelty in my 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 I require just one vanishing point, the best one can be chosen.

5.1 Image Horizon

In chapter 3 we saw how the horizon can be found by having two distinct vanishing points. Knowing the vertical in the camera's referential and the focal distance, an artificial horizon also can also be traced. 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

$$n_x u + n_y v + n_z f = 0 \tag{5.1}$$

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 is the vanishing line of all levelled planes, parallel to the ground plane.

5.2 Estimation of Focal Distance f

With two vanishing points, corresponding to orthogonal sets of parallel lines, the focal distance can be estimated as shown in chapter 3, (3.18) and (3.19). With one vanishing point $\boldsymbol{p}_v = (u, v)^\mathsf{T}$, obtained from a set of parallel lines belonging to some levelled plane, and from equation (5.1) the unknown scaling factor f in equation (3.2) can be estimated as

$$f = -\frac{n_x u + n_y v}{n_z} \tag{5.2}$$

where $\hat{\boldsymbol{n}} = (n_x, n_y, n_z)^\mathsf{T}$ is taken from (4.8). The projective points are conjugate to each other since they correspond to 3D orthogonal sets of parallel lines. The orthogonality of two levelled plane sets of parallel lines, used in chapter 3, 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 $(u, v, f)^\mathsf{T}$.

With a suitable calibration target scene, where ground plane parallel lines can be easily found, the focal distance can be estimated using (5.2).

5.3 Determining the Vanishing Points

In order to use the above equation, the vanishing points of sets of lines have to be determined. Using an image with dominant ground plane parallel lines, the lines have to be detected and intersected so that the vanishing points can be found.

The edges in the image are found with a modified Sobel filter as described in section 3.2.1. The Hough transform is used to group the edge points into lines, as described in section 3.2.2. Since the Sobel filter provides a local edge orientation estimate, the orientation-based fast Hough transform can be used, avoiding the high computational effort of the full parameter space transform.

Having identified the four lines, that must be dominant in the image, they are then sorted by angle, so that they can be paired, intersected, and the vanishing points found. Notice that in my tests I used four lines so that I could obtain two vanishing points and

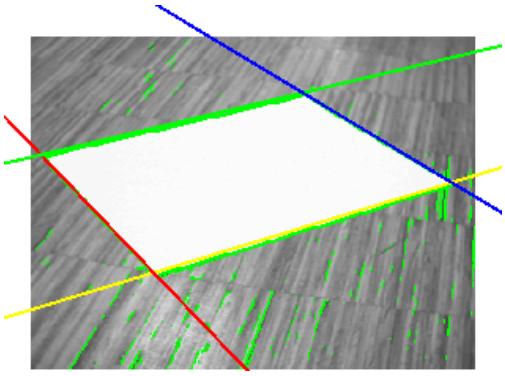


Figure 5.1: Image with dominant ground plane square and detected edges

compare my method with others, but two suffice, since only one vanishing point is needed in my method. However multiple vanishing points should be detected so that the best one can be chosen. Multiple lines for each vanishing point could also be used to improve accuracy, with methods such as the cascaded Hough transform, used by Lutton [Lutton et al., 1994] and Tuytelaars [Tuytelaars et al., 1998].

5.4 Results

A set of tests was performed to verify the validity of the method and quality of the estimation of f. Initially the stereo rig was pointed at a clear desk top with a white paper square as a calibration target, as shown in figure 5.2.

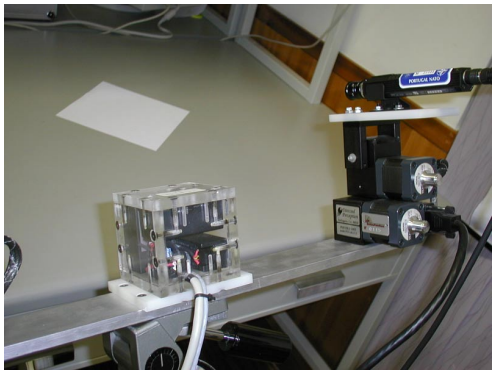


Figure 5.2: Experimental setup with calibration target.

The target was rotated through different positions to evaluate the performance of the method. The square allowed the determination of two distinct vanishing points. The focal distance f is estimated using either one of the vanishing points and the inertial vertical reference. An estimate is also done with just the two vanishing points for comparison. The calibration target was rotated through 19 positions, with 10 samples taken at each position.

Figure 5.3 shows the output of the developed calibration software, which besides logging the estimated values, shows graphically the reconstructed artificial horizon, the vertical reference and the vanishing points. The results are presented in the chart 5.4 and summarized in table 5.1.

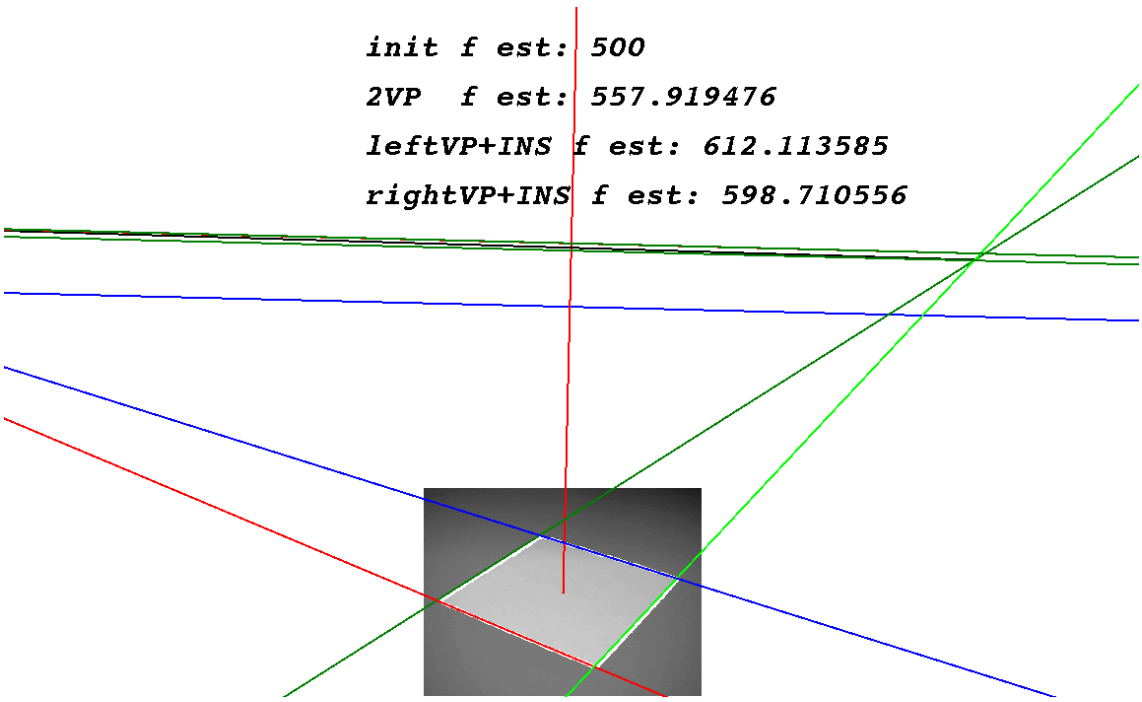


Figure 5.3: Target image and estimated values of f. The vanishing lines are shown, as well as the nearer vanishing point. The lower horizon is based on an initial estimate of f and $\boldsymbol{n}$, and the others are based on the the left and right vanishing points and $\boldsymbol{n}$.

As seen in figure 5.4 and table 5.1, using the best vanishing point and $\hat{\boldsymbol{n}}$ provides the estimate with lower error. The best vanishing point is the one nearer the image centre, which is less prone to instability in its determination.

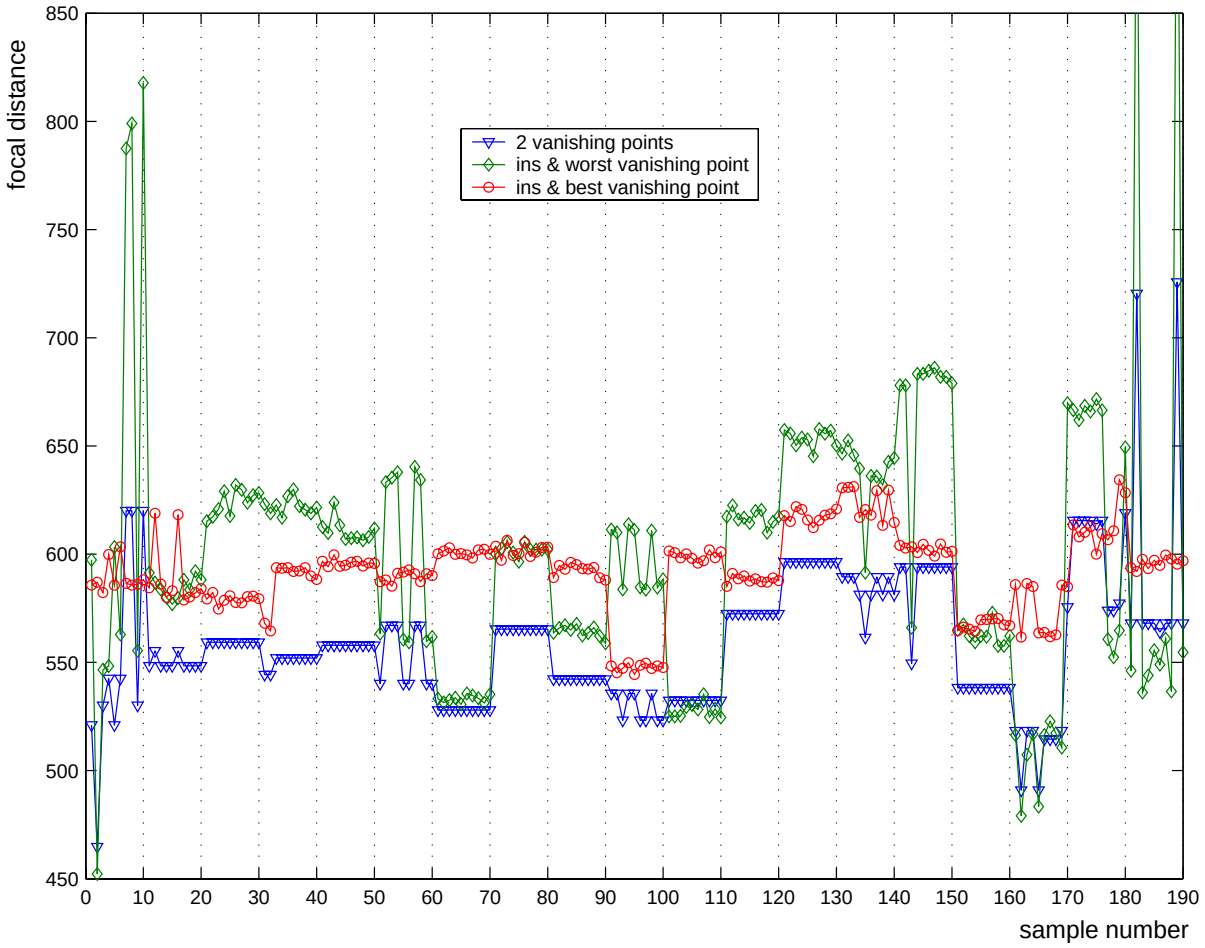


Figure 5.4: Estimation of f with just one vanishing point and $\hat{\boldsymbol{n}}$, compared with estimation based on 2 vanishing points. The target was rotated through 19 positions, with 10 samples taken at each position.

	mean	σ (standard deviation)
2 vanishing points	558.94	32.11
$\hat{\boldsymbol{n}}$ & worst vanishing point	601.23	66.33
$\hat{\boldsymbol{n}}$ & best vanishing point	592.38	18.17

Table 5.1: Estimation of f with just one vanishing point and $\hat{\boldsymbol{n}}$, compared with estimation based on 2 vanishing points. The target was rotated through 19 positions, with 10 samples taken at each position.

