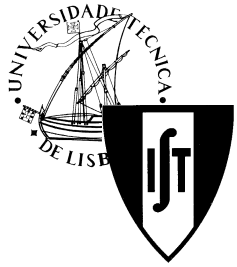

UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

VISUAL PERCEPTION FOR MOBILE ROBOTS :

FROM PERCEPTS TO BEHAVIOURS

JOSÉ ALBERTO ROSADO DOS SANTOS VICTOR

(Mestre)

Tese para obtenção do grau de doutor em

Engenharia Electrotécnica e de Computadores

Lisboa, Novembro de 1994

Tese realizada sob a orientação de

João José dos Santos Sentieiro

Professor Associado do
Departamento de Engenharia Electrotécnica e de Computadores
INSTITUTO SUPERIOR TÉCNICO

Aos meus pais,
à Ana.

AGRADECIMENTOS

Esta tese pertence em parte a um conjunto de pessoas pelo apoio paciência e amizade que sempre revelaram e a quem nunca conseguirei exprimir a minha gratidão.

Ao Prof. João Sentieiro, orientador desta tese, pelo apoio constante e encorajamento. Ao longo dos anos que já passei no Instituto Superior Técnico e ao longo do trabalho de tese foi sempre uma referência de dedicação, entusiasmo e amizade. Obrigado.

Parte do trabalho desta tese foi realizado no Lira-Lab, em Génova, em colaboração com o Prof. Giulio Sandini. Aí fui contagiado pelos seus conhecimentos e grande entusiasmo por estas coisas da Visão Activa. Sem o espírito que soube criar no Lira-Lab, o seu apoio, amizade e inúmeras discussões, esta tese não teria sido possível. Grazie di tutto!

O Grupo de Visão do Instituto de Sistemas e Robótica foi um agradável Fórum para discussões científicas relativas à Visão por Computador, marcadas pelo entusiasmo e amizade. Quero agradecer ao Gaspar, Franc van Trigt, Fernando, e aos colegas mais novos, Alexandre, César, Marco e Victor. Estou certo que este espírito se manterá e alargará no futuro.

Ao meu amigo Franc van Trigt cuja competência, dedicação e entusiasmo permitiram que a Cabeça Stereo fosse uma realidade.

Ao todos os meus amigos do CAPS, pelo ambiente que criaram e que influenciou a realização da tese. Em particular, aos meus amigos e colegas do gabinete 155 : Luis Custódio, Carlos Pinto-Ferreira, Carlos Belo e João Pedro. Ao Pedro Lima pela amizade, inúmeras discussões "cibernéticas" e pela leitura atenta e crítica de uma versão inicial da tese.

A tutti i miei amici del Lira-Lab con cui ho avuto la fortuna di lavorare e che mi hanno aiutato con la loro amicizia. In particolare, a Paolo Questa (e alla Sampdoria!), Francesco Buemi (e alla Rivoluzione!), Giovanni (e alla focaccia col formaggio!), Francesco Panerai, Carla Capurro, Jan Nielsen, Massimo Massa, Fernando, i "Pasa" e Laura. Genova e l'Italia sono state grande esperienze che non dimenticheró mai. Grazie a tutti e arrivederci!

À **aeGIST** pelo apoio à edição desta tese. Em particular aos meus colegas e amigos Horácio, Alexandre, Jorge, Patrícia, Palmira, Miguel e Paula, pelo prazer de trabalharmos juntos por um ideal comum. Foi, é, e será sempre um prazer privar com amigos assim. Obrigado.

A realização de uma tese implica sempre impaciência, muitas noites perdidas, fins de semana estragados e ainda, neste caso, ausências prolongadas. Aos meus pais devo tudo isto e um grande exemplo, que me marcou profundamente, de carácter, carinho, trabalho e dedicação. Para eles vai a minha maior gratidão.

À minha avó, irmão e cunhada pelo apoio constante e compreensão pelas minhas ausências e impaciência. Ao meu sobrinho João Maria, que espero, venha um dia a ler e gostar desta tese.

Por fim, à Ana, pela infinita paciência, carinho e compreensão.

ACKNOWLEDGEMENTS

There is a number of people to whom I will never be able to express my gratitude for their never ending pacience and friendship. Part of this thesis is theirs too.

To Prof. João Sentieiro, my supervisor, for all his support and encouragement. Throughout my years at Instituto Superior Técnico, and during the thesis work, he has always been a reference of dedication, enthusiasm and friendship.

The work described in the thesis was partially developed in Genova, in the Lira-Lab, in collaboration with Prof. Giulio Sandini. I was then "infected" by his profound knowledge and enthusiasm forActive Vision. Without the atmosphere he created in the Lira-Lab, his support and endless discussions, this thesis would have been impossible. Grazie di tutto!

The Vision Group at Instituto de Sistemas e Robótica has been a pleasant Forum for scientific discussions on Computer Vision, marked by the enthusiasm and friendship. I would like to express my gratitude to Gaspar, Franc van Trigt, Fernando, and the younger colleagues, Alexandre, César, Marco and Victor. I am sure that this spirit will continue and grow in the future.

To my friend Franc van Trigt, whose competence, dedication and enthusiasm made the Stereo Head possible.

To all my friends at CAPS for the atmosphere, which has influenced this thesis. In particular to my colleagues and friends in office 155 : Luis Custódio, Carlos Pinto-Ferreira, Carlos Belo and João Pedro. To Pedro Lima for his friendship, numerous "cybernetic" discussions and for careful reviewing an earlier draft of this thesis.

A tutti i miei amici del Lira-Lab con cui ho avuto la fortuna di lavorare e che mi hanno aiutato con la loro amicizia. In particolare, a Paolo Questa (e alla Sampdoria!), Francesco Buemi (e alla Rivoluzione!), Giovanni (e alla focaccia col formaggio!), Francesco Panerai, Carla Capurro, Jan Nielsen, Massimo Massa, Fernando, i "Pasa" e Laura. Genova e l'Italia sono state grande esperienze che non dimenticherò mai. Grazie a tutti e arrivederci!

To **aeGIST** for the support in publishing the thesis. In particular to my colleagues and friends Horácio, Alexandre, Jorge, Patrícia, Palmira, Miguel and Paula, for the pleasure of working together for a common ideal. It has been, still is, and will always be a pleasure to have such friends. Thank you all.

Doing a thesis always implies impacience, late evening work, lost weekends and also, in this case, long absences. I owe all this to my parents who have allways been a marking example of character, care, work and dedication. They deserve my deepest gratitude.

To my grandmother, brother and sister-in-law for their support and understanding during all my absences and impacience. To my nephew João Maria, that I hope will read and enjoy this thesis, one day in the future.

Finally to Ana, for her infinite pacience, love and understanding.

Resumo

A Visão é um dos nossos sentidos mais poderosos para a percepção do espaço à nossa volta. Esta capacidade é determinante para a realização de várias tarefas tais como locomoção, manipulação, auto-localização, reconhecimento, etc. Esta tese aborda o problema da percepção visual no contexto da robótica móvel e ilustra alguns destes comportamentos guiados por visão.

Tradicionalmente, a visão era encarada como um problema geral de "reconstrução", visando a obtenção de um modelo interno do mundo exterior. Estes modelos, seriam então usados para tarefas de um nível cognitivo mais elevado, tal como planeamento e reconhecimento. A primeira parte desta tese, enquadra-se neste paradigma. É tratado o problema de reconstrução tridimensional usando uma sequência de imagens adquirida por uma câmara em movimento. Os resultados obtidos indicam que o sistema realiza uma recuperação robusta e precisa de mapas de profundidade do ambiente, que podem ser usados por um agente móvel para planeamento, reconhecimento, auto-localização, etc.

Contudo, a construção de modelos internos do mundo é uma tarefa difícil. Uma abordagem diferente propõe a utilização do próprio mundo como o modelo, e encara a visão como um modo de "extrair" a informação relevante para um dado objectivo. Esta nova abordagem tem sido designada por termos como Visão Activa, Visão Qualitativa, Visão por objectivos ou Visão Animada.

Neste enquadramento, a segunda parte desta tese aborda problemas de navegação num ambiente desconhecido, propondo um sistema autónomo de navegação inspirado na visão de alguns insectos; detecção de obstáculos baseada em técnicas de projecção inversa; e comportamentos reflexivos de atracagem (*docking*) para veículos móveis. Todos estes comportamentos usam uma parte especializada do campo visual (periférica ou central) e informação do fluxo óptico na imagem para atingir os seus objectivos. Não é feito nenhum esforço no sentido da reconstrução tridimensional do ambiente.

Para além da interpretação dos estímulos visuais, durante o nosso próprio movimento, a biologia demonstra a importância do envolvimento activo do observador no processo de percepção, nomeadamente a faculdade de movimentar os olhos para explorar o espaço circundante. A parte final da tese aborda um sistema para controlo activo da direcção do olhar, capaz de realizar os movimentos básicos do sistema oculomotor do ser humano.

Ao longo da tese, são ilustrados vários resultados destes "comportamentos perceptivos" e é discutida uma perspectiva integrada do sistema completo.

Palavras Chave : Visão por computador, Percepção Visual, Robótica móvel, Visão Activa, Stereo, Comportamentos Visuais.

Abstract

Vision is one of our most powerful senses to perceive the space around us. This ability is determinant to accomplish a variety of operations like locomotion, manipulation, self-localization, recognition, etc. This thesis addresses the problem of visual perception in the context of mobile robotics, and directly illustrates some visually-guided behaviours.

Traditionally, vision was seen as a general "recovery" problem, aiming at building an internal model of the external world. These models, could then be used for higher cognitive tasks, such as planning and recognition. The first part of this thesis, can be identified with this paradigm. It concerns the problem of 3D reconstruction using, as an input, an image sequence acquired by a moving camera. The results obtained, indicate the system ability to accurately recover the depth maps of the environment. Hence, such maps can be used by a mobile agent moving throughout the scene for planning, recognition, self-localization, etc.

However, building internal models of the world is a difficult task, and a different approach proposes the use of the world itself as the model and look at vision as a way to "extract" some features of this model relevant to a given purpose. Active, qualitative, purposive or animate vision are all terms used to designate this new approach.

Within this framework, the second part of the thesis addresses the problems of navigation through an unknown environment, proposing a system for autonomous navigation inspired on insect vision; obstacle detection based on inverse projection techniques; and reflexive docking behaviours for mobile robots. All these behaviours use a specialized part of the visual field (peripheral or central) and image flow information to accomplish their goals. There is no effort to perform 3D reconstruction.

Other than interpreting the input from the visual world, during our own motion, biology shows the importance of the observer active involvement to perceive the world, namely the ability to move the eyes to exploit the space around us. The final part of this thesis addresses a system for active gaze control, capable of the basic eye movements of the human oculomotor system.

Throughout the thesis, many results of each of these "perceptual behaviours" are presented and an integrated view of the whole system is discussed.

Keywords : Computer Vision, Visual Perception, Mobile robotics, Active Vision, Stereo, Visual Bevaviours.

Contents

Chapter 1

Introduction

The perception of the surrounding environment is determinant to accomplish a variety of operations in the domain of mobile robotics. Very often, mobile robotic applications involve some degree of interaction with the world itself : locomotion, manipulation, self-localization, obstacle detection and avoidance, tracking targets, recognition, etc.

Vision is one of our most powerful senses when perceiving the space around us. For example, the human being relies intensively on visual information in order to do a large number of tasks like driving, handling tools, walking, or playing tennis. Looking again at nature and biology, we can find a huge number of animals which use vision efficient and robustly in many activities, often determining their survival in the world.

It is also interesting to see how different animal species have evolved to different physiological solutions to vision (say, human and insect vision) and yet share some common features in the way they interpret the visual cues and interact with the environment.

Similarly, a large number of important problems in mobile robotics can benefit from the use of such powerful sensing modality. This is the scope of this thesis : the use of visual perception for mobile robots.

What is Vision ? Understanding vision has been a difficult challenge to many generations of philosophers, mathematicians, psychologists, neurolobiologists, psychophysicists, and, more recently, computer scientists and engineers.

1.1 The ecological approach to vision

One of the most captivating theories of the psychology of visual perception was proposed by Gibson [Gibson, 1950, Gibson, 1966, Gibson, 1979]. The main idea underlying this theory is that, rather than building internal models of the external world (the *percepts*), the environment is considered to be the repository of all the important information. Therefore, perception would consist in the interaction with the world in order to extract the information relevant to perform a given action. Gibson's theory used the concept of the *optic array* [Gibson, 1961] :

> *An optic array is the light converging to any position in the transparent medium of an illuminated environment insofar as it has different intensities in different directions. (...) Geometrically speaking, it is a pencil of rays converging to a point, the rays taking their origin from textured surfaces , and the point being the nodal point of an eye.*

According to Gibson, all the information needed to act in the world should be computed directly from the interaction with the external world. He called this information *affordances*. The affordances of a surface or object is what it offers to an animal - whether it can be grasped or eaten, trodden or sat upon [Gibson, 1966, Gibson, 1979, Bruce and Green, 1985]. The ambient optic array is passively sensed by a moving observer :

> *The normal human being, however, is active. His head never remains in a fixed position for any length of time except in artificial situations. If he is not walking or driving a car or looking from a train or airplane, his ordinary posture will produce some changes in his eyes in space. Such changes will modify the retinal image in a quite specific way.* ([Gibson, 1950], page 117)

Two ideas are worth stressing here : On one hand, the activity of the observer is a key issue in Gibson's theory, as the observer can watch changes in the ambient light which directly convey information about both the world and himself. On the other hand, the observer motion does not depend on the perceptual needs and, therefore, in some sense, the observer is passively sensing the optic array [Bruce and Green, 1985,

Pahlavan et al., 1993]. In this ecological approach, perception and action are seen as tightly connected and mutually constraining.