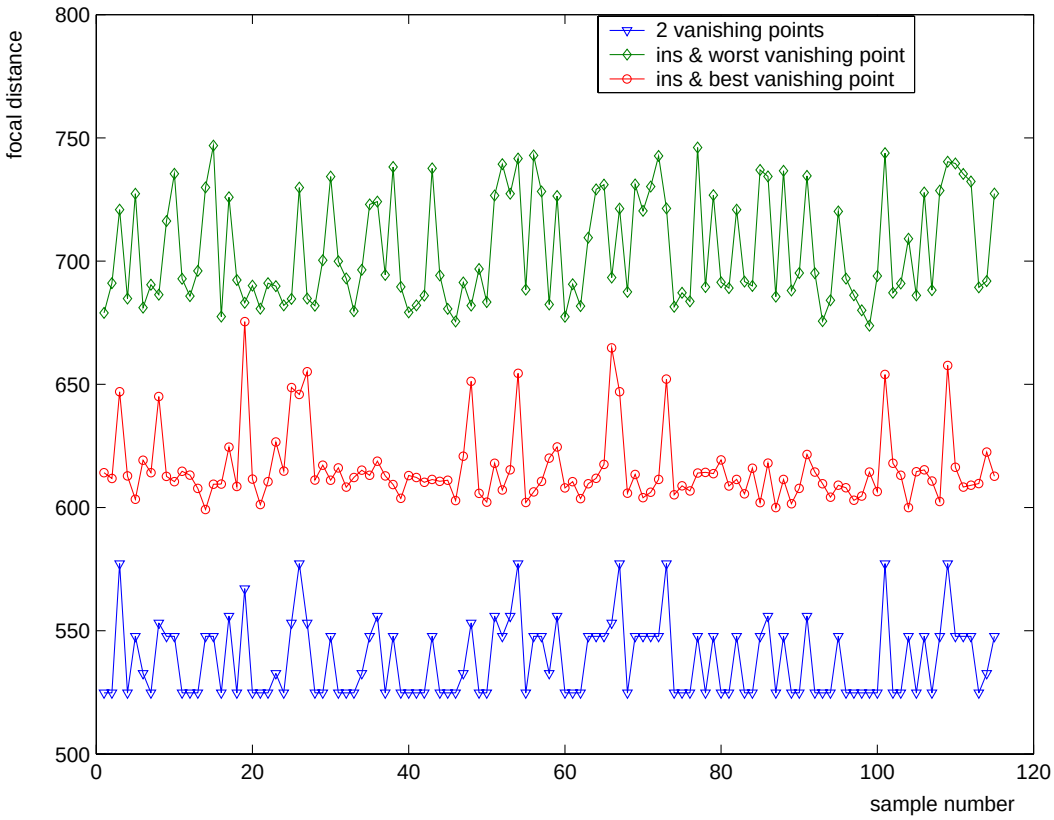


Figure 5.5: Estimation of f with just one vanishing point and $\hat{\boldsymbol{n}}$, compared with estimation based on 2 vanishing points. The target was fixed and 100 samples taken.

	mean	σ (standard deviation)
2 vanishing points	538.30	15.68
$\hat{\boldsymbol{n}}$ & worst vanishing point	704.50	22.46
$\hat{\boldsymbol{n}}$ & best vanishing point	615.80	14.90

Table 5.2: Estimation of f with just one vanishing point and $\hat{\boldsymbol{n}}$, compared with estimation based on 2 vanishing points. The target was fixed and 100 samples taken.

Figure 5.5 and table 5.2 show the estimate results for 100 samples of a single position, favourable for the two vanishing points. Although the vanishing points are both near the image, one is nearer and using this vanishing point and $\hat{\boldsymbol{n}}$ still provides the estimate with smaller error.

After these initial tests, that proved the validity of the method a more complete test was done. The camera calibration toolbox provided by Intel Open Source Computer Vision Library [Intel, 2001] was used to provide a standard calibration method. The calibration recovers the camera's intrinsic parameters as well as the target positions relative to the camera. The calibration algorithm is based on Zhang's work in estimation of planar homographies for camera calibration [Zhang, 1999], 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 not estimated at the initialisation phase.

The calibration was performed with 20 images of a checkerboard target in several positions. Figure 5.6 shows one of the images used, with the image points used and reprojected grid points, and the reconstructed 3D target positions.

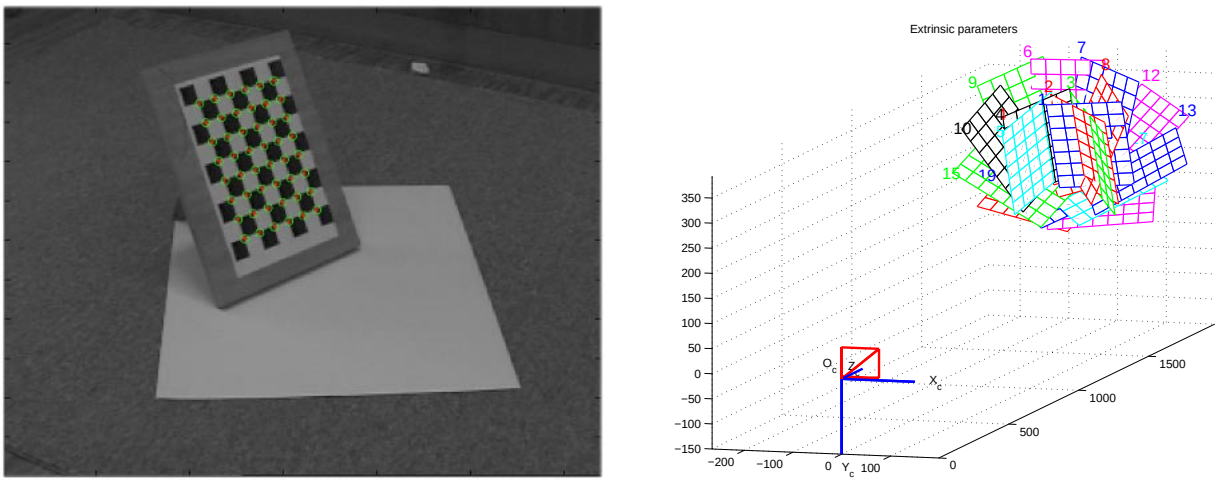


Figure 5.6: One of the 20 images used in the calibration with the image points used (+) and reprojected grid points (o), and reconstructed 3D target positions.

Without changing the camera, the checkerboard target was removed and the calibration was performed using just one vanishing point and the inertial vertical reference. Two

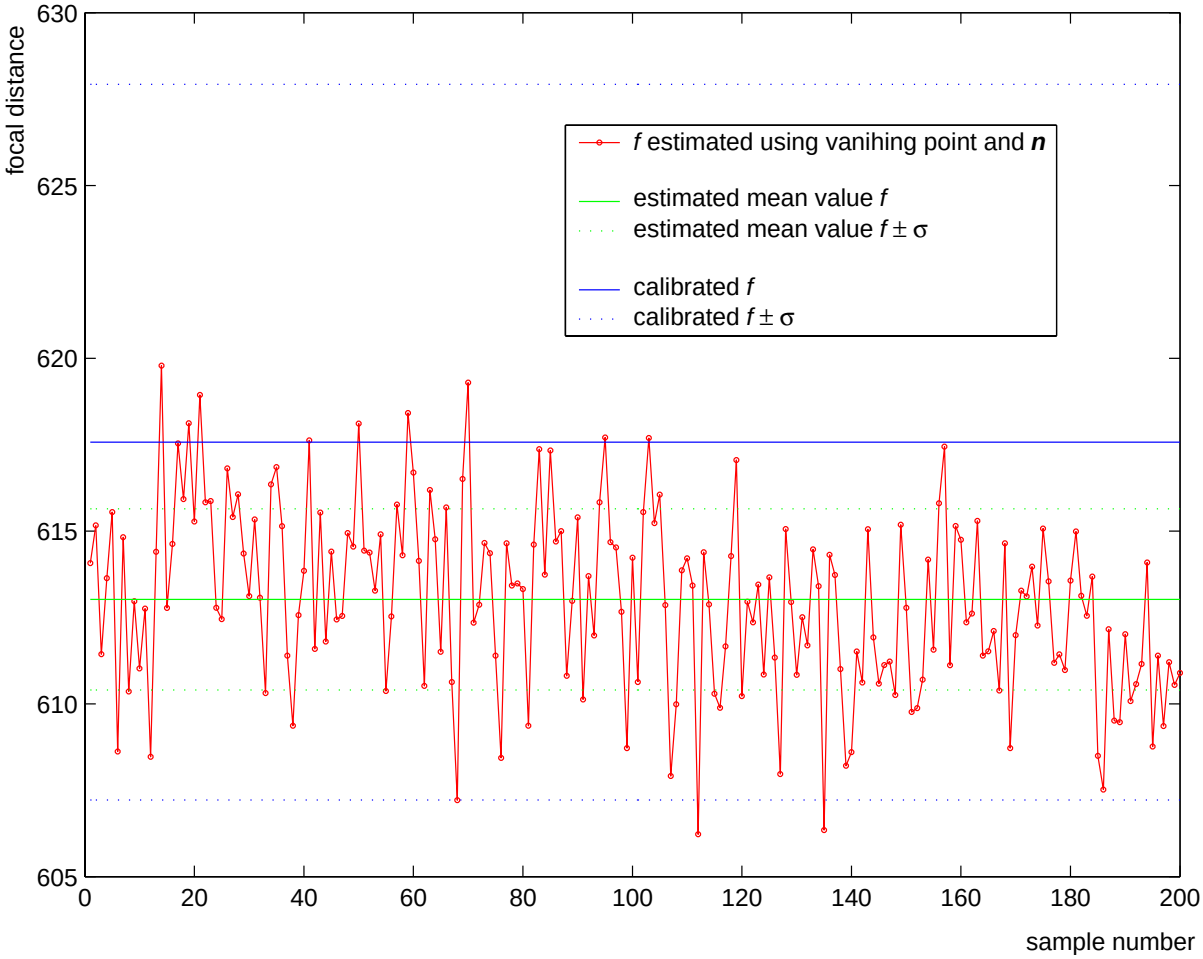


Figure 5.7: Estimation of f with just one vanishing point and $\hat{\boldsymbol{n}}$, compared with camera calibration results from Intel Open Source Computer Vision Library [Intel, 2001]. Two target positions with near vanishing point, 100 samples taken at each position.

	mean	σ (standard deviation)
20 images of checkerboard target	617.57	10.36
$\hat{\boldsymbol{n}}$ & vanishing point	613.02	2.62

Table 5.3: Estimation of f with just one vanishing point and $\hat{\boldsymbol{n}}$, compared with camera calibration results from Intel Open Source Computer Vision Library [Intel, 2001]. Two target positions with near vanishing point, 100 samples taken at each position.

target positions with near vanishing point were used, as seen in figure 5.8, and 100 samples taken at each position. From figure 5.7 and table 5.3 we can see that the proposed method the results provides a good estimate of f, within the uncertainty of the standard method used.