The optic array changes when an animal is moving throughout the environment. However, Gibson referred to the existence of variant and invariant properties of the array, the latter being potential stimuli [Gibson, 1961]. The idea that the information should be "picked up" by an observer, while moving in the world and performing various tasks, suggests the important idea that perception is based on selective attention mechanisms. As Gibson wrote [Gibson, 1950] :

> *The world of significant things is too complex to be attended to all at once, and our perception to it is selective.*

1.2 The representational approach to vision

The foundations of modern computer vision were established by the pioneering work of David Marr [Marr, 1982]. His understanding of visual perception arises, to some extent, as a reaction to the theories of Gibson and is motivated, on the other hand, by the goal of actually building artificial "seeing systems" with computer technology[1]. Marr defined vision as :

> *the process of discovering from images what is present in the world, and where it is.*

The main issue consisted in the "recovery" of information about the external world, in order to build internal models. According to Marr, computational vision should be understood as an information processing system. A central role was played by the internal representations, called intrinsic images : the raw primal sketch, containing information about edges segments, bars, junctions and blobs; the $2\frac{1}{2}$-D sketch to represent depth and shape of objects in viewer centered coordinates; and finally, a 3D world coordinate model of the object, to be used for recognition and navigation. Some of these representations were supported by observations in the visual cortex regarding, for instance, the

[1] In fact, Gibson's theories can be considered as the last precomputational theory [Ballard and Brown, 1992]

existence of cells with different spatial frequency sensitivities. The importance of internal representations is again clear when Marr states that :

> *The study of vision must therefore include not only the study of how to extract from images the various aspects of the world that are useful to us, but also an inquiry into the nature of the internal representations by which we capture this information and thus make it available as a basis for decisions about our thoughts and actions. This duality - the representation and the processing of information - lies at the heart of most information processing tasks and will profoundly shape our investigation on the particular problems posed by vision.*

Understanding vision would comprise the understanding of three levels in a general information-processing system : the computational theory, that formally relates the images and the desired goal of the computation; the representation and algorithmic level, regarding the implementation of the computational theory; and the final level of hardware implementation.

This approach would shape the research in computer vision for many years (see [Tarr and Black, 1994a] for a discussion). In fact, many "vision systems" were built under this recovery paradigm, creating multiple representations of the world around the observer. Examples can be found in the literature in problems like "structure-from-X", where the X can be motion, stereo, shading, texture, etc (see [Aloimonos and Shulman, 1989, Horn, 1986] for an overview). These models or intrinsic parameters, are then used for higher cognitive tasks, such as planning and recognition.

In spite of the difficulties of recovering shape, egomotion or structure from images, basically due to the ill-posed nature of such inverse problems, many systems, able to construct useful models of the world [Witkin, 1980, Grimson, 1981, Davis et al., 1983, Horn and Brooks, 1989] were built throughout the past years. In order to overcome the ill-posed nature of many of these visual processes, constraints are often introduced to find a stable solution : rigidity, smoothness, etc. Moreover, in the past few years we have watched an increase on the complexity of the tools usually used in computer vision.

1.3 Purposive and Qualitative Active Vision

However, people have recognized that building internal models of the world is a difficult task, in spite of the temptation to build a general purpose model. It often faces problems of complexity and unstability. A different approach, partially inspired in the work of Gibson proposes the use of the world itself as the model and look at vision as a way to "extract" some features of this model relevant for a given purpose.

In the earlier approaches to *Active Vision* the key idea was that the observer was actively involved in the perception process. However, this involvement basically consisted in controlling egomotion during the image acquisition process. With this procedure, some of the traditionally ill-posed problems of vision, would become well-posed [Aloimonos et al., 1988] for an active observer.

The *Active Vision* paradigm then evolved to a more general idea in which controlled perception strategies were determined by the interaction with the environment, for a given specific purpose [Bajcsy, 1985, Bajcsy, 1988, Aloimonos, 1990].

Even though the *active observer* may have important advantages when compared to the *passive observer*, the reconstruction of accurate and general representations of the world is still a hard problem. The idea of *purposive vision* advocates that perception and purpose are linked so tightly that they cannot be separated [Aloimonos, 1993]. Therefore, visual perception is grounded on attentional mechanisms in order to process only the information relevant for the current purpose. Therefore, the interaction with the surrounding environment is achieved through a set of visual behaviours, coupling perception and action together with purpose. In principle, these behaviours should not required extensive and complete modeling of the world, and should be able to operate using partial, qualitative measurements of the environment, which is the main contribution of *qualitative vision* [Aloimonos, 1990].

This approach to visual perception, also known as *animate vision* [Ballard, 1991], has often been inspired on observations in biological vision systems and particularly on the human visual system. Other than interpreting the input from the visual world, during our own motion, biology shows the importance of the observer active involvement, to perceive the world. The ability to move the eyes to exploit the space around us and fixate

interesting targets, is a key advantage from a perceptual point of view. Consequently, in most *Purposive and Qualitative Active Vision* systems, problems like real-time gaze control and reactive behaviours, play an important role.

To summarize, we can say that the *Active Vision* paradigm has evolved from the idea that the goal of action is to perceive to a new paradigm where the goal of perception is to act.

1.4 Reconstructionism versus Purposivism

There has been (and continues) a fruitful discussion about the reconstructive and purposive paradigms to computational vision. This discussion, and the opinions of various researchers on this field, are well patent in the debate proposed in [Tarr and Black, 1994a], on representations in computer vision.

One of the main points is related to the purpose of representation. Appropriate models or representations should be derived according to their specific, task-dependent, use (see, for instance [Edelman, 1994, Fischler, 1994, Sandini and Grosso, 1994] and [Jain, 1994]). On the other hand, although purposive vision has led to successful examples of visual behaviours in robotics (there are some examples in this thesis), there is still an open discussion on the coordination and integration of multiple behaviours, and on the emergence of complex behaviours based on the simpler ones [Aloimonos, 1994, Brown, 1994, Christensen and Madsen, 1994, Tsotsos, 1994] (scalability).

The main conclusion drawn is that both approaches have led to important understanding and achievements in computer vision and in understanding human vision (see [Tarr and Black, 1994b] for a balance of the debate) and that research should be pursued in both directions.

1.5 Structure of the Thesis

This thesis addresses the problem of Visual Perception for Mobile Robots. In some sense, it reflects the evolution of the various schools of thought towards computational vision and visual perception. In fact, the thesis conveys contributions both regarding the "re-

constructive" and the behaviour-based, "purposive vision" approaches, thus justifying the title : **Visual Perception for Mobile Robots : from Percepts to Behaviours**.

Chapter 1 describes the main approaches to visual perception in an abstract level, and motivates the relation of perception and action within the context of mobile robots. It also describes the structure of the thesis and outlines its main original contributions.

Chapter 2 is devoted to the problem of 3D reconstruction using, as an input, an image sequence acquired by a single moving camera. The results obtained, indicate the system ability to recover, with some accuracy, the depth maps of the environment. These maps can in turn be used by any mobile agent moving throughout the scene for planning, recognition, self-localization, etc. Mainly due to the difficulties of the reconstruction process, we use techniques like regularization and integration over multiple frames to increase the confidence on the depth estimates. There are examples using images both from underwater and land environments.

The second part of the thesis, instead, addresses various mobile robot tasks, within the framework of purposive and qualitative active vision. **Chapter 3** describes an approach to autonomous navigation in unknown environments, *Divergent Stereo*, inspired on insect vision. This system has the capability of following corridors and walls, avoiding obstacles and motion is controlled using qualitative measurements of the peripheral flow field.

The problem of obstacle avoidance is addressed in **Chapter 4**. We use inverse projection techniques on the normal flow of the central visual field of a forward pointing camera to detect obstacles lying on the ground floor, ahead of a mobile robot.

In **Chapter 5**, we propose visual behaviours for docking in mobile robotics. There are many situations in which the task of approaching an environmental point with a given orientation is important in mobile robotics. This is accomplished by these docking behaviours which use the information of the normal flow to control the robot speed and direction of heading.

All these behaviours use a specialized part of the visual field (peripheral or central) to accomplish their goals. On the other hand, they are all based on a partial description of the optical flow field as there is no attempt nor need to determine both components of the optical flow. Finally, there is no effort to perform 3D reconstruction.

In **Chapter 6**, we address another important characteristic of active vision systems :

active gaze control. We analyse the basic eye movements of the human oculomotor system and describe an active camera head designed for active vision applications. Again, the methodology to track moving targets is based on simple image measurements which are used directly by the control system.

Throughout the thesis, there are many results of each of different "perceptual behaviours" which cover a wide range of visual based functionalities of a mobile robot. Finally, in **Chapter 7** we draw some conclusions and establish further directions of research.

1.6 Original Contributions

This thesis addresses the problem of visual perception in its various facets and implications in robotics. It contributes towards a more global implementation and analysis of vision related robotic capabilities.

Chapter 2 is focused on the estimation of structure from motion. The main contribution is the analysis and characterization of the uncertainty in the matching stage and a modified matching cost functional to cope with illumination changes in the environment. These improvements were partially motivated by an underwater application of the system.

Chapter 3 proposes a new approach to mobile robot navigation. One main feature is the use of a lateral camera positioning, inspired on insect vision. Then, we use partial qualitative optical flow information which is directly used by the control system to implement behaviours like corridor following, wall following, etc. There is no need to reconstruct the environment.

Chapter 4 describes an obstacle detection mechanism which, again, uses solely partial optical flow (normal flow) information to detect obstacles on the ground floor. Other novel aspects include the absence of calibration of the camera intrinsic parameters or position. It can cope with general (translation or rotational) motion.

Chapter 5 proposes autonomous behaviours for the important task of docking in mobile robotics. The main contributions are again the use of the normal flow and the direct coupling of the perception and behaviour in the two proposed docking situations : ego-docking and eco-docking.

In Chapter 6 we describe the design and control of an anthropomorphic active camera head. The main contributions are the direct use of image data in the control loop and a two-level organization of the control system, which allows a simpler analysis of the visual feedback loop and provides insight into the overall system behaviour. As a consequence, we show good tracking results with simple hardware and control methodologies.

Chapter 2

Visual Reconstruction

According to what has been discussed in the introduction chapter, the perception of the three dimensional structure of the working space of a robot allows the completion of different tasks such as moving around the environment, manipulation, self localization, recognition, etc.

For the human being, and for many other living beings, there are several visual cues responsible for the perception of the 3D structure of the surrounding environment [Gibson, 1950, Marr, 1982]. One of the most important of these visual cues is the *stereopsis* [Marr and Poggio, 1979] which allows the recovery of the 3D structure of a visualized scene, using two images acquired from different viewpoints.

In the following sections, the problem of depth extraction based on stereo techniques, will be formalized. The different algorithms that will be described, compose a three-dimensional vision system which may be used by any moving robot, whenever it is important to obtain a depth map of the working environment.

2.1 Introduction

The application scenario of the techniques we describe throughout this chapter, considers an autonomous mobile agent equipped with a video camera. During a mission, the vision system uses the input of an image sequence acquired over time, to estimate the 3D structure of the environment.

There are two important characteristics that should be taken into account for such a system to be useful in practice :

- Recursiveness - Due to the limited on board memory space available in an autonomous mobile robot, the system should be recursive, in the sense that the computations performed at every sampling instant, should not require a large amount of past information. In our system, the depth map estimated at each instant corresponds to the current position of the robot. Alternatively one could use a world-fixed coordinate system as well.

- Uncertainty - All the processes involved in the depth estimation problem, are affected by some degree of uncertainty. Hence, recognizing the presence of uncertainty and making an effort to model the different error sources in the overall process is an important step to ensure the usefulness of the system. This work uses well known estimation techniques to deal with the uncertainty associated to the depth maps.

In the system proposed, the depth map estimation is the result of three major processes : the matching process, the regularization process and the Kalman filtering process.

The matching process consists in determining the correspondences between homologous points in images acquired at different time instants. These correspondences, or equivalently, the disparities, are determined by the use of a correlation-based method extended by a geometric constraint (the epipolar constraint or epipolar line) introduced by the camera motion.

Since the matching process is known to be an ill-posed problem [Bertero et al., 1988, Poggio et al., 1985], a regularization process was introduced in order to constrain the disparity field to a class of solutions satisfying some smoothness properties. Hence, it is possible to reduce significantly the noise associated to the disparity estimates and fill in areas in the disparity vector field, left void by the matching process.

In the final stage of the processing system, a Kalman filter integrates over time multiple depth measurements, thus reducing the uncertainty. With this approach, one can use closely spaced images (simplifying the matching process but leading to a poor precision in the depth estimates) and obtain precision levels that could only be achieved by a

long baseline system (which, on the other hand, would harden the matching process significantly).

The system has been applied to a variety of synthetic and real images, the results being shown later in this chapter. The following sections describe thoroughly each of the different processes involved in the overall depth estimation system.

2.2 Camera model

The most commonly used model to describe the image formation mechanism (from a geometric point of view) is the perspective projection (*pinhole camera model*). This model [Ballard and Brown, 1982, Horn, 1986] is illustrated in Figure 2.1[1].

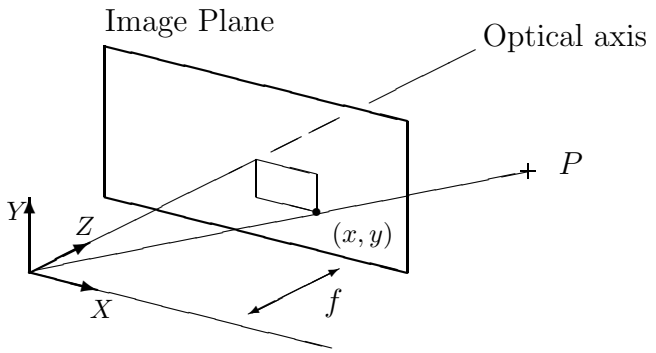

Figure 2.1: Pinhole camera model

According to this model, a point in the 3D space $[X\ Y\ Z]^T$, is projected onto the image point $(x,\ y)$ obtained by intersecting the line defined by the 3D point and the camera optic center (projection center), with the image plane.

$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z}. \tag{2.1}$$

[1]Although it may sound surprising, the laws and principles of perspective were only recently established, during the Italian Renaissance.

where f stands for the focal length and the x, y coordinates are expressed in meters. To convert these units into pixel coordinates, x_a, y_a, one must resort to the camera *intrinsic parameters* [Horn, 1986] :

$$x_a = K_x\, x + C_x \tag{2.2}$$

$$y_a = K_y\, y + C_y, \tag{2.3}$$

where K_x, K_y depend on the size of each image pixel, and C_x, C_y define the image coordinate center, the point where the optical axis intersects the image plane.

In a more compact notation, these parameters are often defined in the literature as :

$$x_a = f_x \frac{X}{Z} + C_x \tag{2.4}$$

$$y_a = f_y \frac{Y}{Z} + C_y, \tag{2.5}$$

where f_x, f_y can be interpreted as the focal length expressed in pixels (therefore, for non square pixels, we have two different values). A number of methods have been proposed on how to calibrate the camera model parameters and on the use of more complex camera models [Tsai, 1986, Lenz and Tsai, 1988, Faugeras et al., 1992]. However, we assume that these parameters are known with some accuracy for the 3D reconstruction. In the system we describe here, we also assume that the navigation system provides information on the vehicle angular and linear velocities.

2.3 Matching

When two images of a given scene are acquired from different viewpoints, the various objects present in the scene will appear in different locations on both images. The object displacement vector in the image is called the disparity. Similarly, the movement of a camera in a static environment induces a disparity vector field, in the images successively acquired. The disparity field depends on two main factors : the characteristics of the camera motion and the three-dimensional structure of the scene. Therefore, one can use disparity measurements to estimate 3D information about the world. This section is devoted to the problem of estimating the disparity, known as the matching problem.

The matching problem consists in determining correspondences between points or features of two images of the same scene, acquired from different view points. Two image points/features are correspondent or homologous if they are the projections in both images of the same 3D point/feature [Dhond and Aggarwal, 1989]. Most known methods start by considering a point/feature in one image and define some cost criterion to drive the search for an homologous point/feature in a given region of the second image. There are basically two main classes of matching methods : feature based and area based [Dhond and Aggarwal, 1989]. In the former [Marr and Poggio, 1979, Pollard et al., 1981, Grimson, 1984], the images are preprocessed to extract relevant features like edges, corners [Moravec, 1977], edge segments [Ayache and Faverjon, 1987], curves, regions, etc, which are then matched based on a set of local characteristics (like intensity, orientation, area, length, etc). In the latter the goal is to obtain correspondences for every image pixel, usually relying on some kind of correlation method [Anandan, 1989, Mathies et al., 1989, Okutomi and Kanade, 1991]. A comparative analysis of different methods can be found in [Barron et al., 1994].

The procedure used here is an area based method aiming at recovering a dense disparity field, and the matching criterion assumes the image brightness constancy hypothesis. That is to say that even though the camera motion induces a velocity field of the image brightness patterns, these brightness patterns remain unchanged over time, which is a plausible hypothesis in the absence of extreme illumination variations.

Let us consider a 3D point which is projected in two images acquired consecutively during the camera movement. Then, the gray level distributions, in a neighbourhood of the correspondent or homologous points, p'_t and $p'_{t+\tau}$, should be very similar provided that the illumination conditions did not change. Based on this intuitive idea, one of the most used gray-level matching criteria is computed by summing the squared gray-level difference between pixels within windows centered in p'_t and $p'_{t+\tau}$ as in [Mathies et al., 1989]. This criterion is known in the literature as the *sum of the squared differences* [Anandan, 1989, Heel, 1989] method :

$$SSD(u, v, x, y) = \sum_{\alpha,\beta} \phi_{(\alpha,\beta)} \left[I_{(t,\alpha,\beta)} - I_{(t+\tau,\alpha+u,\beta+v)} \right]^2 \tag{2.6}$$

where $I_{(t,x,y)}$ denotes the pixel (x, y) of the image acquired at time t, u and v are the x

and y disparity components, and $\phi_{(\alpha,\beta)}$ is a weighing function.

Computing the SSD over a range of possible disparity values (which bounds the search area), as depicted in Figure 2.2, leads to the definition of an error surface over the domain

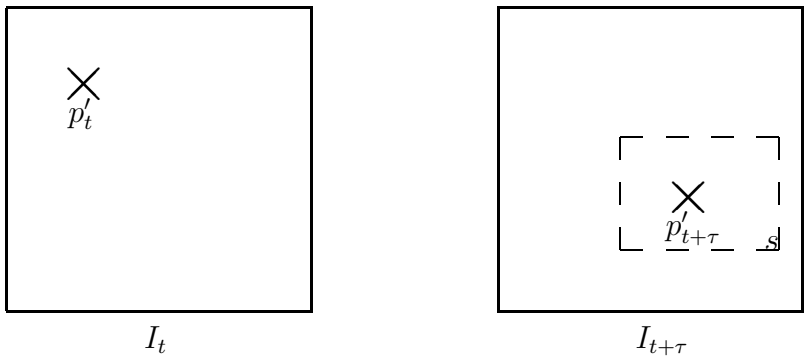

Figure 2.2: The problem of correspondence search in a stereo pair. To determine the point $p't + \tau$, correspondent to p'_τ, the area S has to be searched.

of plausible disparities. For the optimal disparity, the error surface should attain a minimal value [Anandan, 1989], which is zero for the SSD criterion, in the case of perfectly equal gray-level distributions around the correspondent points.

2.3.1 The epipolar constraint

Let the pixel p'_t be considered for the matching problem, as shown in Figure 2.3. As depth is unknown, the 3D point projected in p'_t could either be P_1, P_2 or P_3. In any case, the corresponding projection onto the right image is determined by the intersection of the right image plane, and the lines defined by the 3D point (P_1, P_2 or P_3) and the projection center, O_2. However, all these lines lie on the plane defined by p'_t, O_1 and O_2, known as the epipolar plane. Thereby, the position of the pixel $p'_{t+\tau}$ correspondent to the pixel p'_t is constrained to lie on a line (the *epipolar line*) [Horn, 1986, Faugeras, 1993] defined by the intersection of the epipolar plane and the right image plane, as shown in Figure 2.3. The epipolar line is the locus of all the homologous points of p'_t when depth ranges from zero to infinity.

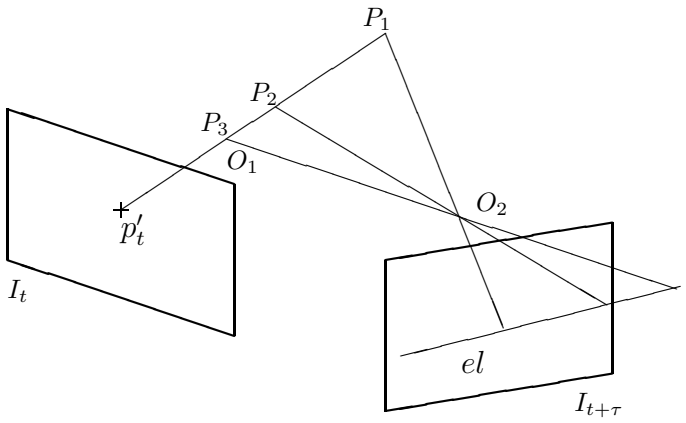

Figure 2.3: The epipolar line, el, is the locus of all the corresponding points of p'_t when depth ranges from zero to infinity.

This constraint is of the outmost importance in the matching process, as it reduces the dimension of the search space from 2 (over an image region) to 1 (along the epipolar line). The epipolar line depends on the camera motion and camera parameters, which determine the relative position of O_1 and O_2.

Therefore, the epipolar line can only be computed if the camera motion and camera intrinsic parameters are known. Alternatively, the epipolar geometry between an image pair can also be determined using a set of correspondent point matches [Longuet-Higgins, 1981, Faugeras, 1992, Hartley, 1992]. Moreover, if the camera motion is not known, the reconstruction can still be performed up to a projective transformation, which may suffice for some applications (see [Mundy and Zisserman, 1992] for a discussion). However, in our approach we assume that the motion parameters are provided by the mobile vehicle navigation system.

By using the epipolar constraint as a form of *a priori* knowledge in the matching problem, one can define [Santos-Victor and Sentieiro, 1992b] a ESSD (*Extended Sum of Squared Differences*) matching criterion that not only penalizes the gray level difference between two candidate pixels but also weighs deviations from the epipolar constraint :

$$ESSD(u,v,x,y) = \sum_{\alpha,\beta} \phi_{(\alpha,\beta)} \, [I_{(t,\alpha,\beta)} - I_{(t+\tau,\alpha+u,\beta+v)}]^2 + \lambda_{ep} \, d^2_{ep}(x,y,u,v), \qquad (2.7)$$

where $d_{ep}(x, y, u, v)$ is the distance from the matching candidate and $I_{(t+\tau,x+u,y+v)}$ to the epipolar line. The influence of the *a priori* knowledge in the final cost functional is controlled by the parameter λ_{ep}[2].

2.3.2 Matching with equalization

The use of the epipolar constraint leads to significant improvements in the global performance of the matching process. However, even a slight change over time, in the illumination conditions, may lead to an important degradation of the process, as the image brightness constancy assumption is no longer valid. Whenever this is the case, the ESSD functional will not exhibit a sharp minimum in for the true disparity, and the matching matching operation fails. Unfortunately, this is often the case in, for instance, underwater images where this system was applied [Santos-Victor and Sentieiro, 1993].

To overcome this problem, one can introduce some changes in the algorithm, to compensate for, at least, uniform changes in the illumination. This is accomplished by considering different classes of image regions. Let the image be partitioned in two sets of pixels :

$\mathcal{A}$ - A pixel p is a member of $\mathcal{A}$ if, in a given neighbourhood of p, there are significant changes in the gray level values of the image, or equivalently if there is some local texture.

$\mathcal{B}$ - A pixel p is classified as a member of $\mathcal{B}$ if, in a neighbourhood of p the gray levels are approximately constant, which amounts to say that p lies in an untextured image region.

For every pixel in $\mathcal{A}$, the matching process uses solely the varying component of the gray level distribution, which should be sufficient to identify homologous points and therefore, canceling out any constant offset of illumination that may exist. To some extent, this operation introduces a local equalization of the image patches, expressed in

[2]Since the epipolar line depends on the camera motion and camera parameters, the value of λ_{ep} should reflect the uncertainty associated to these parameters.

the following modified cost criterion :

$$ESSD_{(eq)}(u,v,x,y) = \sum_{\alpha,\beta} \phi(\alpha,\beta)[I_{(t,\alpha,\beta)} - \frac{I_t^{dc}}{I_{(t+\tau)}^{dc}} I_{(t+\tau,\alpha+u,\beta+v)}]^2 + \lambda_{ep} d_{ep}^2(x,y,u,v), \quad (2.8)$$

where I_t^{dc} is the mean gray level value within the matching window of image $I(t)$. For every other pixel in $\mathcal{B}$, where the image presents a "flat" brightness distribution, the matching process is based on the ESSD cost criterion as described in equation (2.7).

This approach succeeds in matching a much larger number of image patches, provided that there is some texture content, even in the presence of illumination changes. To determine whether a pixel is a member of $\mathcal{A}$ or $\mathcal{B}$, the gray level variance within the matching window is estimated and compared to a threshold value.

2.3.3 Computing the disparity

Once the ESSD values have been calculated for the domain of plausible disparities, a strategy must be defined to determine the optimal disparity. In the work described in [Mathies et al., 1989], the disparity vector is estimated by fitting a quadratic surface to a neighbourhood of the minimum value of the SSD, whereas in [Heel, 1989] a simpler solution consists in fitting to the SSD values, two one-dimensional parabolas in both the x and y directions. Usually, the minima found by these two strategies are different, since the two one-dimensional fit is not equivalent to a single two-dimensional fit (except when the paraboloid axes coincide with the x and y directions). The two-dimensional fit, however, needs a much larger spatial support in order to yield robust estimates and, therefore, the second order approximation to the ESSD surface, in a neighbourhood of the minimum, is no longer appropriate.

Hence, we have chosen to fit two one-dimensional parabolas in the x and y directions in a neighbourhood of the ESSD surface minimum :

$$q(u) = au^2 + bu + c, \quad (2.9)$$

where a, b and c are estimated using the data points. Once an analytical expression for the error surface is available, the minimum can be determined, analytically. This value is the optimal disparity value, according to the established criterion, estimated with sub-pixel

accuracy :

$$\hat{u}_{opt} = -\frac{b}{2\,a}, \tag{2.10}$$

provided that the coefficient a is different from 0.

The uncertainty associated to this estimate can be related to the shape of the ESSD error surface [Anandan, 1989, Heel, 1989, Mathies et al., 1989]. In [Anandan, 1989], the uncertainty is determined as a function of the curvature of the SSD surface along its main axes, whereas in [Mathies et al., 1989] the uncertainty is calculated using error propagation techniques in the SSD cost functional. Alternatively, in [Heel, 1989], the variance estimate can be expressed as :

$$\sigma_u^2 = \left(\frac{d^2 q(u_{opt})}{du^2} \right)^{-2} q(u_{opt}) \cdot \tag{2.11}$$

The first term in equation (2.11) expresses the decrease of the uncertainty with the increase of the error surface curvature, while the second term is a normalization factor which depends on the minimal value of the ESSD surface.

2.4 Regularization

Many visual reconstruction processes, aiming at recovering the three-dimensional information based on two-dimensional information, are often inverse ill-posed problems (e.g. the estimation of the disparity field between a stereo pair) [Bertero et al., 1988, Poggio et al., 1985, Szeliski, 1987, Terzopoulos, 1986b].

Due to the ill-posed nature of the matching process, the estimated disparity vector field is degraded by noise and may exhibit void areas, corresponding to matching failures. In this section, we present a regularization approach which, by introducing prior smoothness constraints in the disparity field, allows the reduction of disturbances and the filling in of the void areas.

A problem is said to be ill-posed, in the sense of Hadamard [Bertero et al., 1988] whenever either there is no solution; or the solution is not unique; or the solution does not change continuously on the data, thus being numerically unstable.

Using the regularization framework, it is possible to reformulate ill-posed problems into well-posed variational principles, by including a priori knowledge about the solu-

tion. The standard Tikhonov regularization [Tikhonov and Arsenin, 1977], uses stabilizing functionals to constrain the space of admissible solutions to smooth functions.

To determine the regularized solution $\mathcal{U}$, based on a set of data points $\mathcal{D}$, we define an error functional, $\Psi_d(\mathcal{D},\mathcal{U})$, that measures the proximity between the data and the proposed solution, and a stabilizing functional, $\Psi_p(\mathcal{U})$, that quantifies the smoothness constraints on the desired solution. The solution, $\mathcal{U}^*$, is obtained by the minimization of the following composed functional [Szeliski, 1990] :

$$\Psi(\mathcal{U},\mathcal{D}) = \lambda\Psi_d(\mathcal{U},\mathcal{D}) + \Psi_p(\mathcal{U})\cdot \tag{2.12}$$

The choice of both functionals, Ψ_p and Ψ_d, ensures that under weak conditions, the solution to the optimization problem exists [Anandan, 1989]. The stabilizing functional we have chosen for the regularization of the disparity vector field is the *thin membrane* [Anandan, 1989, Terzopoulos, 1986b] model, which represents a small deflection approximation [Szeliski, 1990] to the surface area :

$$\begin{aligned}\Psi(\mathcal{U},\mathcal{D}) &= \lambda\sum_{x,y}(\mathbf{u}-\mathbf{d})^T Q^{-1}(\mathbf{u}-\mathbf{d}) \\ &+ \int\int \text{trace}\,\{\nabla\mathbf{u}\nabla\mathbf{u}^T\}dxdy, \end{aligned}\tag{2.13}$$

$$Q = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}, \tag{2.14}$$

where $\mathbf{u}(x,y) = [u(x,y) \;\; v(x,y)]^T$ denotes the regularized disparity vector field, σ_u^2 and σ_v^2 are the variances of both x and y components of the observed disparity vectors, ∇ is the gradient operator and λ quantifies the relative weight of the fitness-to-data term in the global cost functional[3]. A point worth mentioning is that the fitness-to-data term depends on the confidence of the measurement data. If the uncertainty associated to a given disparity measurement is very large, the fitness-to-data term is automatically relaxed.