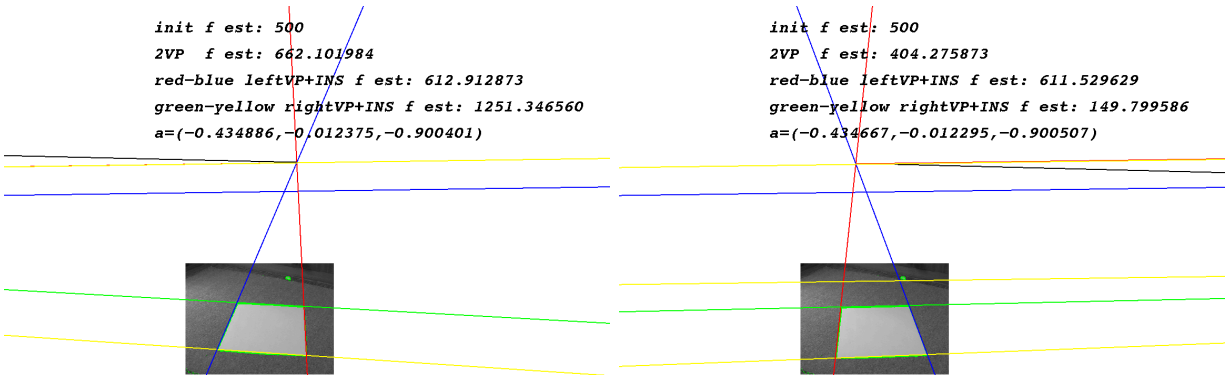


Figure 5.8: Estimation of f at two target positions with near vanishing point.

A test was also performed without a specific calibration target, but known features of the scene. The corridor scene shown in figure 5.9 provides one ground plane vanishing point. The results are raw, without filtering or outlier removal. Table 5.4 shows the result obtained.

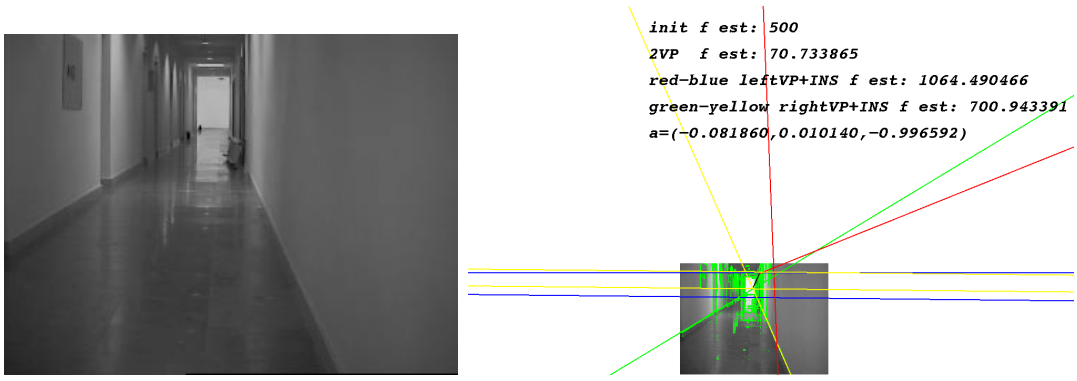


Figure 5.9: Corridor scene used for estimation of f.

Part of this work on focal distance estimation based on vanishing points and the vertical inertial reference, was presented at [Lobo and Dias, 2001].

Table 5.4: Estimation of f using vanishing point and $\hat{\boldsymbol{n}}$ from corridor scene

	mean	σ (standard deviation)
$\hat{\boldsymbol{n}}$ & vanishing point	699.32	15.24

The main sources of error are the vanishing point instability, evidenced by the stepwise results obtained in the first test, 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 when digitally acquiring an image, as in [Coorg, 1998].

Chapter 6

World Feature Detection

Autonomous robots can find use in many applications. Some may require them to work in completely unstructured environments. But many will work in man made environments, where the detection of vertical lines can play a key role in the preliminary perception process. Identifying the ground plane is a crucial part of the mobile robot's navigation and obstacle avoidance tasks. The pose data provided by the inertial sensors enables the extraction of such features from the stereo images.

6.1 Ground Plane

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

$$ {}^{\mathcal{C}}\hat{\boldsymbol{n}}.{}^{\mathcal{C}}\boldsymbol{P} + d = 0 \tag{6.1} $$

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, but it can be determined by the stereo system through a process of visual fixation [Dias et al., 1995] [Dias et al., 1998].

If the active vision system fixates in a point ${}^{\mathcal{C}}\boldsymbol{P}_f$ that belongs to the ground plane, and assuming symmetric vergence (*i.e.*, $\theta = \theta_R = -\theta_L$), the distance d is given by the

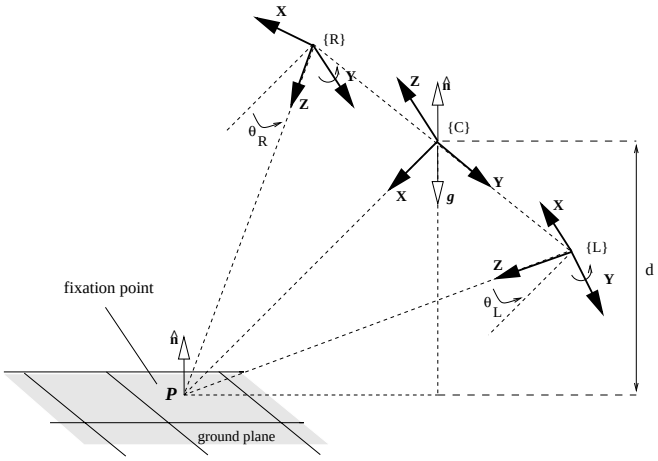


Figure 6.1: Ground plane point fixated

projection of ${}^{\mathcal{C}}\boldsymbol{P}_f$ on the gravity vector direction

$$d = -{}^{\mathcal{C}}\hat{\boldsymbol{n}}.{}^{\mathcal{C}}\boldsymbol{P}_f = -{}^{\mathcal{C}}\hat{\boldsymbol{n}}.\begin{bmatrix} \frac{b}{2}\cot\theta \\ 0 \\ 0 \end{bmatrix} = -n_x\frac{b}{2}\cot\theta \tag{6.2}$$

as can easily be seen in figure 6.1. In this figure there is no lateral inclination, but 6.2 is valid for any angle, since the attitude is given by ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$. This value of d will be used to determine if other points in the image belong or not to the level plane containing the image fixation point.

Figure 6.2 shows a pair of stereo images obtained with the old system, where fixation was obtained for a ground plane point. In this example the inclinometers were used, α_x was null and $\alpha_y = 13.3\ deg$, $\theta_R = -\theta_L = 5.36\ deg$ and b, the baseline distance, was $0.296\ m$. Using equations 4.4 and 6.2 we obtain $d \simeq 0.363\ m$.

The ground plane can therefore be determined in the Cyclop $\{\mathcal{C}\}$ referential, 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.

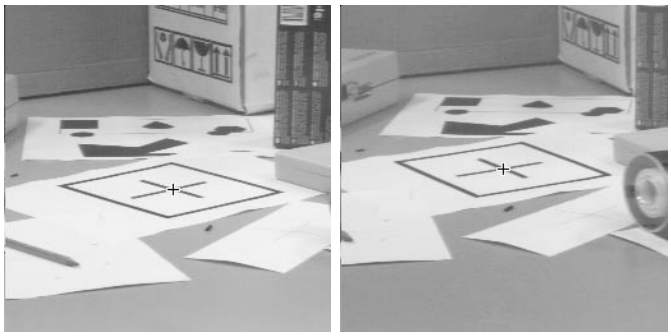


Figure 6.2: Example of ground point fixation.

6.2 Robot Navigation Frame of Reference

When detecting world features, a convenient frame of reference has to be established. With the system mounted on a mobile robot, all detected features should be charted onto the robot's world map, constructed as the robot moves in its environment. It is not convenient to construct this map in the Cyclop $\{\mathcal{C}\}$ referential, and a better choice is to convert the points to a robot navigation frame of reference $\{\mathcal{N}\}$. 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

$$ {}^{\mathcal{N}}\boldsymbol{P} = {}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}\boldsymbol{P} \tag{6.3} $$

where

$$ {}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}} = \begin{bmatrix} \sqrt{1-n_x^2} & \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{-n_x n_z}{\sqrt{1-n_x^2}} & 0 \\ 0 & \frac{n_z}{\sqrt{1-n_x^2}} & \frac{-n_y}{\sqrt{1-n_x^2}} & 0 \\ n_x & n_y & n_z & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}} = \begin{bmatrix} \sqrt{1-n_x^2} & 0 & n_x & -n_x d \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{n_z}{\sqrt{1-n_x^2}} & n_y & -n_y d \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} & \frac{-n_y}{\sqrt{1-n_x^2}} & n_z & -n_z d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.4} $$

This is obtained as follows. Consider a frame of reference $\{\mathcal{N}_c\}$ with origin at the Cyclop 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 referentials as follows

$$^{\mathcal{C}}\boldsymbol{P} = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} .^{\mathcal{N}_c}\boldsymbol{P} = \begin{bmatrix} \hat{\boldsymbol{r}}_1 & \hat{\boldsymbol{r}}_2 & \hat{\boldsymbol{r}}_3 & 0 \\ & 0 & & 1 \end{bmatrix} .^{\mathcal{N}_c}\boldsymbol{P} \tag{6.5}$$

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 Cyclop $\{\mathcal{C}\}$ referential. But the Z axis of $\{\mathcal{N}_c\}$ is just the vertical given by the inertial sensors:

$$^{\mathcal{N}_c}\hat{\boldsymbol{z}} = \hat{\boldsymbol{r}}_3 =^{\mathcal{C}} \hat{\boldsymbol{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \tag{6.6}$$

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:

$$\hat{\boldsymbol{r}}_1 = a.\hat{\boldsymbol{x}} + b.\hat{\boldsymbol{r}}_3 = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} a + bn_x \\ bn_y \\ bn_z \end{bmatrix} \tag{6.7}$$

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

$$||\hat{\boldsymbol{r}}_1|| = a^2 + 2abn_x + b^2 = 1 \tag{6.8}$$

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

$$\hat{\boldsymbol{r}}_1.\hat{\boldsymbol{r}}_3 = 0 = \begin{bmatrix} a + bn_x \\ bn_y \\ bn_z \end{bmatrix}^T \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = n_x a + b = 0 \tag{6.9}$$

From the above equation we get:

$$\hat{\boldsymbol{r}}_1 = \begin{bmatrix} \sqrt{1 - n_x^2} \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} \end{bmatrix} \tag{6.10}$$

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:

$$\hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{r}}_3 \times \hat{\boldsymbol{r}}_1 = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \times \begin{bmatrix} \sqrt{1-n_x^2} \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{n_z}{\sqrt{1-n_x^2}} \\ -\frac{n_y}{\sqrt{1-n_x^2}} \end{bmatrix} \tag{6.11}$$

and so the transformation matrix is given by:

$${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}_c} = \begin{bmatrix} \sqrt{1-n_x^2} & 0 & n_x & 0 \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{n_z}{\sqrt{1-n_x^2}} & n_y & 0 \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} & -\frac{n_y}{\sqrt{1-n_x^2}} & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.12}$$

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 (6.4).