The domain of the surface $\mathbf{u}(x,y)$ is usually discretized using either the finite differences method or the finite element method [Horn, 1986], [Terzopoulos, 1986b]. Applying

[3]Whenever an observation $\mathbf{d}$ is unavailable, λ is set to zero, thus disabling the fitness-to-data term.

the finite element analysis, as proposed by Terzopoulos [Terzopoulos, 1986b] to the cost functionals to be minimized, yields :

$$\Upsilon(\mathcal{U},\mathcal{D}) = \sum_{x,y}[\lambda(\mathbf{u}-\mathbf{d})^T Q^{-1}(\mathbf{u}-\mathbf{d}) + \parallel \mathbf{u}_{(x+1,y)} - \mathbf{u}_{(x,y)} \parallel^2 + \parallel \mathbf{u}_{(x,y+1)} - \mathbf{u}_{(x,y)} \parallel^2]\cdot \quad (2.15)$$

The cost functional (2.15) is minimized using the Gauss-Seidel relaxation method, where $\mathbf{u}$ is determined iteratively for each point in the image grid as a function of the neighbouring values [Blake and Zisserman, 1987, Terzopoulos, 1986b]. The iterative mechanism is the same for u and v and can be written as :

$$u^{n+1}_{(x,y)} = \bar{u}^n_{(x,y)} + \frac{\lambda\sigma_u^{-2}}{1+\lambda\sigma_u^{-2}}[\, u^0_{(x,y)} - \bar{u}^n_{(x,y)}\,], \quad (2.16)$$

where $u^0_{(x,y)}$ is the x component of the disparity measured for pixel (x,y) and $\bar{u}_{(x,y)}$ is a local mean given by :

$$\bar{u}_{(x,y)} = \frac{u_{(x+1,y)} + u_{(x-1,y)} + u_{(x,y+1)} + u_{(x,y-1)}}{4}. \quad (2.17)$$

Figure 2.4 provides some geometric insight into the iterative process. At each iteration,

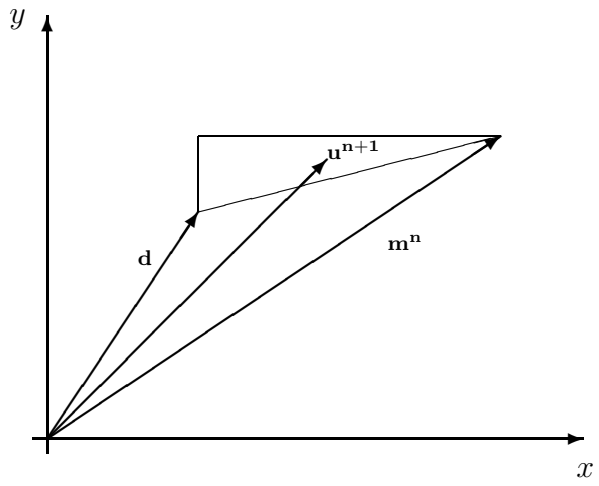

Figure 2.4: Geometric interpretation of the relaxation process.

the new disparity vector, $\mathbf{u^{n+1}}$ lies within the triangle shown in the figure, always on or above the line joining the observed disparity $\mathbf{d}$, and the local mean disparity vector, $\mathbf{m}^n$.

To determine the uncertainty associated to the regularized disparity field, we have to update the uncertainty as the regularization process evolves, imposing changes in the initial disparity field.

As the regularization iterations proceed, $\bar{u}^n$ will depend on an increasing number of data points, therefore hardening the problem of propagating the uncertainty. For the sake of computational efficiency we have used, instead, an approximate estimate, based on the relaxation equation (2.16) :

$$var[u^{n+1}] = \frac{var[\bar{u}^n] + (\lambda\sigma_u^{-2})^2 var[u^0]}{1 + (\lambda\sigma_u^{-2})^2}. \tag{2.18}$$

2.5 Coarse-to-fine control strategy

To improve the efficiency and accuracy of the results, the whole algorithm is running under a coarse-to-fine control strategy, based on a Gaussian pyramid [Anandan, 1989, Rosenfeld, 1984]. The basis of the pyramid corresponds to the finest level of resolution and contains the images acquired. The successive lower levels of resolution are obtained by low-pass filtering (with a gaussian filter) and subsampling the original images. In each level of the pyramid, the image dimension if halved (in each direction) when compared to the precedent finer resolution level.

The matching process is started at the coarsest level of resolution, where the disparity field is estimated and regularized. At the coarsest resolution, it is possible to obtain a rough estimate of the disparity field without excessive computational effort, due to the reduced size of the images and search regions used for the matching.

Once the matching/regularization processes are finished at the coarsest level, the disparity field is projected to the next (higher resolution) level. These values are then used as initial estimates of the disparity and refined by the matching algorithm applied to a small local region around the predicted value. Once again, the regularization takes place and these steps are repeated until the final estimates at the highest level of resolution are available. Figure 2.5 illustrates the method, showing the structure of the pyramid and the flow of information among the different levels.

The coarse-to-fine control strategy greatly reduces the computational effort, since the

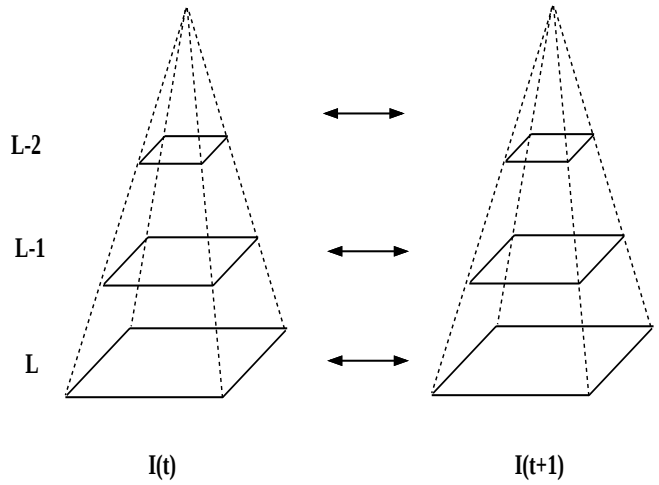

Figure 2.5: Coarse-to-fine control strategy based on a Gaussian pyramid.

most demanding computations are performed at the coarsest level of resolution. Furthermore, this strategy increases the speed of spatial propagation of the constraints involved in the regularization functional [Terzopoulos, 1986a]. The results obtained are often better than those produced by a single resolution algorithm, since the matching search space is reduced and the spatial support of the ESSD functional extended [Terzopoulos, 1986a].

2.6 Motion model

This section addresses the problem of establishing all the dynamic models to be considered in the depth estimation problem. First, we introduce the equations describing the motion of the camera with respect to a static 3D point. Then, the camera model is used to determine the velocity field induced in the image plane as a consequence of the camera motion. Finally, some considerations concerning the uncertainty sources will be made and some uncertainty models derived.

Consider a camera moving relative to a fixed point in space. Let {C} be a coordinate frame attached to the camera optic center, let $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]^T$ and $\mathbf{T} = [T_x\ T_y\ T_z]^T$ be the angular and linear velocities of the camera with respect to a fixed world frame, and let $\mathbf{P} = [X\ Y\ Z]^T$ denote the position vector of a point in the 3D space, relative to the fixed world frame. These frames are shown in Figure 2.6.

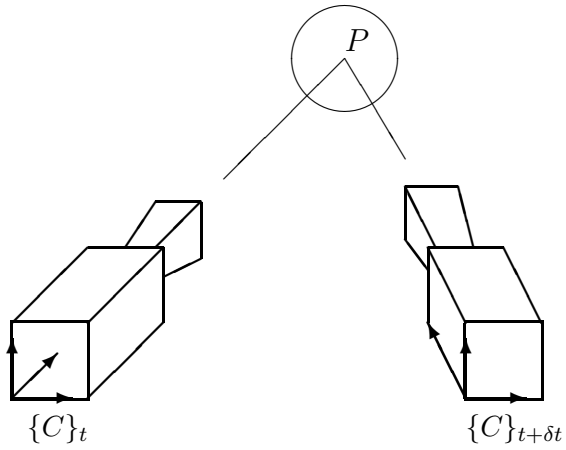

Figure 2.6: Camera in motion and the related coordinate frames.

Using the rigid body motion model, the velocity of **P** relative to {C} is described by the following differential equation [Horn, 1986] :

$$\frac{d\mathbf{P}}{dt} = -\mathbf{T} - \boldsymbol{\omega} \times \mathbf{P}. \tag{2.19}$$

To determine how this motion is projected onto the image plane, the camera model must be used. Using the pinhole camera model (described in Section 2.2) together with equation (2.19), and eliminating Z, we get a new set of equations that express the apparent motion (velocity field) induced on the image plane by the real movement of the camera [Ballard and Brown, 1982, Heel, 1989, Horn and Shunck, 1981][4] :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\mathbf{T} + \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\boldsymbol{\omega},$$

$$\dot{Z}(t) = (\omega_x y - \omega_y x)Z(t) - T_z. \tag{2.20}$$

These equations show how the induced image velocity depends on the camera motion and on the scene structure, $Z(t)$. It also shows that, in the absence of translation, the

[4]To simplify the notation, some variables were not explicitly indicated as time dependent (e.g. $x(t)$ instead of x), as it is nevertheless clear from the context.

velocity field does not depend on the scene structure and, therefore, cannot be used to recover depth.

It should be noted, however, that the motion of a mobile vehicle can only be known with limited precision and therefore the camera motion parameters cannot be known exactly. Also, the pinhole model is a simplified description of the image formation process and the camera intrinsic parameters are also not known exactly. Furthermore, the image acquisition process and the methods used to determine the disparity are again responsible for the introduction of errors that affect the disparity estimates. Characterizing the error is then a key issue in developing the system [Szeliski, 1990]. The existence of uncertainty was incorporated into the model as additive white Gaussian noise in equations (2.20), and the discrete model is obtained by approximating the time derivatives by finite differences (τ being the sampling period)[5]. We have :

State equation :

$$Z_{[t+\tau]} = a_{[t]}\, Z_{[t]} + b_{[t]} + \eta \cdot \tag{2.21}$$

Observation equation :

$$\mathbf{d}_{[t]} = \begin{bmatrix} x_{[t+\tau]} - x_{[t]} \\ y_{[t+\tau]} - y_{[t]} \end{bmatrix} = \mathbf{C}_{[t]} \frac{1}{Z_{[t]}} + \mathbf{D}_{[t]} + \boldsymbol{\mu}, \tag{2.22}$$

where η is a zero mean Gaussian random variable with variance r, $\boldsymbol{\mu}$ is a zero mean Gaussian random vector with covariance matrix Q. It is further assumed that $\{ \boldsymbol{\mu}, \eta\}$ are independent. The terms $a_{[t]}$, $b_{[t]}$, $\mathbf{C}_{[t]}$ and $\mathbf{D}_{[t]}$ depend on the motion parameters and the image point coordinates and are given by :

$$a_{[t]} = 1 + \tau(\omega_y x - \omega_x y)$$

$$b_{[t]} = -T_z \tau$$

[5]The image coordinates in the discrete model assume a camera lens with unitary focal length. To obtain the physical coordinates on the image sensor, we have to use the camera intrinsic parameters [Horn, 1986].

$$\mathbf{C}_{[t]} = \tau \begin{bmatrix} xT_z - T_x \\ \\ yT_z - T_y \end{bmatrix}$$

$$\mathbf{D}_{[t]} = \tau \begin{bmatrix} \omega_x xy - \omega_y(x^2+1) + \omega_z y \\ \\ \omega_x(1+y^2) - \omega_y xy - \omega_z x \end{bmatrix}$$

2.7 Kalman Filtering - Recursive depth estimation

Once the disparity has been estimated and a model relating depth and disparity formulated, the problem of how to estimate depth based on the disparity measurements has to be addressed. In this section we show how a Kalman filtering approach can be used for the purpose of depth estimation and for combining multiple disparity estimates over time, thus improving the reliability of the depth estimates.

Using the state space description (see Section 2.6), it is possible to define an estimation problem to determine the value of $Z_{[t]}$, based on noisy observations of the disparity, $\mathbf{d}_{[t]}$. We will consider the estimation of depth independently at each pixel whereas the spatial dependencies are embodied in the regularization stage.

This estimation problem can be conveniently dealt with using Kalman filtering techniques. Since the observation equation is non linear on the state variable $Z_{[t]}$, the discrete time Extended Kalman Filter (EKF) has to be used [Jazwinski, 1970]. Even giving rise to a suboptimal solution the EKF was chosen rather than the optimal non linear filter to reduce the complexity. The estimation process comprises a prediction phase and a filtering/updating phase. In the prediction phase, the expected values of depth and related uncertainty are estimated using exclusively past information and the dynamic model :

Prediction :

$$\hat{Z}_{(t/t-1)} = a_{[t-1]}\hat{Z}_{(t-1/t-1)} + b_{[t-1]}, \tag{2.23}$$

$$\sigma^2_{Z_{(t/t-1)}} = a^2_{[t-1]}\sigma^2_{Z_{(t-1/t-1)}} + r, \tag{2.24}$$

where $\hat{Z}_{(t/t-1)}$ is the depth value predicted at time t, based on the data available up to

time $t-1$, $\sigma^2_{Z_{(t/t-1)}}$ is the corresponding variance and r is the variance of the noise affecting the camera motion equation.

At time t, when a new disparity observation is available, the predicted depth value can be updated. This is the core of the EKF filtering step which uses a linearized version of the observation equation around the predicted value :

$$\mathbf{d}_{[t]} \approx \mathbf{d}^L_{[t]} = \boldsymbol{\mathcal{C}}_{[t]} Z_{[t]} + \boldsymbol{\mathcal{D}}_{[t]} + \mu, \tag{2.25}$$

where $\boldsymbol{\mathcal{C}}_{[t]}$, $\boldsymbol{\mathcal{D}}_{[t]}$ are the coefficients of the linearized model given by :

$$\begin{aligned} \boldsymbol{\mathcal{C}}_{[t]} &= -\frac{\mathbf{C}_{[t]}}{\hat{Z}^2_{(t/t-1)}} \\ \boldsymbol{\mathcal{D}}_{[t]} &= 2\frac{\mathbf{C}_{[t]}}{\hat{Z}_{(t/t-1)}} + \mathbf{D}_{[t]} \end{aligned}$$

The filtering equations are given by :

Filtering :

$$\mathbf{K}_t = \sigma^2_{Z_{(t/t-1)}} \boldsymbol{\mathcal{C}}^T_{[t]} \left[\boldsymbol{\mathcal{C}}_{[t]} \sigma^2_{Z_{(t/t-1)}} \boldsymbol{\mathcal{C}}^T_{[t]} + \mathbf{Q}_t \right]^{-1}, \tag{2.26}$$

$$\sigma^2_{Z_{(t/t)}} = \left(1 - \mathbf{K}_t \boldsymbol{\mathcal{C}}_{[t]} \right) \sigma^2_{Z_{(t/t-1)}}, \tag{2.27}$$

$$\hat{Z}_{(t/t)} = \hat{Z}_{(t/t-1)} + \mathbf{K}_t \left(\mathbf{d}_{[t]} - \boldsymbol{\mathcal{C}}_{[t]} \hat{Z}_{(t/t-1)} - \boldsymbol{\mathcal{D}}_{[t]} \right), \tag{2.28}$$

where $\hat{Z}_{(0)}$ and $\sigma^2_{Z_{(0)}}$ are the initial depth estimate and related uncertainty and $\mathbf{K}_t$ is the Kalman gain.