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 seen in the previous chapter, 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 (6.7) with the heading given by the vanishing point $\hat{\boldsymbol{m}}$ we have

$$\hat{\boldsymbol{r}}_1 = \hat{\boldsymbol{m}} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \tag{6.13}$$

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

$$\hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{r}}_3 \times \hat{\boldsymbol{r}}_1 = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \times \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} n_y m_z - n_z m_y \\ n_z m_x - n_x m_z \\ n_x m_y - n_y m_x \end{bmatrix} \tag{6.14}$$

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

$$
{}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}_c} = \begin{bmatrix} m_x & n_y m_z - n_z m_y & n_x & 0 \\ m_y & n_z m_x - n_x m_z & n_y & 0 \\ m_z & n_x m_y - n_y m_x & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.15}
$$

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

$$
{}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}} = \begin{bmatrix} m_x & n_y m_z - n_z m_y & n_x & -n_x d \\ m_y & n_z m_x - n_x m_z & n_y & -n_y d \\ m_z & n_x m_y - n_y m_x & n_z & -n_z d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.16}
$$

and

$$
{}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}} = \begin{bmatrix} m_x & m_y & m_z & 0 \\ n_y m_z - n_z m_y & n_z m_x - n_x m_z & n_x m_y - n_y m_x & 0 \\ n_x & n_y & n_z & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.17}
$$

Providing suitable vanishing points can extracted extracted from the scene, we are able to have $\{\mathcal{N}\}$ coherent with the inertial vertical and the scene heading. The update of $\{\mathcal{N}\}$ as the robot moves along is beyond the scope of this work, where $\{\mathcal{N}\}$ is just relative to the robot's position, and not truly world fixed. Using the robot's odometry, the inertial sensors and landmark matching this update can be accomplished.

6.3 Homography of Ground Plane Points

Consider a world point ${}^{\mathcal{N}}\boldsymbol{P}$ that belongs to the ground plane (i.e., $Z = 0$). We can rewrite the plane equation in (6.1) as

$$
\begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} . {}^{\mathcal{N}}\boldsymbol{P} = 0 \tag{6.18}
$$

and we can express the ground point as

$$^{\mathcal{N}}\boldsymbol{P}_{gnd} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{6.19}$$

To find its projection in the camera image plane we can rewrite equation (3.3) as

$$s\boldsymbol{p}_i = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{6.20}$$

Denoting the i^{th} column of the rotation matrix $\boldsymbol{R}$ by $\boldsymbol{r_i}$ we have

$$s\boldsymbol{p}_i = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{r_1} & \boldsymbol{r_2} & \boldsymbol{r_3} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{r_1} & \boldsymbol{r_2} & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{6.21}$$

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

$$s\boldsymbol{p}_{ri} = \boldsymbol{H_r}.\widetilde{\boldsymbol{P}} \tag{6.22}$$

where $\boldsymbol{p}_{ri}$ is the right projective image point, s an arbitrary scale factor, $\widetilde{\boldsymbol{P}} = \begin{bmatrix} X & Y & 1 \end{bmatrix}^{\mathsf{T}}$ and

$$\boldsymbol{H_r} = \boldsymbol{C_R} \begin{bmatrix} \boldsymbol{r_1} & \boldsymbol{r_2} & \boldsymbol{t} \end{bmatrix}_R \tag{6.23}$$

and similarly we have for the left image

$$s\boldsymbol{p}_{li} = \boldsymbol{H_l}.\widetilde{\boldsymbol{P}} \tag{6.24}$$

with

$$\boldsymbol{H_l} = \boldsymbol{C_L} \left[\begin{array}{ccc} \boldsymbol{r_1} & \boldsymbol{r_2} & \boldsymbol{t} \end{array} \right]_L \tag{6.25}$$

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 [Hartley and Zisserman, 2000], or using equations (6.22) and (6.24), provided that $\boldsymbol{H_r}$ or $\boldsymbol{H_l}$ are invertible. For the direct mapping $\boldsymbol{H}$ of right image points to the left image we have

$$s\boldsymbol{p}_{li} = \boldsymbol{H}.\boldsymbol{p}_{ri} = \boldsymbol{H_l}.\boldsymbol{H_r}^{-1}.\boldsymbol{p}_{ri} \tag{6.26}$$

To obtain $\boldsymbol{H}$ we must first compute $\boldsymbol{H_l}$ and $\boldsymbol{H_r}$. From (6.25) and using ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}}$ obtained from the inertial data and ${}^{\mathcal{L}}\boldsymbol{T}_{\mathcal{C}} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{L}}{}^{-1}$ obtained form the geometric setup, we can write

$$\boldsymbol{H_l} = \boldsymbol{C_L} \left[\begin{array}{ccc} \boldsymbol{r_1} & \boldsymbol{r_2} & \boldsymbol{t} \end{array} \right]_L = \boldsymbol{C_L} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} .{}^{\mathcal{L}}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}}. \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.27}$$

and from the previous equations we obtain

$$\boldsymbol{H_l} = \begin{bmatrix} f\frac{-\sin\theta + n_x^2\sin\theta + n_x n_y\cos\theta}{\sqrt{1-n_x^2}} & -f\frac{n_z\cos\theta}{\sqrt{1-n_x^2}} & fn_x d\sin\theta + fn_y d\cos\theta + \frac{1}{2}fb\cos\theta \\ f\frac{n_x n_z}{\sqrt{1-n_x^2}} & f\frac{n_y}{\sqrt{1-n_x^2}} & fn_z d \\ \frac{\cos\theta - n_x^2\cos\theta + n_x n_y\sin\theta}{\sqrt{1-n_x^2}} & -\frac{n_z\sin\theta}{\sqrt{1-n_x^2}} & -n_x d\cos\theta + n_y d\sin\theta + \frac{1}{2}b\sin\theta \end{bmatrix} \tag{6.28}$$

and proceeding analogously for the right image we obtain

$$\boldsymbol{H_r} = \begin{bmatrix} -f\frac{-\sin\theta + n_x^2\sin\theta - n_x n_y\cos\theta}{\sqrt{1-n_x^2}} & -f\frac{n_z\cos\theta}{\sqrt{1-n_x^2}} & -fn_x d\sin\theta + fn_y d\cos\theta - \frac{1}{2}fb\cos\theta \\ f\frac{n_x n_z}{\sqrt{1-n_x^2}} & f\frac{n_y}{\sqrt{1-n_x^2}} & fn_z d \\ -\frac{-\cos\theta + n_x^2\cos\theta + n_x n_y\sin\theta}{\sqrt{1-n_x^2}} & \frac{n_z\sin\theta}{\sqrt{1-n_x^2}} & -n_x d\cos\theta - n_y d\sin\theta + \frac{1}{2}b\sin\theta \end{bmatrix} \tag{6.29}$$

From (6.26) we have that the homography between left and right images of ground plane points, for the geometry established for our system, is given by

$$\boldsymbol{H} = \boldsymbol{H_l}.\boldsymbol{H_r}^{-1} = \begin{bmatrix} -\frac{2n_x b\cos\theta\sin\theta+n_y b+2d-4d\cos^2\theta}{-n_y b+2d} & \frac{2bn_z\cos\theta}{-n_y b+2d} & f\frac{-2\cos\theta(2d\sin\theta+bn_x\cos\theta)}{-n_y b+2d} \\ 0 & 1 & 0 \\ 2\frac{2d\sin\theta\cos\theta-n_x b+n_x b\cos^2\theta}{f(-n_y b+2d)} & \frac{2bn_z\sin\theta}{f(-n_y b+2d)} & -\frac{2n_x b\cos\theta\sin\theta+n_y b-4d\cos^2\theta+2d}{-n_y b+2d} \end{bmatrix} \tag{6.30}$$

This equation will be fundamental for my world feature detection methods described in following sections. This homography 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 $\hat{\boldsymbol{n}}_{slope}$ given by the robots tilt, and proceed as before.

6.4 Detecting the Ground Plane

As seen in the previous sections, the homography of ground plane image points can be determined from the vertical inertial reference and some system parameters. Using a suitable image interest point detector and the estimated homography, ground plane points can be detected in the stereo pair and mapped in 3D. For on-the-fly visualisation of the segmented images and the detected ground points a VRML (Virtual Reality Modelling Language) viewer is used.

6.4.1 Stereo Correspondence of Ground Plane Points and 3D Position

Since we known the homography of the ground plane image points from equation (6.30), we can parse all detected points and test for matches. An algorithm for the 3D reconstruction of vertical features can now be presented. For each detected point in the right image, map it to the other image using equation (6.30). The correspondent point and its neighbourhood in the left image can then be tested for a match with the original

point of interest in the right image by parsing all the left image lines, determining the corresponding left image point $\boldsymbol{p}_{li}$ with

$$\boldsymbol{p}_{li} = \boldsymbol{H}.\boldsymbol{p}_{ri} \tag{6.31}$$

where $\boldsymbol{p}_{li}$ the point in the right image being parsed, and testing an allowed neighbourhood window $\boldsymbol{p}_{li} \pm \boldsymbol{\delta}$ for a left image feature point. 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. 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 equation 6.22 the 3D position ${}^{\mathcal{N}}\boldsymbol{P}$ of this point is given by

$${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = \boldsymbol{H}_{\boldsymbol{r}}^{-1}\boldsymbol{p}_{ri} \tag{6.32}$$

where ${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = \begin{bmatrix} X & Y & 1 \end{bmatrix}$ and ${}^{\mathcal{N}}\boldsymbol{P} = \begin{bmatrix} X & Y & 0 & 1 \end{bmatrix}$.

With the system mounted on the mobile robot, the ground plane points 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 6.2, and enables the construction of a virtual world rendered in VRML as shown in the following sections.

6.4.2 Results

The above algorithm was implemented with the old system mounted on a mobile robot as seen in figure 6.3.a. The setup is similar to one previously described, but with different cameras and baseline. The vertical reference is taken form the inclinometers. Figure 6.3.b shows a pair of stereo images where fixation was obtained for a ground plane point. In this example α_x is null and $\alpha_y = 16.05^o$, $\theta = 2.88^o$ and $b = 29.6\ cm$. Making $d = b.\cot(\theta)sin(\alpha_y)/2 \simeq 81,3\ cm$. The ground plane is thus determined.

The points shown in figure 6.3.b were obtained using SUSAN [Smith and Brady, 1997] corner detector described in section 3.2.3. The points of interest in the right image were than parsed as described in the previous section. Figures 6.3.c and 6.3.d show the