Warping the Depth Map

To complete the analysis of the EKF process, there is still an additional problem to solve in the prediction phase.

The predicted depth value, $\hat{Z}_{(t/t-1)}$ is obtained using the camera motion model. However, this predicted value does no longer correspond to the original pixel location (x, y), as the coordinates have changed to a new location, (x', y'). As, in general, this new position does not correspond to any point in the image grid, the depth at a pixel (x, y) must

be inferred based on a set of depth predictions in points, (x'_i, y'_i), outside of the image grid. This problem can be addressed using various interpolation mechanisms such as bi-linear or bi-cubic interpolation [Heel, 1989, Mathies et al., 1989]. This idea is depicted in Figure 2.7

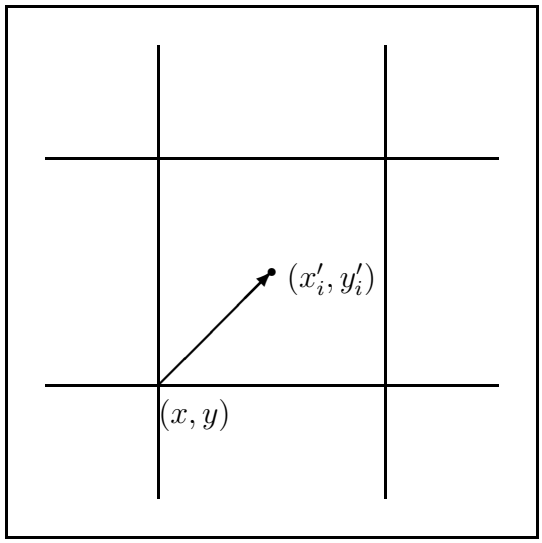

Figure 2.7: Warping the depth map. During the prediction phase the depth map has to be interpolated.

The depth estimate in (x, y) is determined as a weighed sum of the estimates laying within a 3×3 window centered in (x, y). The uncertainty of the warped map is estimated using error propagation techniques :

$$Z_{(x,y)} = \frac{\sum_{i=1}^{n} d_i^{-2} Z_{(x'_i,y'_i)}}{\sum_{i=1}^{n} d_i^{-2}}$$

$$\sigma^2_{Z(x,y)} = \frac{\sum_{i=1}^{n} d_i^{-4} \sigma^2_{Z(x'_i,y'_i)}}{(\sum_{i=1}^{n} d_i^{-2})^2} \tag{2.29}$$

where $Z_{(x'_i,y'_i)}$ represents the predicted depth values and d_i is the Euclidean distance from (x'_i, y'_i) to (x, y).

2.8 System description

The global operation of the recursive depth estimation system is shown in Figure 2.8. A new image is acquired at every sampling instant and a new depth map is estimated.

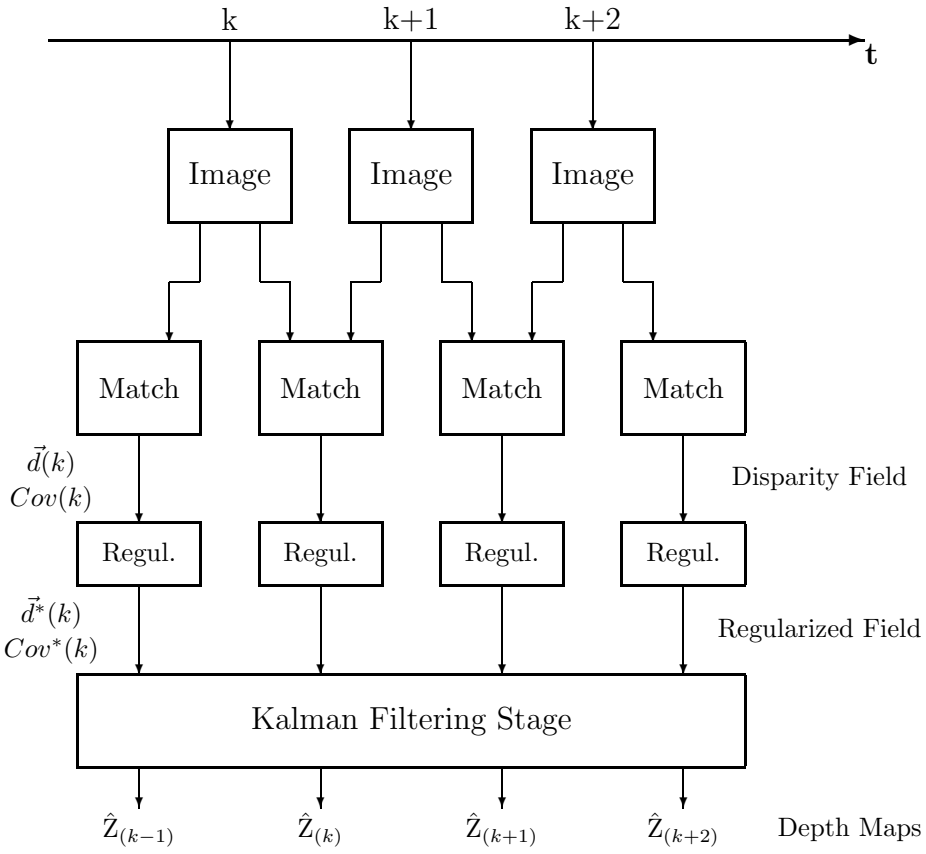

Figure 2.8: Block diagram of the 3D vision system.

The matching process is applied to each pair of successive images to determine the disparity vector field and the associated uncertainty. The disparity vector field is then regularized to reduce the uncertainty level and fill in image areas where the matcher has failed to determine the disparity. Each new observation (regularized disparities and uncertainties) are finally used to update the depth estimates resulting from previous measurements, by a Kalman filtering stage which leads to a decrease of the uncertainty over time, as more information is being gathered and taken into account.

2.9 Results

The 3D vision system presented in the previous sections has been tested for a wide variety of synthetic and real images. The initial tests have allowed improvements in the matching and regularization stages and suggested directions to overcome the different problems found. In this section, we present several results, covering applications for both underwater and land robotics.

2.9.1 Underwater application

The vision system was partially developed during the MOBIUS project of the European Community MAST (Marine Science and Technology) Programme. The overall goal consisted in estimating the bathymetry of the seabed during a mission of an autonomous underwater vehicle (AUV) or a remotely operated vehicle (ROV). Marine robotics is becoming a research field of major interest for applications such as environmental surveillance, cable laying or inspection, exploration, etc. Within this scope, the Mobius project aimed at developing a sensor combining an acoustical channel (sonar) and optic channel (vision) for high resolution mapping of the seabed. The results described in this section correspond to the 3D maps recovered by the depth from motion vision system [Santos-Victor and Sentieiro, 1992a, Santos-Victor and Sentieiro, 1992b].

The underwater image acquisition was done in a special test tank with a specially designed camera for underwater applications. The camera main characteristics are the high sensitivity and spectral response of the CCD sensor, particularly suited for underwater imagery without any external illumination [Santos-Victor and Sentieiro, 1993]. The camera was fixed to a special mount and displaced vertically inside the tank, while some objects were placed on the opposite side of the test tank. The experimental setup is shown in Figure 2.9.

The underwater environment is quite challenging for vision applications due to the extreme and difficult illumination conditions. Quite often, in fact, there are illumination changes which harden the matching process and consequently the problem of depth recovery. This is a good example of how the use of equalization techniques is of paramount importance in the matching process.

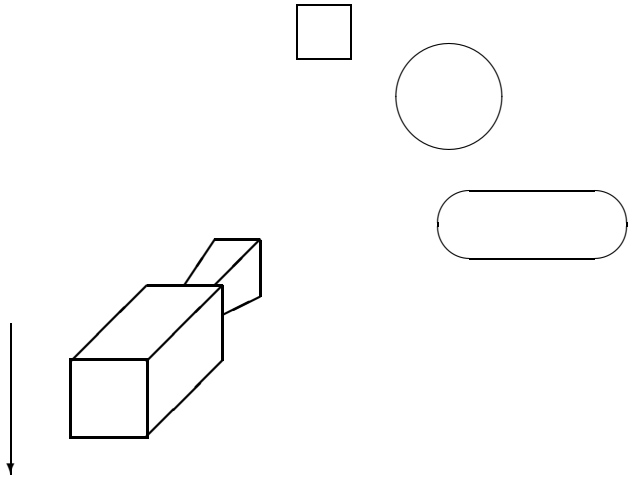

Figure 2.9: Experimental setup for the underwater application. The camera motion is vertical downwards, and obstacles were placed in front.

Figure 2.10 shows on the left column, an image pair acquired during the experiments. It is seen that the image contrast is very poor and that there are important illumination changes between both images. Applying the matching process without the equalization technique leads to the depth map shown on the top right image of Figure 2.10. Depth is coded in gray level intensity, the darker points being closer to the observer. The white areas correspond to matching failures. This result shows that without any equalization (the top right image in Figure 2.10) only small areas within the image are successfully matched. This is due to the lack of texture and to illumination changes which violate the image brightness constancy hypothesis. On the other hand, the depth map obtained using equalization is shown on the lower right image of Figure 2.10, where a large number of image points have been correctly matched. Therefore, the equalization techniques described in Section 2.3 proved to be essential to get some good results [Santos-Victor and Sentieiro, 1993].

During the experiments in the test tank, the camera was moved vertically with a downwards speed of 1 m/s, and the images are acquired every 0.4 seconds, yielding a separation of 40 cm between successive images. Regarding the matching process, a 7×7 matching window was used at the coarsest level. The equalization process is used whenever the ratio between the gray level standard variation and the average gray level value, within

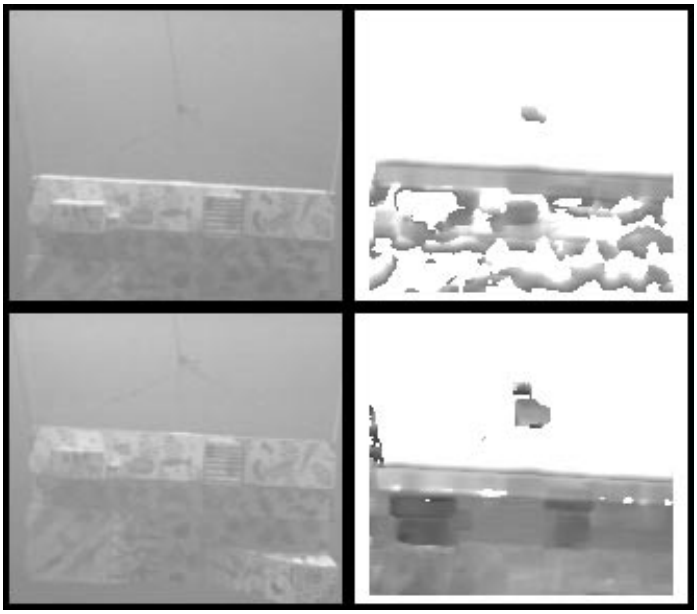

Figure 2.10: Left : Underwater stereo image pair. Right : The top image shows the depth map recovered using the equalization procedure, while the lower image corresponds to the results using equalization.

the matching window, is greater than 0.01 (see Section 2.3.2).

In all the examples tested, a three-level pyramidal structure was used for the matching and regularization procedures. The *a priori* depth map estimate is 12 meters for every image pixel, with 0.001 m^2 variance. The regularization parameter was set at $\lambda = 1$, and the epipolar constraint weight, $\lambda_{ep} = 5$.

Figure 2.11 shows results for one of the image sequences. The left column shows the image sequence acquired during the camera motion, while the right column shows the corresponding depth maps in perspective. It should be noticed the low contrast of the input images and the brightness variation along the image sequence. Nevertheless, the system has succeeded in estimating the depth structure of the scene. These results show that the system is able to reconstruct the 3D shape of the scene. It is also noticeable the improvement in the depth maps as time goes by. This is due to the integration of multiple estimates by the Kalman filtering procedure. Very often, during the experiments, the camera motion is affected by perturbations, which did not prevent the system from retrieving the scene three dimensional structure.

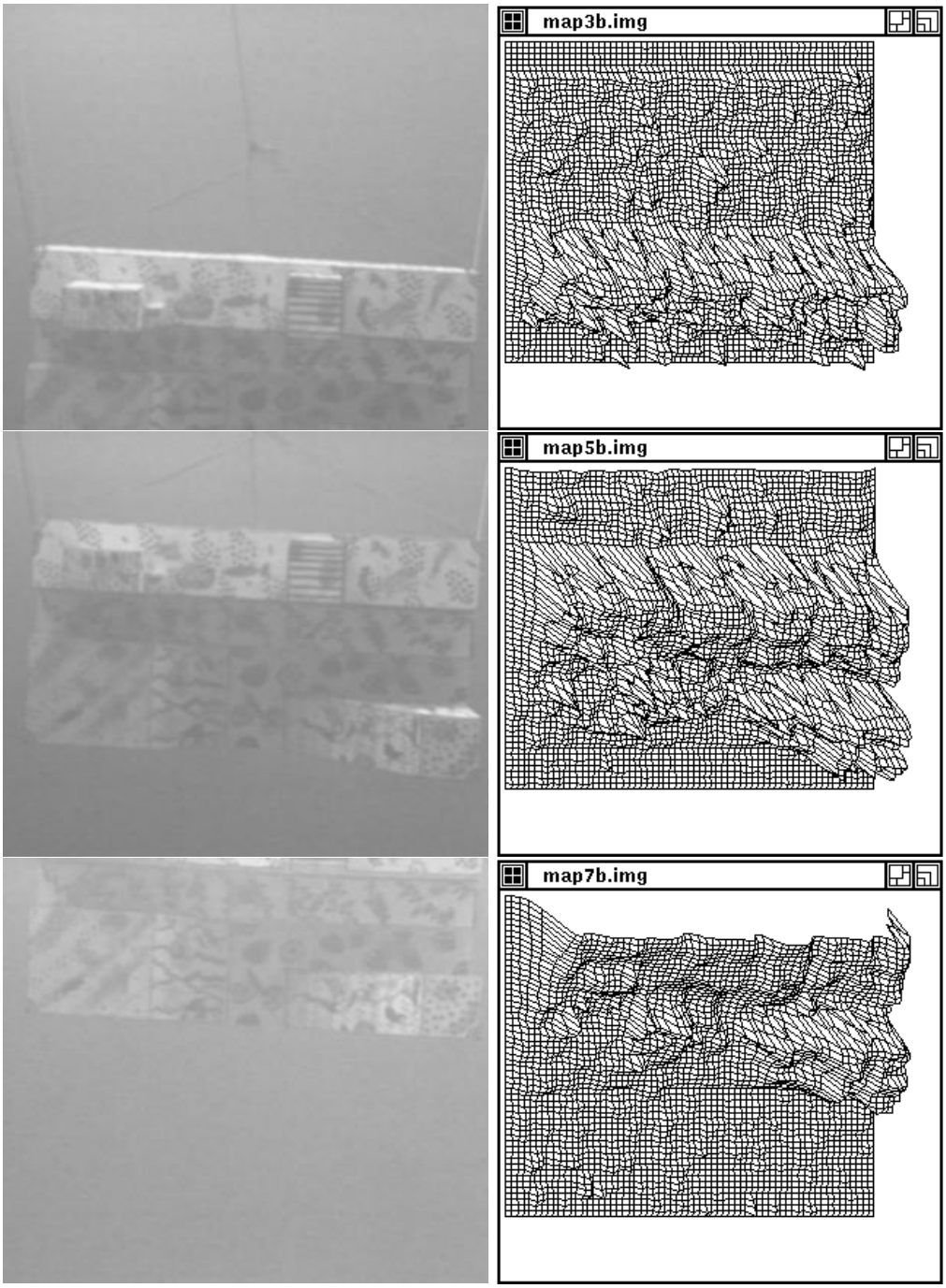

Figure 2.11: The left column shows the input image sequence, acquired during the camera motion in the experimental pool. On the left side, the perspective view of the reconstructed depth maps are shown.

2.9.2 Land Robotics Application

The same vision system was tested with terrestrial images. In these circumstances, the contrast is usually good when compared to the underwater images. However, the adaptive equalization mechanism may still be useful in the presence of illumination changes (which often happen in outdoor environments).

The input images used in one trial[6], are shown on Figure 2.12. The camera motion is divided in two segments. During the first half, the motion is horizontal towards the right. The images corresponding to this part are arranged horizontally, thus suggesting the camera motion. After a while, the camera starts moving vertically upwards, acquiring the images shown in vertical arrangement. The complete image sequence comprises 12 images, 5 of which are shown in Figure 2.12.

The results obtained using this input sequence are shown on Figure 2.13. The representation consists in a perspective view with the gray level textured mapped on top of the 3D surface. With this kind of visualization it is easier to evaluate the quality of the reconstruction. Again, the geometrical arrangement of the maps, suggests the camera motion and corresponds to the input images shown in Figure 2.12.

The results shown illustrate the quality of the 3D reconstruction. The first estimated depth map is still very distorted, while the accuracy is improved during the sequence and time integration of more information. An important observation is worth mentioning : during the first half of the sequence, the reconstruction of horizontal edges is difficult, when compared to vertical edges. The reason why is that those edges are aligned with the motion direction, thus hardening the matching process. When the camera starts the vertical motion, then the reconstruction of horizontal edges becomes much easier, while problems arise with vertical edges.

This idea stresses that, ideally, for the purpose of visual perception, the camera motion should not be defined independently of the perceptual processes themselves. Instead, much could be gained by linking action/motion and perception where motion would be controlled to optimize, in some way, [Maver and Bajcsy, 1993] the quality of the perceived data. This topic will be central in some of the following chapters of this thesis within the

[6]The images shown were kindly provided by the Robotics Institute of the Carneggie Mellon University.

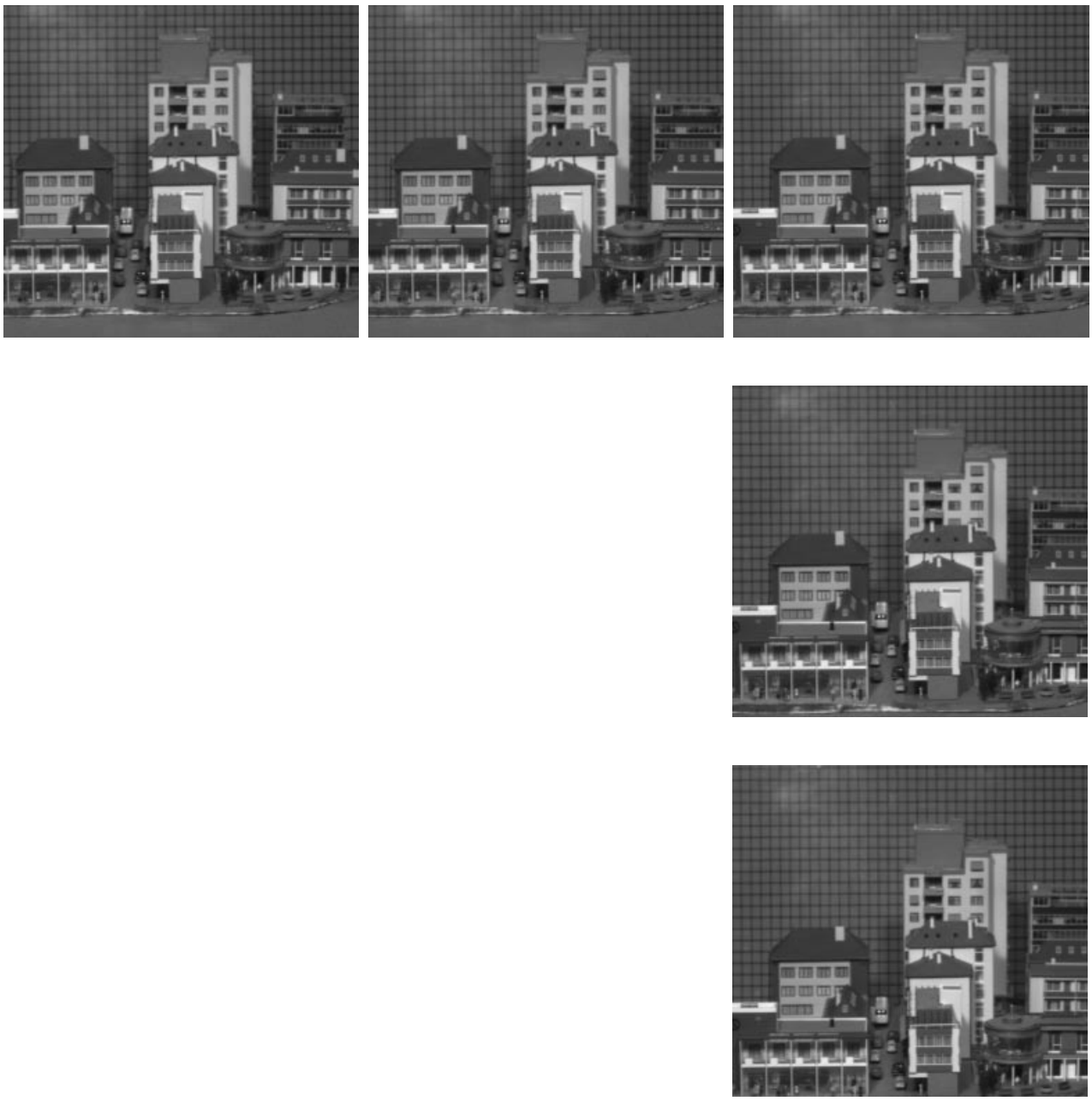

Figure 2.12: Image sequence for the land robotics example. Initially (images shown horizontally), the camera moves along the x axis, to the right. Afterwards the camera performs a upwards vertical motion (images shown vertically).

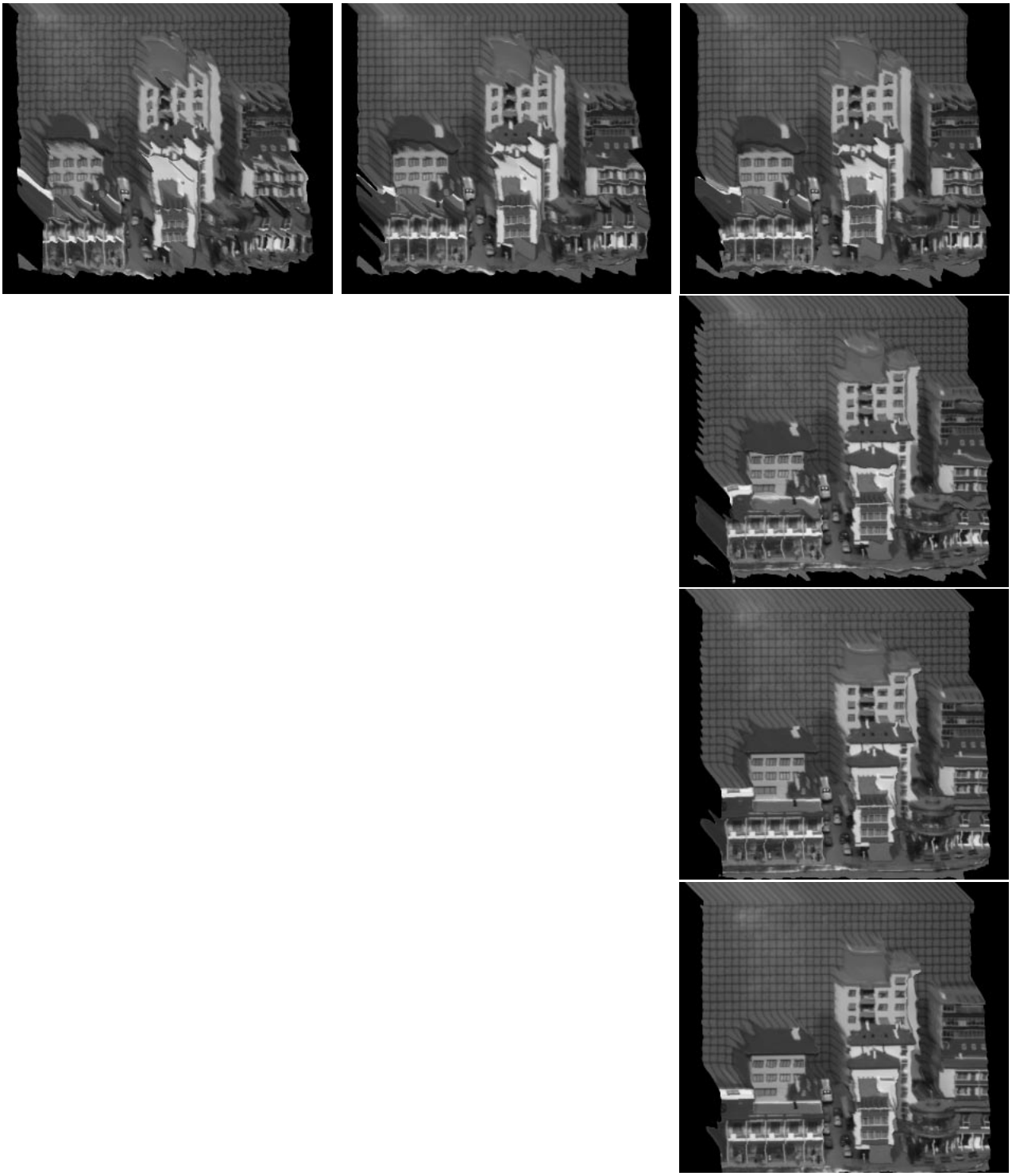

Figure 2.13: Reconstructed depth maps using the land images. The maps are shown in correspondence with the images of Figure 2.12.

scope of active vision.

2.10 Conclusions

We have discussed that a mobile actor in order to perform numerous tasks has to be able to perceive the structure of the surrounding space. Even though this may not imply the need for a thorough reconstruction of the 3D structure of the environment, such a map could be used for a variety of tasks ranging from navigation and manipulation to the higher level tasks of planning and recognition. Naturally, the 3D depth reconstruction may be a purpose in itself if, for example, we want to check the 3D characteristics of a given part for quality control.

We have presented, in this chapter, a system for 3D visual reconstruction based on stereo techniques. The system input is an image sequence acquired over time by a moving camera. In the context of mobile vehicles the camera should be installed on the vehicle during a given mission. Meanwhile, the images are used to build and continuously update, over time, a three dimensional map of the visualized scene.

There are three main features on this system. First, the existence of uncertainty is considered, from the beginning, and incorporated in the models. We have modeled and estimated the uncertainty associated to the disparity measurements. Second, we have used a regularization approach to reduce the uncertainty in the disparity vector field, and fill in image areas where the matcher has failed to estimate the disparity. Finally, we use a kalman filter to integrate, over time, multiple disparity measurements, which improves the depth estimates accuracy.

The matching procedure is the most delicate process in the overall system, therefore justifying the attention paid to this problem. We use an area based matching criterion and, as mentioned before, we address not only the problem of determining the disparity vectors but also the process of modeling and estimating the measurement uncertainty. The matching criterion is a combination of an image gray-level difference term and the geometric constraints imposed by the camera motion (epipolar line). This criterion has been modified in order to compensate for illumination changes in the images, which often occur, particularly in underwater images.

The results herein presented, both in underwater and land environments, show the 3D reconstruction capabilities of the system. The dramatic improvement due to the illumination compensation in the matching criterion was shown, namely for the underwater images. Moreover, the results show the improvement, over time, of the reconstructed depth maps due to the time integration procedure.

It should be stressed that most of the computations required in the depth estimation problem are mostly local and therefore, much could be gained in terms of computing time, by using massively parallel computation.

Even though we believe that driving a mobile robot throughout an unknown environment may not need a full 3D reconstruction (as it will be shown in the remaining chapters of this thesis), the ability of building such a map, even at a low frequency, can nevertheless be useful for high level planning or recognition or when the map itself is the goal (as in seabed bathymetry or photogrammetry). Therefore, whenever there is the purpose of estimating the scene structure, this system can be applied to a large number of problems, both in underwater and land robotics.

Chapter 3

Visual based navigation

We have seen how to estimate the 3D structure of a given scene, using an image sequence as the input. This knowledge can be used to plan a safe trajectory between two points, grasp an object or for recognition purposes. However, since the environment is usually dynamic, we have to consider a navigation strategy allowing the robot to react, in due time, to environmental changes.

In this chapter we present a new approach for a real-time navigation system, which is driven by two cameras pointing laterally to the navigation direction (*Divergent Stereo*). The main assumption is that, for navigation purposes, the driving information is not distance (as obtained by a stereo setup) but motion or, more precisely, the qualitative optical flow information computed over non-overlapping areas of the visual field of the two cameras. This qualitative information (no explicit measure of depth is performed) is used in many experiments to show the robustness of the approach.