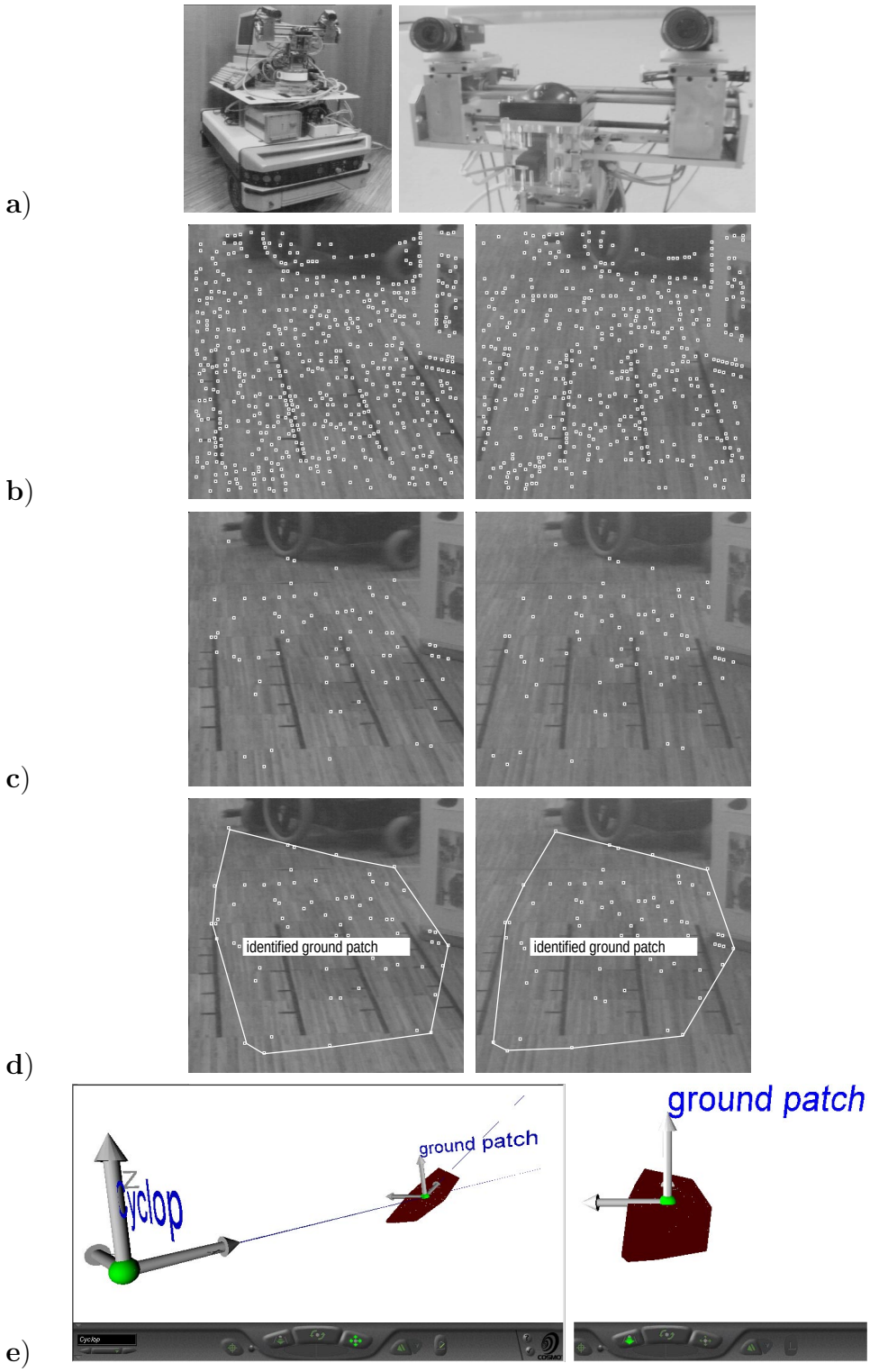


Figure 6.3: Ground plane detection. The old vision system and mobile robot were used as shown in **a**). The system was initially fixated on a ground plane point and the parameters extracted. **b**) Stereo images with a set of initial points. **c**) Detected ground points. **d**) Identified ground patch. **e**) VRML view of ground patch.

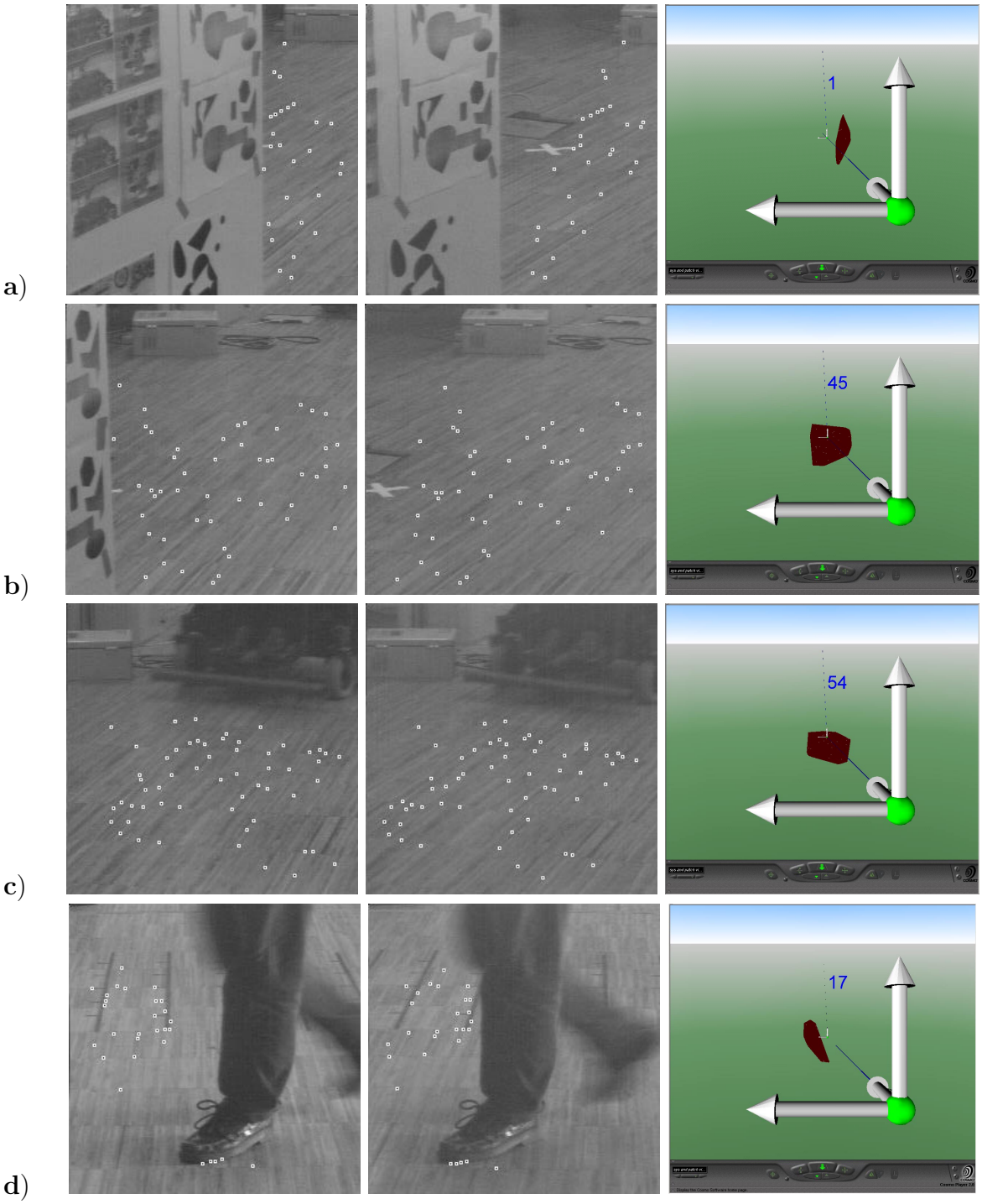


Figure 6.4: Frames from ground plane detection sequence obtained with the old vision system and mobile robot. On the right side is the VRML view of the ground patch. Notice that in frame **d**) an error occurs since the algorithm fits all detected points to a single polygon, ignoring the two distinct clusters of points.

matched ground plane points of interest. Grahams Algorithm [O'Rourke, 1993] was used for computation of the convex polygon involving the set of points. Figure 6.4 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 [Ames et al., 1997]. The identified ground plane patch was mapped onto the 3D scene, as seen in figure 6.3.e. The complete sequence was processed, generating polygons corresponding to the identified ground plane patch for each frame. The VRML images were generated using Netscape Communicator 4.04 web browser with the Cosmo Player 2.0 plug-in from Silicon Graphics, Inc. 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.

As seen in figure 6.4.d, 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. As seen in figure 6.3.b there are many initial 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.

6.5 Detecting Vertical Lines

6.5.1 Image Line Segmentation

Knowing the vertical, the vanishing point of all image lines that correspond to world vertical features is know. 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. 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 normalized image projection of the vertical $\hat{\boldsymbol{n}}$ as defined in equation (4.8). The image vertical reference given in (4.8) corresponds to the unit sphere projection of the vanishing point of all 3D vertical lines in the image plane.

In order to detect vertical lines I extracted the edges in the image using the optimised Sobel filter as described in section 3.2.1. By choosing an appropriate threshold for the gradient magnitude, the potential edge lines can be identified. The square of the gradient was used in my application to allow faster integer computation.

To only obtain the vertical edges I 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.

$$\boldsymbol{\mathcal{D}}.\hat{\boldsymbol{n}}_i < tolerance \tag{6.33}$$

But since this can lead to erroneous results since the pixel gradient provides a very local information and is affected by the pixel quantization, a large tolerance is used. In order to extract the vertical lines in the image, all edge points that satisfied equation (6.33) were mapped to a rectified image table (equation (6.34)), so that continuity could be tested along the vertical edge direction. So each edge point $\boldsymbol{p}_i = (u, v)$ contributed to the table at position

$$vert\_points\,(x, y) = \left(\boldsymbol{p}_i.\hat{\boldsymbol{h}}, \boldsymbol{p}_i.\hat{\boldsymbol{n}}\right) \tag{6.34}$$

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

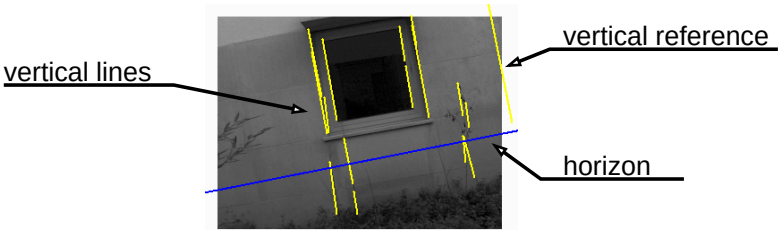


Figure 6.5: Vertical line detection.

$$\hat{\boldsymbol{n}}.\hat{\boldsymbol{h}} = 0 \tag{6.35}$$

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.

I implemented this method with my system, working real time at 10 frames per second. Figure 6.5 shows an example of the results obtained. The results 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.

6.5.2 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 known the homography of the ground plane image points from equation (6.30) 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.

An algorithm for the 3D reconstruction of vertical features can now be presented. For each detected line in the right image, map its *foot* to the other image using equation (6.30). The correspondent point and its neighbourhood in the left image can then be tested for a match with the original point of interest in the right image by parsing all the left image lines, determining the corresponding left image point $\boldsymbol{p}_{li}$ with

$$\boldsymbol{p}_{li} = \boldsymbol{H}.\boldsymbol{p}_{ri} \tag{6.36}$$

where $\boldsymbol{p}_{ri}$ the *foot* of the right image line, and testing an allowed neighbourhood window $\boldsymbol{p}_{li} \pm \boldsymbol{\delta}$ for the *foot* of a left image vertical line. If there is a match, the point belongs to the ground plane and must be the *foot* of a true 3D vertical world feature. From equation 6.22 the 3D position ${}^{\mathcal{N}}\boldsymbol{P}$ of the *foot* of this vertical element is given by

$${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = \boldsymbol{H}_{\boldsymbol{r}}^{-1}\boldsymbol{p}_{ri} \tag{6.37}$$

where ${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = \begin{bmatrix} X & Y & 1 \end{bmatrix}$ and ${}^{\mathcal{N}}\boldsymbol{P} = \begin{bmatrix} X & Y & 0 & 1 \end{bmatrix}$.

With the system mounted on the 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 6.2.

6.5.3 Results

I implemented the vertical world feature detector with my 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. A ground point was fixated by the vision system so that system height could be determined. Figure 6.6 shows a stereo image pair with the identified image vertical lines and the matched *feet* of the true 3D vertical lines.

Figure 6.7 shows another 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. Figure 6.7 shows some of the frames and the experimental setup.

Using (6.37) and (6.3) the vertical features are then charted on a world map. Figure 6.8 and 6.9 show 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

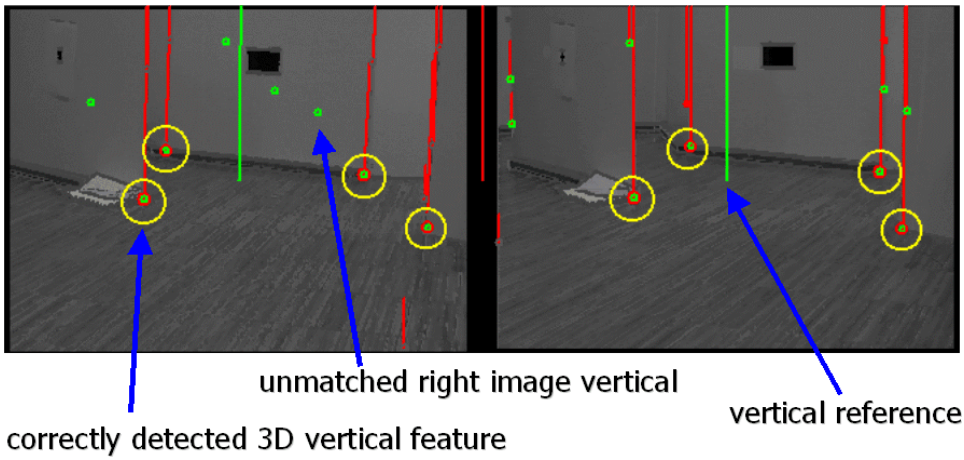


Figure 6.6: Stereo images with detected vertical world features.

image noise. Proper time filtering and outlier removal has to be performed to have a consistent map. The map has than to be updated the robot moves in its environment, but this was not implemented in my work. Part of this work was presented at [Lobo et al., 2001] and ongoing work at my lab is being done in the mapping [Queiroz et al., 2001].

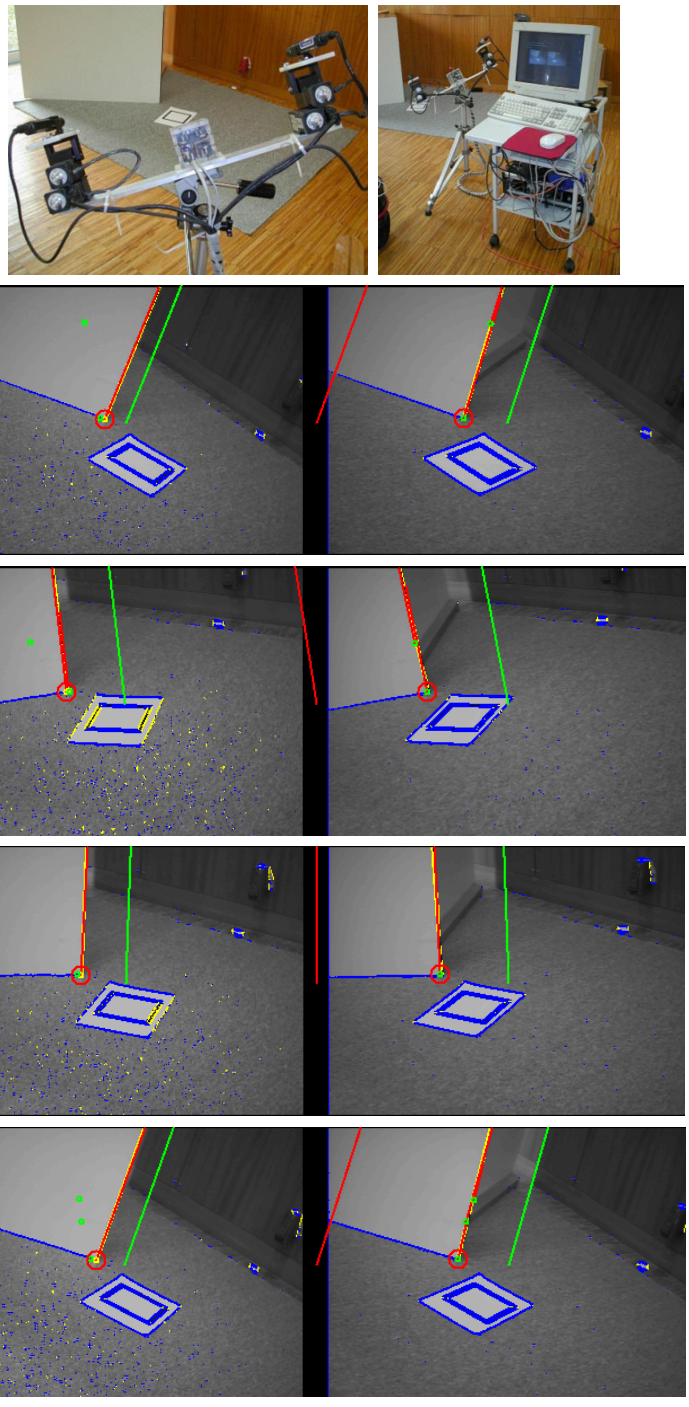


Figure 6.7: Vertical world feature detection. The experimental setup, **a**), was initially fixated on a ground plane point and the parameters extracted. Keeping a constant height, the system was tilted sideways, **b**), **c**), **d**) and **e**) show the results. The vertical reference is shown for the left and right image (notice their difference due to the verge angle). The larger 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.

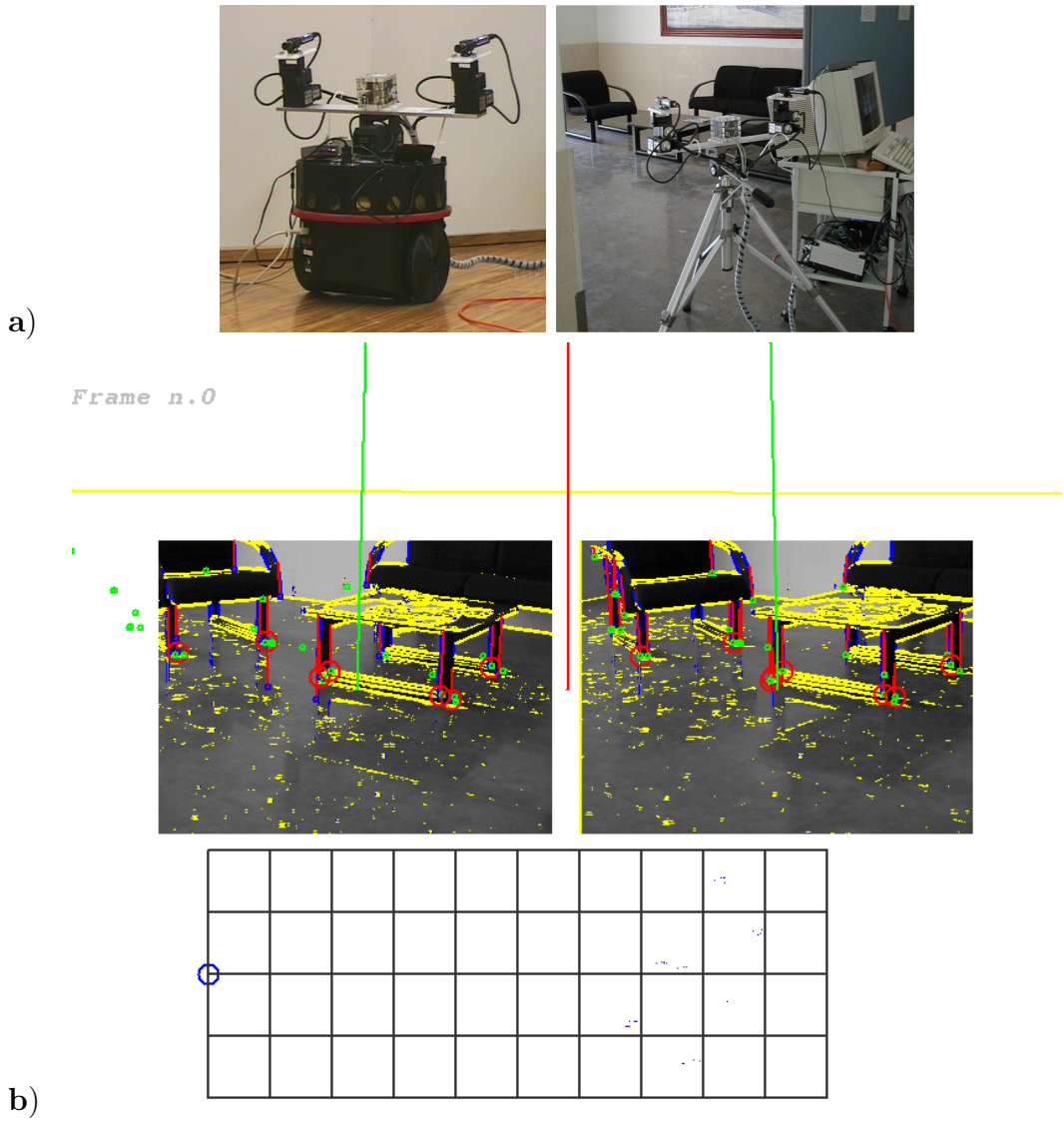
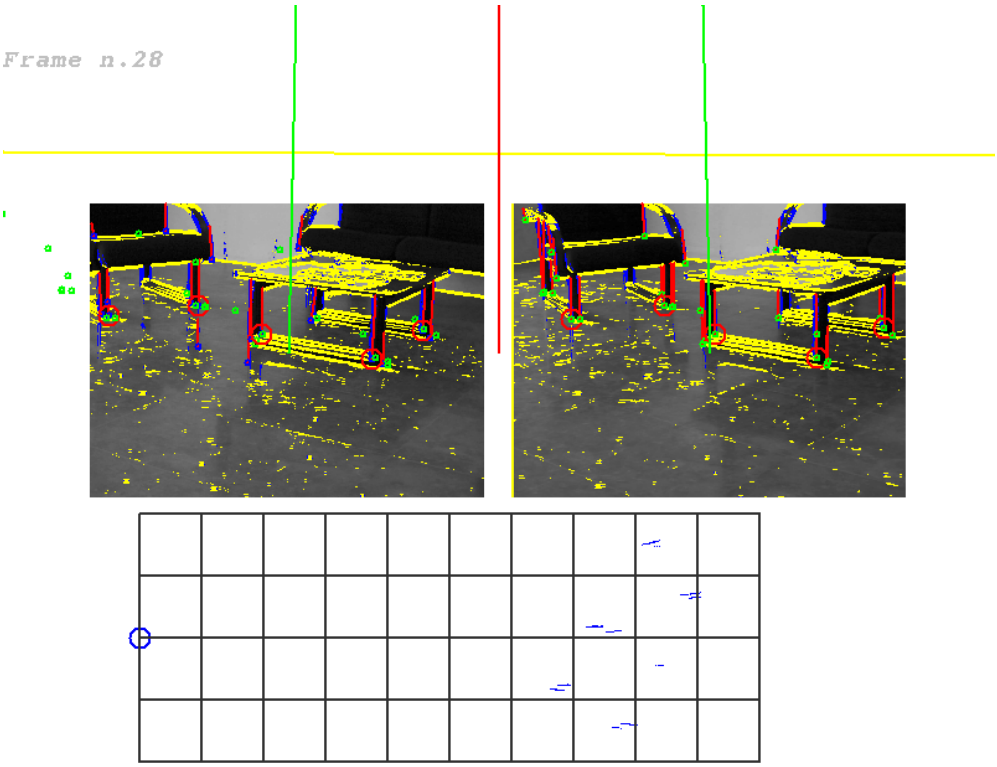
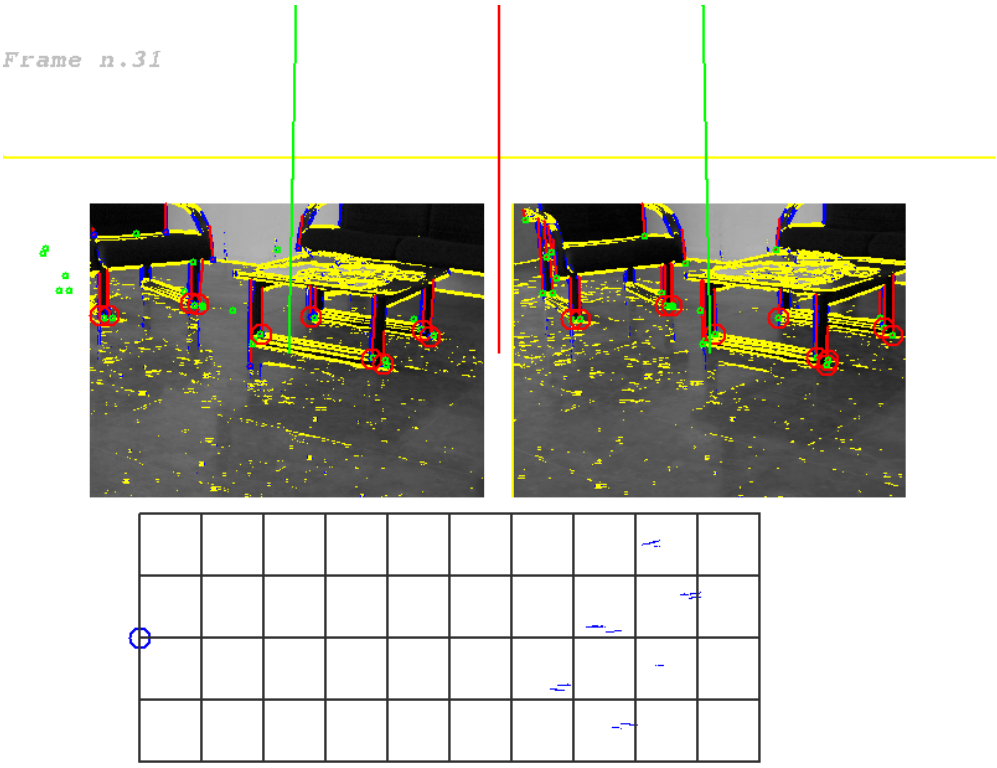