A mobile vehicle has been equipped with a pair of cameras looking laterally (much like honeybees) and a controller, based on fast real-time computation of optical flow, has been implemented. The control of the mobile robot (*Robee*) is driven by the difference between the apparent image velocity of the left and the right cameras. The proposed navigation strategy is inspired on recent studies describing the behaviour of freely flying honeybees and the mechanisms they use to perceive range.

3.1 Introduction

Research in computer vision has often been motivated by similarities with solutions adopted in natural living systems. Since the early work in image processing and computer vision, the structure of the human visual system has often been used, as the "natural" model of artificial visual systems. Comparatively less effort has been devoted to the study and implementation of artificial vision systems based on the so-called "lower level" animals. The posture stems mainly from the idea that by "understanding human vision" one can obtain a "general" solution to the visual process, and eventually be able to develop a "general purpose visual system".[1] We certainly agree that there is a lot to be learned from human vision (and conversely there is a lot to be learned by trying to implement artificial systems with anthropomorphic features). However, such a "general purpose vision system" does not exist, and we are convinced that a lot can be gained and understood by looking at much simpler biological systems.

In fact, a more careful analysis reveals that the human visual system is not as general purpose as it may look. For example, it is of little use in underwater environments, it has a limited spectral band, it cannot measure distance, size or velocity in metric terms. Even its recognition capabilities can be fooled very easily, for example by turning upside-down pictures of even familiar faces. The apparent generality of the system comes from the fact that we actually perform a limited number of motor and cognitive "actions" and, within this limited domain, our visual system (or more generally the integration of our perceptual systems) performs very efficiently.

Following these ideas, one could say that the goal of a vision system in a "living" agent is not generality but specificity : the physical structure and the purpose drive perception [Aloimonos, 1990, Bajcsy, 1985, Ballard et al., 1989].

Within the scope of this work we would like to argue that the frontal position of the eyes (with a very large binocular field) is mainly motivated by manipulation requirements and, if one restricts the purpose to navigation control, a potentially more effective eye-positioning is the one adopted by flies, bees and other flying insects, namely with the eyes pointing laterally. The tradeoff between these two extreme situations is that in the

[1] And, possibly, the notion that "human vision is complex while insect vision is simple".

latter case the global visual field (i.e. the union of the visual fields of the two eyes) can be enlarged *without increasing the computational power* [2]. On the other hand, by increasing the binocular part of the visual field, the region where a stereo-based depth measure is possible, becomes larger.

Looking again at biology, one finds out that the position of the eyes in different species becomes more frontal as the manipulative abilities increase (it is not by chance that humans and primates have, among all species, the wider binocular field and the more frontal positioning of the eyes).

Other aspects worth considering are the fact that stereo acuity is maximal at short distances which, behaviourally, correspond to the manipulative workspace. Moreover, stereo is the only visual modality providing depth information in static environments (which is behaviourally relevant particularly in manipulative tasks). Conversely, motion parallax is useful if the "actor" or the environment are dynamic and its accuracy (and the corresponding range of operation) can be tuned by an active observer, by varying its own velocity. In this respect motion-derived features, such as time-to-crash, seem more relevant to dynamically control posture and other navigation parameters : if one has to jump over an obstacle, the change in posture while he/she is approaching the obstacle (for example when to start to lift the leg), may be driven by "time-to-crash" more than by distance (which would be dependent upon the approaching speed).

The biological model of the navigating actor proposed here, is inspired on insects and on the use they make of flow information to solve apparently complex motor tasks like flying in unconstrained environments and landing on surfaces [Franceschini et al., 1991, Horridge, 1987, Lehrer et al., 1988, Srinivasan, 1992]. Particularly relevant to this work is the experiment reported in [Srinivasan et al., 1991] where honeybees were trained to navigate along corridors in order to reach a source of food. The behavioural observation is the fact that, even if the corridor was wide enough to allow for "irregular" trajectories, bees were actually flying in the middle of the corridor. This finding is even more surprising if we take into consideration that insects do not have accommodation and, that the stereo baseline is so small that disparity cannot be reliably measured. Apparently, then, no

[2] Of course one could use "technological tricks" like rear-viewing mirrors, but this is not a valid argument in this context.

depth information can be derived using "traditional" methods. The solution presented in [Srinivasan et al., 1991], is rather simple and it is based on the computation of the difference between the velocity information computed from the left and the right eyes : if the bee is in the middle of the corridor the two velocities are the same, if the bee is closer to one wall, the velocity of the ipsilateral eye is larger. A simple control mechanism (the so-called *centering effect*) may, therefore, be based on motor actions that tend to minimize this difference : move to the left if the velocity measured by the right eye is larger than that measured by the left (and vice versa).

Following this line of thought, a mobile robot (*Robee*) has been equipped with laterally looking cameras and a controller has been implemented, based upon motion-derived measures, which does not rely on precise geometric information but takes full advantage of the continuous (in time) nature of visual information. Navigation is controlled on the basis of the optical flow field computed over windows of images acquired by a pair of cameras pointing (in analogy with the position of the eyes in the honeybees and other insects) laterally and with non overlapping visual field [Santos-Victor et al., 1993, Sandini et al., 1993b]. We called this camera placement *Divergent Stereo*.

While working on the experiments presented here, we became aware of a similar implementation (the *beebot* system) proposed in [Coombs and Roberts, 1992]. A discussion of the differences between the two approaches will be presented later on. However, the main difference lies on the fact that, while in the system proposed by Coombs and Roberts gaze control is part of the navigation strategy, in the present implementation a simpler setup has been analysed where gaze control becomes unnecessary by appropriately positioning the two cameras and by controlling some behavioural variables such as the "turning speed".

The experiments reported describe the behaviour of *Robee* in tasks like : following a wall, avoiding obstacles, making sharp and smooth turns, and navigating along a funneled corridor with a very narrow exit.

In Section 3.2, we describe the *Divergent Stereo* approach for robot navigation and present the experimental setup. The computation of the optical flow is discussed in Section 3.3. The control aspects will be addressed in Section 3.4. Finally, after the presentation of the experimental results (in Section 3.5), a brief discussion will summarize

the major points of this approach within the framework of qualitative vision.

3.2 The Divergent Stereo Approach

The basis of the visually guided behaviour of *Robee* is the *centering reflex*, described in [Srinivasan et al., 1991] to explain the behaviour of honeybees flying within two parallel "walls". The qualitative visual measure used is the difference between the image velocities computed over a lateral portion of the left and the right visual fields.

Even if the principle of operation is very simple a few points are worth discussing before entering into the implementation details, in order to explain the underlying difficulties and the design principles adopted. The first, and possibly major, driving hypothesis is the use of qualitative depth measurements : no attempt is made to actually compute depth in metric terms. The second guideline is simplicity : whenever possible, the tradeoff between accuracy and simplicity has been biased towards the latter criterion. Finally, the goal of our visuo-motor controller is limited to the "reflexive" level of a navigation architecture acting at short-range. In spite of these limitations, we will demonstrate a variety of navigation capabilities which are not restricted to obstacle avoidance or to the "centering reflex".

Among the practical problems encountered in the actual implementation, two are worth discussing in order to better appreciate some of the concepts presented later on. The first point regards the relationship between heading direction and optical flow computation. Strictly speaking, flow information can be reliably used for the centering reflex only when the directions of the two cameras are symmetric with respect to the heading direction. For the camera placement of *Robee* (see Figure 3.1) this requirement is satisfied only during translational motion of the robot. During rotational motions (as during obstacle avoidance), the flow field is not solely dependent on the scene structure. This problem has been solved by Coombs and Roberts [Coombs and Roberts, 1992] by stabilizing the cameras against robot rotations and by introducing a control loop keeping the gaze aligned with the heading direction. The solution adopted in *Robee* is different, as the two cameras are fixed. Section 3.3.1 presents a detailed analysis of the rotational field and discusses ways to overcome this problem, either by using special control strategies,

or by carefully selecting the camera positions.

A second, relevant, point is the unilateral or bilateral absence of flow information, caused by the absence of texture, or by localized changes in environmental structure (e.g. an open door along a corridor)[3]. If the centering reflex is applied in a crude mode, the absence of flow information would produce a rather unsteady behaviour of the robot. This problem has been solved by introducing a *sustaining* mechanism to stabilize, in time, the unilateral flow amplitude in case of lack of information. This simple qualitative mechanism does not alter the reflexive behaviour of *Robee* (in the sense that it is neither based on prior knowledge of the environment, nor on metric information) and extends the performance of the system to a much wider range of environmental situations (see Section 3.4.3 for more details).

The experimental setup is based on a computer controlled mobile platform, *TRC Labmate*, with two cameras pointing laterally. The two cameras, with 4.8mm auto-iris lenses, are connected to a VDS 7001 Eidobrain image processing workstation. The left and right images are acquired simultaneously, during the vehicle motion. The setup is illustrated in Figure 3.1. During the experiments, the vehicle forward speed is approximately 80 mm/s.

3.3 Optical Flow Computation

To compare the image velocity observed by the left and right cameras, the average of the optical flow on each side is computed. The optical flow is usually defined as the apparent motion of the image brightness patterns observed when a camera is moving relative to various objects [Horn and Shunck, 1981, Horn, 1986], and many authors have studied its main characteristics [Koenderink and van Doorn, 1975, Koenderink, 1986, Verri et al., 1989].

The major assumption used to compute the optical flow from an image sequence is the image brightness constancy hypothesis. Let $I(x, y, t)$ be the gray level value at time t at the image point (x, y). Then, if $u(x, y)$ and $v(x, y)$ are the x and y components of the optical flow vector at that point, the image brightness constancy hypothesis can be

[3]This situation occurs, even with textured environments, if the relationship between "wall(s)" distance and vehicle speed is such that the resolution of the optical flow computation is lower than image velocity (e.g. if the robot is moving slowly and the "walls" are very far away).

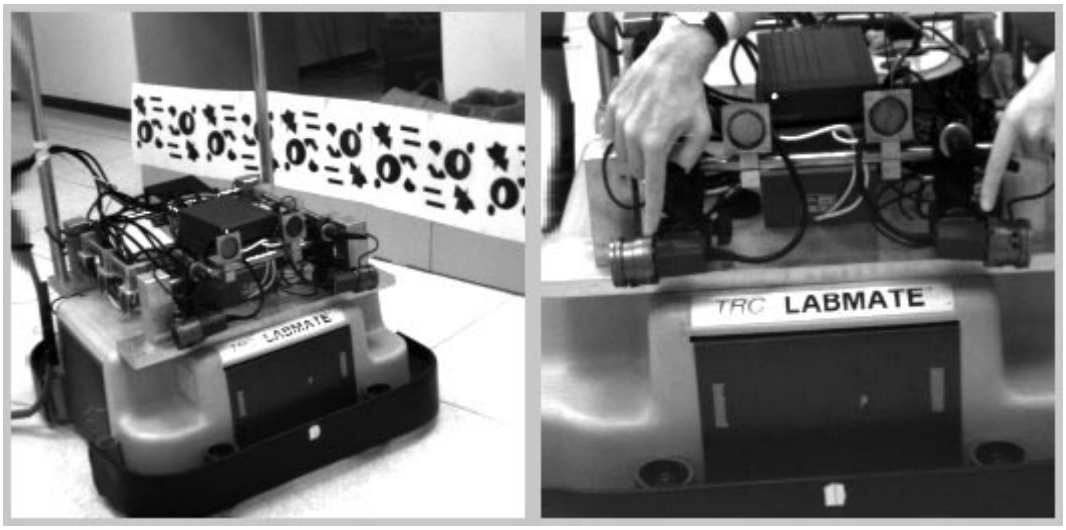

Figure 3.1: *Robee* with the divergent stereo camera setup

expressed as :

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t) \tag{3.1}$$

for a small time interval δt. If we expand the left-hand side of this equation in a Taylor series, and let δt tend to zero, we obtain :

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{3.2}$$

This equation is known as the *optical flow constraint* [Horn and Shunck, 1981] and is actually just an expansion of the equation

$$\frac{dI}{dt} = 0 \tag{3.3}$$

in the total derivative of $I(x, y, t)$ with respect to time. The constraint equation (3.2) can be rewritten in the form :

$$[I_x \;\; I_y].[u \;\; v]^T = -I_t \tag{3.4}$$

where I_x, I_y and I_t stand for the first order spatial and time derivatives of the image.

This equation shows that using local image measurements alone, we can only determine the component of the optical flow, $U_\perp$, in the direction of the image brightness gradient :

$$U_\perp = \frac{I_t}{\sqrt{I_x^2 + I_y^2}} \tag{3.5}$$

which is called *normal flow*. However, nothing can be said about the optical flow component along the direction of the image contour. This ambiguity is widely known as the *aperture problem* [Horn and Shunck, 1981, Horn, 1986].

This structural limitation, has motivated the search for alternative methods and further constraints on the optical flow allowing the recovery of both of its components [Horn and Shunck, 1981], [Nagel, 1983], [Nagel and Enkelmann, 1986], [Nagel, 1987] or in [Girosi et al., 1989, Little and Verri, 1989]. These methods are often extremely complex and/or unstable[4].

More recently, however, it has been shown that the normal flow conveys sufficient information to accomplish many tasks related to visual perception [Fermüller, 1993b, Huang and Aloimonos, 1991]. This is the approach followed in this thesis for different visually guided robotic tasks (see Chapter 3 to Chapter 5). Therefore, we avoid imposing extra constraints on the optical flow field.

The experiments used to demonstrate the *Divergent Stereo* navigation concept are based upon a mobile platform moving on a flat floor. As the robot motion is constrained to the ground plane, it can be assumed that the flow along the vertical direction is negligible (unless there is a significant lateral motion, which is seldom the case). Hence, we can use a simpler computation procedure, which is fast and robust (since, for example, it does not involve the computation of second derivatives), by simply assuming in equation (3.2) that the vertical flow component, v, is 0. The horizontal component of the flow vectors, u, are then simply given by :

$$u = -\frac{I_t}{I_x} \, . \tag{3.6}$$

In order to obtain useful flow estimates, it is necessary, as in other similar approaches, to smooth the images, in both space and time domains. Usually, this is accomplished through the use of gaussian smoothing (in space and time). The temporal smoothing is generally computed, centering the filter kernel in the image to be processed. This procedure requires not only past images, but also images to be acquired *after* the time instant under consideration (a non causal filter).

[4]Usually, these methods impose smoothness constraints on the optical flow field using some sort of regularization technique, or relying on second or higher order time-space image derivatives [Micheli et al., 1988, Uras et al., 1988, Otte and Nagel, 1994] which tend to be a very ill-posed problem.