Figure 6.8: Vertical world feature detection with map. **a**) The system was placed on a mobile robot and placed in the entry hall of our lab. The system was initially fixated on a ground plane point and the parameters extracted. **b**), **c**), **d**) and **e**) show the results. The larger circles indicate the *foot* of a detect vertical world feature, the smaller circles the points tested, *i.e.*, the lower end of image vertical lines. The circle in the map represents the robot, and the points the detected vertical world features.



c)



d)

Figure 6.9: Vertical world feature detection with map (cont. figure 6.8).

Chapter 7

Conclusions and Future Work

With this work I was able to explore the integration of inertial data in computer vision systems. From the studies on human vision it is clear that inertial cues play an important role. The notion of vertical is important at the first stages of image processing. Based on these concepts I was able to implement a focal distance estimation method, and to detect 3D world features such as the ground plane and vertical features. These features can be used to build a *metric map* useful to improve mobile robot autonomy and navigation.

I overviewed the available low-cost inertial sensors, and built a system with an inertial system prototype and verging cameras to carry out my experiments. The inertial system prototype was used at a very basic level. It served the purpose of testing concepts and algorithms. Sensor error models must be used as in [Barshan and Durrant-Whyte, 1995] and suitably fused to obtain better estimates for attitude. The integration of magnetoresistive sensors should also be pursued, since they provide an independent external attitude reference. The focus will not be in developing a stand-alone navigation system, but rather an improved artificial vestibular system that can aid the vision system in both static and dynamic situations.

The unit sphere model I used provides a good framework for integration of scene recovered projective data and inertial references. The camera's focal distance is a key factor for scene metric perception and reconstruction. Having a suitable target or known real scene elements, I was able to estimate this parameter using one vanishing point and the vertical reference provide by the inertial sensors. The orthogonality between the

vanishing point direction and the vertical is explored, similarly to the method that used two vanishing points corresponding to orthogonal directions.

The results obtained show that the proposed method is feasible. The main sources of error are the vanishing point instability, evidenced by some stepwise results obtained, and the noise in the vertical reference provided by the low cost accelerometers. It provides a reasonable estimate for a completely uncalibrated camera. The advantage over using two vanishing points is that the best (*i.e.*, the more stable) vanishing point can be chosen, and that in indoors environment the vanishing point point can sometimes be obtained from the scene without placing any specific calibration target. 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 when digitally acquiring an image, as in [Coorg, 1998], so that images with just one good vanishing point can be used in the reconstruction. Part of this work on focal distance estimation, based on vanishing points and the vertical inertial reference, was presented at [Lobo and Dias, 2001]. Future work involves improving the vanishing point detection, exploring the methods used by Palmer [Palmer et al., 1993], Lutton [Lutton et al., 1994] and Tuytelaars [Tuytelaars et al., 1998], so that my single vanishing point approach can benefit. Appropriate statistical error models and sensitivity analysis must be performed as well as tests with diverse indoor scenes.

Stereo vision has to deal with matching features between the left and right images, so that 3D position can be determined. If the detected points are restricted to a plane, we can consider a direct mapping, or homography $\boldsymbol{H}$, of points between the stereo pair. $\boldsymbol{H}$ can be obtained by calibration using know ground plane points [Hartley and Zisserman, 2000], or computed from known geometric parameters of the setup and the vertical inertial reference, as I did in my work. Direct calibration of $\boldsymbol{H}$ provides a better estimate, but my method is simpler to implement, update, and provides a suitable estimate for the ground plane and vertical feature detectors. Having the system parameters also allows me to correctly map the features to a 3D world map.

The ground plane detector parses image feature points and identifies the ones that belong to the ground plane. A convex polygon is then fit to the points to segment a ground patch. Some changes have to be made to the algorithm and special cases taken

into account, such as having multiple isolated polygons 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. Many false negatives occur and assumptions have to be made 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. Part of this work was presented at [Lobo and Dias, 1997],[Lobo et al., 1998] and [Lobo and Dias, 1998].

The detection of vertical lines can play a key role in the mobile robot preliminary perception process. The vertical inertial reference enables the detection of image vertical lines. Making the assumption that the relevant world vertical features start from the ground plane, and using the homography of the ground plane image points I was able to identify a common unique point, the lower point or *foot* of each vertical feature in one image should map to the corresponding *foot* in the other image. The results showed that the 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. 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 the robot moves in its environment, but this was not implemented in this work. Part of this work was presented at [Lobo et al., 2001] and ongoing work is being done in the mapping [Queiroz et al., 2001]. The results presented are preliminary, and further work is required in improving them. The vertical line detection can be improved by considering the known vertical vanishing point, removing the small tilt restriction, but increasing the computational complexity.

In this work I explored some aspects of inertial and vision sensing combination. Figure 7.1 shows a framework of possible combinations, identifying the contributions from each sensing modality, and summarising the results obtained in this work. The two sensing modalities overlap since camera rotation and translation can be directly measured by inertial sensors, or inferred from the image flow and known scene features. Combining

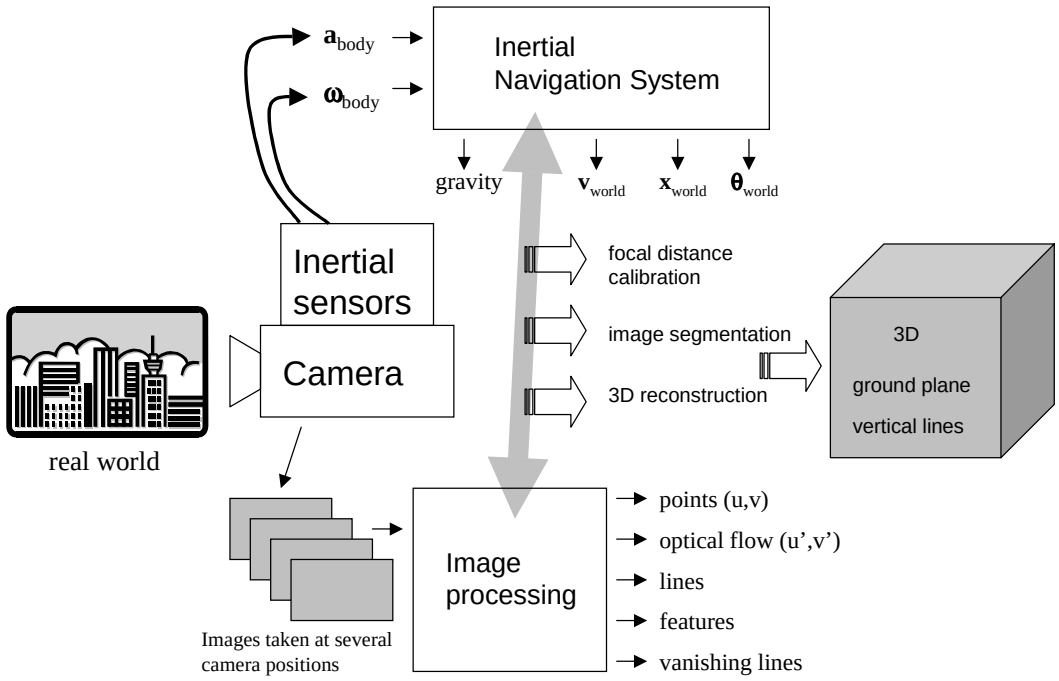


Figure 7.1: Framework of possible combinations of inertial and vision sensing

information can lead to better calibration, image segmentation and 3D structure recovery, as shown in my work.

Current work is being done on more dynamic situations, such as using optical flow and inertial data for dense depth map estimation. The image of optical flow due to camera's self motion is given by

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{1}{Z}.A.\boldsymbol{v} + B.\boldsymbol{\omega} \tag{7.1}$$

where $\boldsymbol{v}$ is the linear velocity of the camera optical centre obtained by integrating acceleration over time and $\boldsymbol{\omega}$ the angular velocity given by the gyrometers. The inertial cues can be used at a predictive stage for a better optical flow estimation, as happens in human vision. With the ego-motion parameters provided by the inertial sensors and the computed optical flow, depth can be estimated.

References

Advantech (2001). http://www.advantech.com.

Allen, H. V., Terry, S. C., and Knutti, J. W. (1989). Understanding Silicon Accelerometers. *Sensors*.

Allen, J., Kinney, R., Sarsfield, J., Daily, M., Ellis, J., Smith, J., Montague, S., Howe, R., Boser, B., Horowitz, R., Pisano, A., Lemkin, M., Clark, W., and Juneau, T. (1998). Integrated Micro-Electro-Mechanical Sensor Development for Inertial Applications. In *Proceedings of the 1998 Position Location and Navigation Symposium*.

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

Analog Devices (1996). *ADXL05 Datasheet*.

Analog Devices (2001). Integrated micro-electromechanical systems, analog devices. http://www.analogdevices.com/industry/iMEMS/.

Ausman, J. S. (1962). Theory of Inertial Sensing Devices. In Pitman, G. R., editor, *Inertial Guidance*, pages 72–91. John Wiley & Sons.

Barshan, B. and Durrant-Whyte, H. (1994a). Evaluation of a Solid-Sate Gyroscope for Robotics Applications. *IEEE Transactions on Instrumentation and Measurement*, 44(1):61–67.

Barshan, B. and Durrant-Whyte, H. (1995). Inertial Navigation Systems for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342.

Barshan, B. and Durrant-Whyte, H. F. (1994b). Orientation Estimate for Mobile Robots using Gyroscopic Information. In *Proceedinds of the 1994 IEEE International Conference on Intelligent Robots and Systems*, pages 1867–1874.

Base&SPP (1996). *Motion Sensors*. British Aerospace Systems and Equipment, Sumitomo Precision Products. Product Literature.

Base&SPP (2001). British aerospace systems and equipment, sumitomo precision products. http://www.base-usa.com/sensor/index.htm.

Berthoz, A. (2000). *The Brain's Sense of Movement*. Havard University Press. ISBN: 0-674-80109-1.

Bhanu, B., Roberts, B., and Ming, J. (1990). Inertial Navigation Sensor Integrated Motion Analysis for Obstacle Detection. In *Proceeding of the 1990 IEEE International Conference on Robotics and Automation*, pages 954–959, Cincinnati, Ohio, USA.

Bortz, J. E. (1971). A New Mathematical Formulation for Strapdown Inertial Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-7(1):61–66.

Bres, S. and Jolion, J.-M. (1999). Detection of Interest Points for Image Indexation. In *3rd. Int. Conf. on Visual Information Systems*, pages 427–434, Amsterdam.

Brillault, B. and O'Mahony (1991). New Method for Vanishing Point Detection. *CVGIP: Image Understanding*, 54(2):289–300.

Caprile, B. and Torre, V. (1990). Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, 4(2):127–140.

Carpenter, H. (1988). *Movements of the Eyes*. London Pion Limited, 2nd edition. ISBN 0-85086-109-8.

Clark, W. A., Howe, R. T., and Horowitz, R. (1996). Surface Micromachined Z-axis Vibratory Rate Gyroscope. In *Tech. Dig. Solid-State Sensor and Actuator Workshop*, pages 283–287.

Cline Labs (2001). http://www.clinelabs.com.

Collinson, R. (1996). *Introduction to Avionics*. Chapman & Hall. ISBN 0-412-48250-9.

Coorg, S. R. (1998). *Pose Imagery and Automated Three-Dimensional Modeling of Urban Environments*. PhD thesis, Massachusetts Institute of Technology.

Coren, S., Ward, L. M., and Enns, J. T. (1994). *Sensation and Perception*. Harcourt Brace & Company, fourth edition edition. ISBN 0-15-500103-5.

Crossbow (2001). Crossbow technology. http://www.xbow.com.

Deriche, R. and Goraudon, G. (1993). A Computational Approach for Corner and Vertex Detection. *Int. Journal of Computer Vision*, 10(2):101–124.

Dias, J., Paredes, C., Fonseca, I., Araujo, H., Baptista, J., and de Almeida, A. T. (1998). Simulating Pursuit with Machine Experiments with Robots and Artificial Vision. *IEEE Transactions on Robotics and Automation*, 3(1):1–18.

Dias, J., Paredes, C., Fonseca, I., and de Almeida, A. T. (1995). Simulating Pursuit with Machines. In *Proceedings of the 1995 IEEE Conference on Robotics and Automation*, pages 472–477, Japan.

Directed Perception (2001). http://www.dperception.com.

Encyclopaedia Britannica (2001). www.britannica.com.

Entran (2001). Entran devices. http://www.entran.com.

Everett, H. (1995). *Sensors for Mobile Robotics*. A.K. Peters. ISBN 1-56881-048-2.

FreeBSD (2001). http://www.freebsd.org.

Gillingham, K. K. and Previc, F. H. (1996). *Spatial Orientation in Flight*, chapter 11. Williams and Wilkins, second edition.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge Univercity Press. ISBN: 0521623049.

Hitachi Cable (2001). http://www.hitachi-cable.com.

ICSensors (1988). *Silicon Accelerometers*. Technical Note TN-008.

ICSensors (1995). *Model 3255 Accelerometer*. Application Note TN-010.

ICSensors (2001). http://www.msiusa.com/icsensors.

Intel (2001). Intel Open Source Computer Vision Library. http://www.intel.com/ research/ mrl/research/opencv/.

Jahne, B. (1997). *Digital Image Processing*. Springer-Verlag. ISBN 3-540-62724-3.

Jewell Instruments (2001). http://www.jewellinstruments.com.

Kanatani, K. (1993). *Geometric Computation for Machine Vision*. Oxford University Press. ISBN 0-19-856385-X.

Kelly, A. (1994). Modern Inertial and Satelllite Navigation Systems. Technical Report CMU-RI-TR-94-15, Carnegie Mellon University.

Komoriya, K. and Oyama, E. (1994). Position Estimation of a Mobile Robot Using Optical Fiber Gyroscope (OFG). In *Proceedings of the 1994 IEEE International Conference on Intelligent Robots and Systems*, pages 143–149.

Kurazume, R. and Hirose, S. (2000). Development of image stabilization system for remote operation of walking robots. In *Proceedings of the 2000 IEEE Inernational Conference on Robotics and Automation*, pages 1856–1860, San Francisco, CA, USA.

Kuritsky, M. M. and Goldstein, M. S. (1990). *Inertial Navigation*, pages 96–116. Autonomous Robot Vehicles. Springer-Verlag New York.

Lawrence, A. (1998). *Modern Inertial Tecnology: Navigation, Guidance, and Control*. Mechanical Engineering Series. Springer-Verlag, second edition. ISBN 0-387-98507-7.

Lemkin, M., Ortiz, M., Wongkomet, N., Boser, B. E., and Smith, J. H. (1997). A 3-axis surface micromachined Sigma-Delta accelerometer. *ISSCC Digest of Technical Papers*, pages 202–203.

Li, M. (1994). Camera calibration of the kth head-eye system. In *ECCV94*, pages A:543–554.

Lobo, J. (2001). Inertial Sensors Home Page. http://www.deec.uc.pt/˜ jlobo/ inersense/ inersense.html.

Lobo, J. and Dias, J. (1997). Integration of Inertial Information with Vision towards Robot Autonomy. In *Proceedings of the 1997 IEEE International Symposium on Industrial Electronics*, pages 825–830, Guimaraes, Portugal.

Lobo, J. and Dias, J. (1998). Ground plane detection using visual and inertial data fusion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS'98*, pages 912–917, Victoria, Canada.

Lobo, J. and Dias, J. (2001). 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.

Lobo, J., Marques, L., Dias, J., Dias, U., and de Almeida, A. T. (1998). *Sensors for Mobile Robot Navigation*, pages 50–81. Number 236 in Autonomous Robotic Systems - (Lecture Notes in Control and Information Sciences). Springer-Verlag. ISBN 1-85233-036-8.

Lobo, J., Queiroz, C., and Dias, J. (2001). 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.

Lucas (2001). Lucas novasensor aquired by varity. http://www.lucasvarity.com.

Luther, A. C. (1998). *Video Camera Technology*. Artech House Publishers.

Lutton, E., itre, H. M., and Lopez-Krahe, J. (1994). Contribution to the Determination of Vanishing Points Using Hough Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):430–438.

Matrox (2001). http://www.matrox.com.

Murata (1991). *Piezoelectric Vibrating Gyroscope GYROSTAR*. Cat. No. S34E-1.

Murata Electronics (2001). http://www.murata.com.

Nebot, E. and Durrant-Whyte, H. (1992). Initial calibration and alignment of low-cost inertial navigation units for land vehicle. *Journal of Robotic Systems*, 16(2):81–92.

Nomadic Technologies (2001). http://www.robots.com.

O'Rourke, J. (1993). *Computational Geometry in C*. Cambridge University Press. ISBN 0-512-22592-2.

Palmer, P., Kittler, J., and Petrou, M. (1993). Accurate Line Parameters from an Optimising Hough Transform for Vanishing Point Detection. In *Proceedings of the Fourth Inetrnational Conference on Computer Vision*, pages 529–533, Berlin, Germany.

Panerai, F., Metta, G., and Sandini, G. (2000). Visuo-inertial stabilization in space-variant binocular systems. *Robotics and Autonomous Systems*, 30(1-2):195–214.

Panerai, F. and Sandini, G. (1997). Visual and Inertial Integration for Gaze Stabilization. In *Proc. SIRS'97*, Stockholm.

Panerai, F. and Sandini, G. (1998). Oculo-Motor Stabilization Reflexes: Integration of Inertial and Visual Information. *Neural Networks*, 11(7-8):1191–1204.

Pitman, G. R. (1962). *Inertial Guidance*. John Wiley & Sons.

Poynton, C. A. (1996). *A Technical Introduction to Digital Video*. John Wiley and Sons. ISBN 0-471-12253-X.

Queiroz, C., Lobo, J., and Dias, J. (2001). World Feature Detection and Mapping using Stereovision and Inertial Sensors. In *Proceedings of the Fourth European Workshop on Advanced Mobile Robots*, pages 67–74, Lund, Sweden.

Savage, P. G. (1984). *Strapdown System Algorithms*, chapter 3, pages 3.1–3.30. Lecture Series 133. AGARD, Advisory Group for Aerospace Research and Development.

Schmid, C., Mohr, R., and Bauckhage, C. (1998). Comparing and evaluating interest points. In *Proceeding of the sixth International Conference on Computer Vision*, Bombay, India.

Shuster, M. D. (1993). The Kinematic Equation for the Rotation Vector. *IEEE Transactions on Aerospace and Electronic Systems*, 29(1):263–267.

Slater, J. M. (1962). Principles of Operation of Inertial Sensing Devices. In Pitman, G. R., editor, *Inertial Guidance*, pages 47–71. John Wiley & Sons.

Smith, S. and Brady, J. (1997). SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23(1):45–78.

Stolfi, J. (1991). *Oriented projective geometry, a framework for geometric computations* . Boston Academic Press.

Summit Instruments (1994). *34100A Theory of Operation*. Technical Note 402.

Summit Instruments (2001). http://www.summitinstruments.com.

Systron Donner (1995). *GyroChip*. Systron Donner Inertial Division. Product Literature.

Systron Donner (2001). Systron donner inertial division. http://www.systron.com.

Texas Instruments (2001). http://www.ti.com.

Tilt Measurement (2001). http://www.tilt-measurement.com.

Tuytelaars, T., Proesmans, M., and van Gool, L. (1998). The Cascaded Hough Transform. In *ICIP Proceedings*, pages 736–739.

Vaganay, J., Aldon, M. J., and Fournier, A. (1993). Mobile Robot Attitude Estimation by Fusion of Inertial Data. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pages 277–282, Atlanta, Georgia, USA.

Viéville, T. (1997). *A Few Steps Towards 3D Active Vision*. Springer-Verlag.

Viéville, T., Clergue, E., and Facao, P. (1993a). Computation of ego-motion and structure from visual an inertial sensor using the vertical cue. In *ICCV93*, pages 591–598.

Viéville, T. and Faugeras, O. (1989). Computation of Inertial Information on a Robot. In Miura, H. and Arimoto, S., editors, *Fifth International Symposium on Robotics Research*, pages 57–65. MIT-Press.

Viéville, T. and Faugeras, O. (1990). Cooperation of the Inertial and Visual Systems. In Henderson, T. C., editor, *Traditional and NonTraditional Robotic Sensors*, volume F 63 of *NATO ASI*, pages 339–350. SpringerVerlag Berlin Heidelberg.

Viéville, T., Romann, F., Hotz, B., Mathieu, H., Buffa, M., Robert, L., Facao, P., Faugeras, O., and Audren, J. (1993b). Autonomous navigation of a mobile robot using inertial and visual cues. In Kikode, M., Sato, T., and Tatsuno, K., editors, *Intelligent Robots and Systems*, Yokohama.

Wang, L.-L. and Tsai, W.-H. (1991). Camera Calibration by Vanishing Lines for 3-D Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370–376.

Zhang, Z. (1999). Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 666–673, Kerkyra, Greece.