The time delay introduced with this procedure, becomes relevant when the visual information is to be used for real-time control because it introduces a lag in the feedback control loop. In our approach, instead, having the control application in mind, a *causal* first order time filter has been used, which recursively updates the time smoothed image :

$$\begin{aligned} I_s(0) &= I(0) \\ I_s(t) &= \lambda I_s(t-1) + (1-\lambda) I(t), \quad \text{with } 0 \leq \lambda \leq 1 \end{aligned} \tag{3.7}$$

where $I(t)$ and $I_s(t)$ stand for the image acquired at time t and the corresponding *temporal smoothed* image. The parameter λ controls the desired degree of smoothing. When λ is larger, the images are more smoothed in time. Using the recursive time filtering procedure, only present and past images are required for the time filtering process [5].

The spatial smoothing is performed by convolving the time smoothed images, with a gaussian filter, and the time derivative is simply computed by subtracting two consecutive space and time smoothed images.

To speed up the optical flow computation, a set of five images is acquired at video rate and the temporal smoothing of both, left and right image sequences, starts concurrently to the acquisition. The last two images (on each side) of the time smoothed sequence, are then used to compute the average, left and right, optical flow vectors. Finally, the difference between the average flow on the left and right images is used to synthesize the control law. Then, a new sequence of 5 images is acquired, and the whole process is repeated. In this way, the images are sampled at video rate even if the complete system operates at a slower rate (determined by the computation time which varies with the number of non-zero flow vectors).

The images acquired have a resolution of 256x256 pixels, and the optical flow is computed over a window of 32x64 pixels on each side. In the current implementation, the averaged optical flow vectors are computed at a frequency of approximately 1.5Hz.

In order to clarify the *Divergent Stereo* approach, we performed an open-loop experiment, illustrating the obstacle detection capabilities. Figure 3.2 shows the experiment setup and an image with the flow vectors superimposed. The vehicle is moving at a

[5]Regarding memory storage, it is only necessary to store $I(t)$, $I_s(t)$ and $I_s(t-1)$.

forward speed of 80mm/s. Along the rectilinear trajectory, the robot will pass midway between the left wall and an obstacle placed on the right side. At this point, the robot is centered relatively to the left wall and the obstacle, and the bilateral flow difference should be 0.

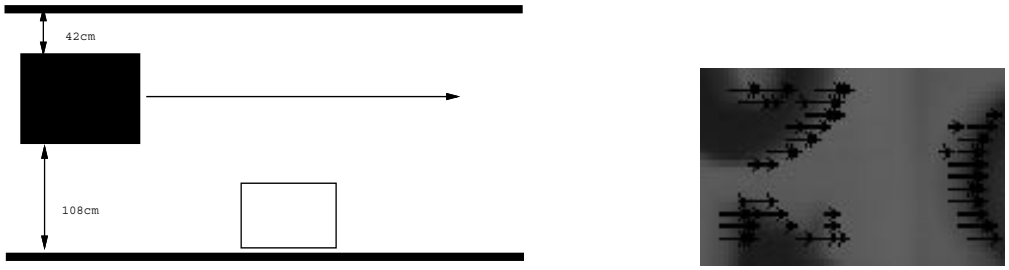

Figure 3.2: Experimental setup for the obstacle detection experiment. A sample of the computed optical flow is shown on the right.

The evolution in time of the difference between the left and right average flow fields is shown in Figure 3.3. As expected, the difference is initially positive, as the robot is closer to the left than to the right wall and, when approaching the obstacle, the error tends to zero, as the vehicle is approximately centered with respect to the obstacle and the left wall. After passing the obstacle, the difference tends approximately to the initial value, again, as the robot proceeds, following a rectilinear path.

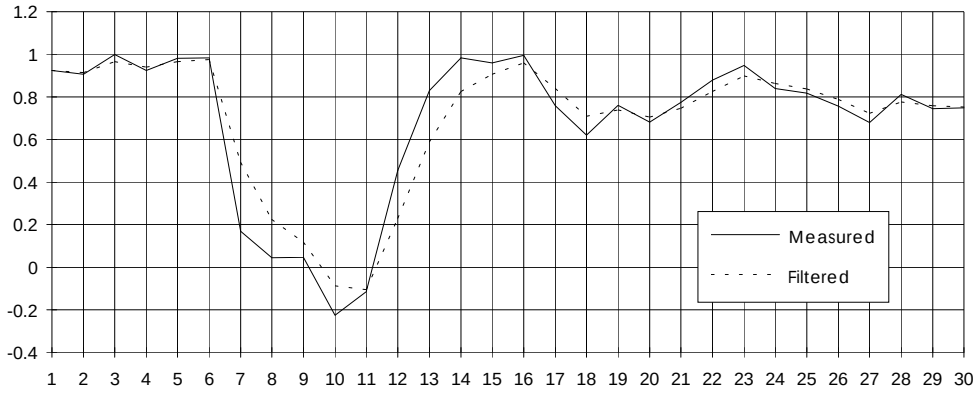

Figure 3.3: Difference between the left and right average flows, during the obstacle detection experiment. The full path takes 20 seconds, covering a distance of about 2.4m. The filtered signal (dashed line) results from applying a first order filter to the measured values.

This experiment shows the sensitivity of the proposed perception process when performing an obstacle detection task, thus motivating its use in a closed loop fashion. An important remark is that the system does not critically depend on the accuracy of the optical flow computation, because the measurements are used continuously in closed loop.

3.3.1 Analysis of rotational flow

The vision-based centering reflex relies on the assumption that the optical flow amplitude is only dependent on the distance between the cameras and the environment. Strictly speaking, this assumption is valid if the two cameras are pointing symmetrically with respect to the heading direction, and the heading direction does not change. However, this constraint does not hold during the rotational motion, necessary to adjust the heading direction, because the roto-translation of the cameras introduces a component on the flow field which does not depend on distance.

This section presents a detailed analysis of the rotational flow of the *Divergent Stereo* setup, and explains how the disturbances introduced during robot rotations, can be reduced and eventually made irrelevant, by appropriately positioning the cameras with respect to the vehicle rotation center and by tuning some of the centering reflex control variables.

Figure 3.4 describes the set of coordinate systems associated to the robot. Let the robot coordinate system be denoted by {X,Y}, and the left/right cameras coordinate systems by {x,y}. The setup is symmetric and the right camera optic center has polar coordinates (ρ, θ) with respect to the origin of the robot coordinate system.

The translational component of the camera motion, $(\dot{X}, \dot{Y})$, due to the rotation of the robot around its geometric center is given by :

$$\begin{aligned} \dot{X}_L &= -\rho\,\dot{\theta}\,\sin\theta & \qquad \dot{X}_R &= -\rho\,\dot{\theta}\,\sin\theta \\ \dot{Y}_L &= -\rho\,\dot{\theta}\,\cos\theta & \qquad \dot{Y}_R &= \rho\,\dot{\theta}\,\cos\theta \end{aligned}$$

where the subindex refers to the left or right cameras. By expressing this motion parameters in the cameras coordinate systems, it yields :

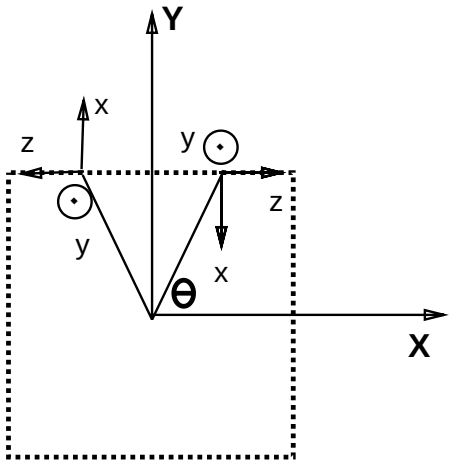

Figure 3.4: Coordinate systems associated to the vehicle for the analysis of the rotational motion.

$$\begin{aligned} \omega_y^L &= \dot{\theta} & \omega_y^R &= \dot{\theta} \\ T_x^L &= -\rho\,\dot{\theta}\,\cos\theta + T_M & T_x^R &= -\rho\,\dot{\theta}\,\cos\theta - T_M \\ T_z^L &= \rho\,\dot{\theta}\,\sin\theta & T_z^R &= -\rho\,\dot{\theta}\,\sin\theta \end{aligned}$$

where T_x, T_z denote the x and z components of camera linear velocity, ω_y is the y component of the camera angular velocity and T_M is the robot forward speed.

The influence of the rotation can be perceived in the following equations describing the horizontal component, u, of the image motion field :

$$\begin{aligned} u_L &= \frac{1}{Z_L}\left[x_a\rho\,\dot{\theta}\,\sin\theta + \rho\,\dot{\theta}\,\cos\theta - T_M\right] - (1+x_a^2)\,\dot{\theta} \\ u_R &= \frac{1}{Z_R}\left[-x_a\rho\,\dot{\theta}\,\sin\theta + \rho\,\dot{\theta}\,\cos\theta + T_M\right] - (1+x_a^2)\,\dot{\theta} \end{aligned}$$

where x_a denotes an image point coordinate expressed in units of focal length.

Observing that the left and right flow fields have opposite directions, (due to the choice of the cameras coordinate systems) the comparison of both left/right lateral flows is given by the sum of u_L and u_R :

$$\begin{aligned} e &= u_L + u_R & (3.8) \\ &= \left[T_M - x_a\rho\,\dot{\theta}\,\sin\theta\right]\left(\frac{1}{Z_R} - \frac{1}{Z_L}\right) + \rho\,\dot{\theta}\,\cos\theta\left(\frac{1}{Z_R} + \frac{1}{Z_L}\right) - 2(1+x_a^2)\,\dot{\theta} & (3.9) \end{aligned}$$

In the absence of rotation, this equation is simplified to :

$$e = T_M \left(\frac{1}{Z_R} - \frac{1}{Z_L} \right). \tag{3.10}$$

This equation shows that, without rotation, the error signal, e, is directly proportional to the deviation from the center trajectory. The robot forward speed appears as a scaling factor which has the role of a *sensitivity gain*. However, if the rotational motion is important, the circumstances under which the lateral flow comparison is still meaningful as a trajectory deviation measurement, have to be considered. There are three main contributions to be analysed :

1. In the first term, $[T_M - x_a \rho \dot{\theta} \sin\theta](\frac{1}{Z_R} - \frac{1}{Z_L})$, the rotation affects the *sensitivity gain*. Avoiding large variations of this gain, which would influence the system closed-loop behaviour, introduces a constraint between the maximum rotation speed and the robot forward speed, T_M. This requirement can be met by suitably selecting the values of θ and ρ, or increasing the vehicle speed.

2. The term $\rho \dot{\theta} \cos\theta(\frac{1}{Z_R} + \frac{1}{Z_L})$ depends on the unknown 3D structure but does not convey any information on the deviation from the optimal trajectory. It can be made small enough, by installing the cameras closer to the vehicle rotation center, hence reducing ρ.

3. The term $-2(1 + x_a^2)\, \dot{\theta}$ is not dependent on the 3D structure of the environment. A *calibration* procedure could be designed to compute the correspondence between rotational speed (which is internally controlled) and displacements in the image plane. During normal operation, this term can be compensated.

3.3.2 Design specifications

The problem of determining a set of design constraints to overcome the problems due to the rotational motion will be analysed in this section. The analysis will be restricted to the structure dependent terms which appear in equation (3.9).

Concerning the first term, it can be seen that the rotation leeds to changes in the *sensitivity* of the error signal to the trajectory deviation factor (see equation (3.10)).

Very fast rotations could even lead to unstability. Therefore, a natural constraint will be such that the rotation term does not affect, too significantly, the *sensitivity gain* :

$$x_a \rho \, \dot{\theta} \, \sin\theta < \alpha T_M \tag{3.11}$$

where $\alpha \in [0, 1]$ is the admissible relative change in the *sensitivity gain*. Alternatively, the second term in the error signal should also be kept small relatively to T_M :

$$\rho \, \dot{\theta} \, \cos\theta < \beta \, T_M \tag{3.12}$$

where $\beta \in [0, 1]$ quantifies the admissible relative magnitude of this term.

A suitable setup (ρ, θ) can be chosen to meet these specifications. There are two other ways to satisfy these constraints in practice. The first one, consists in fixing the value for the forward speed, and limiting the value of $\dot{\theta}$, by saturating the control variable before it is applied to the robot (in natural systems this may be an intrinsic constraint) :

$$\dot{\theta} < \min\left(\frac{\alpha \, T_M}{x_a \rho \, \sin\theta}, \, \frac{\beta \, T_M}{\rho \, \cos\theta} \right). \tag{3.13}$$

Another way of addressing the problem consists in having an extra control loop, that dynamically adjusts the vehicle forward speed, to fulfill the design constraints :

$$T_M > \max\left(\frac{x_a \rho \, \dot{\theta} \, \sin\theta}{\alpha}, \, \frac{\rho \, \dot{\theta} \, \cos\theta}{\beta} \right). \tag{3.14}$$

As a numerical example, let the values of the current setup be considered :

$$\begin{aligned} \theta &= 72^o \\ T_M &= 0.08 \;\; m/s \\ \rho &= 0.34 \;\; m \, . \end{aligned}$$

Since the focal length of the lenses is $4.8mm$, roughly the same order of magnitude as the size of the CCD chip, a reasonable approximation for the upper bound of x_a is $x_a \simeq 1$. The following constraints arise from these settings :

$$\dot{\theta} < \min(\, 0.247\alpha, \; 0.759\beta \,) \;\; rad/s \, . \tag{3.15}$$

For example, by setting the project parameters α and β to 0.5 and 0.2, respectively, yields[6] :

$$\dot{\theta} < 7.1 \ \ deg/s \ . \tag{3.16}$$

To conclude, one can say that through careful placement of the camera system and choice of some design parameters, the constraints derived in this section can be met. It is worth stressing, however, that the proposed solution only represents an approximation based on qualitative observations. On the other hand, the experiments performed, clearly show that the desired behaviour is obtained and that the choice of the design parameters is by no means critical, illustrating the robustness of the approach proposed.

Obviously, another possibility to overcome the rotation problem is to inhibit the control action whenever the robot is required to rotate faster than the maximum allowable speed (a sort of "saccadic" motion which resembles very much the behaviour of insects). During our experiments we have tried several approaches such as using the design specifications explained in this section or adopting the "inhibitory" solution, by actually disregarding the flow measurements during fast rotations. *Robee* performed equally well in both situations.

The solution adopted by Coombs and Roberts is also satisfactory. However, the higher complexity, introduced by the stabilization of the cameras against head rotations and by alignment of the eyes with the heading direction, does not seem to improve behavioural performance.

3.4 Real-time Control

The overall structure of *Robee* control system involves two main control loops :

1. Navigation loop - Controls the robot rotation speed in order to balance the left and right optical flow.

2. Velocity loop - Controls the robot forward speed by accelerating/decelerating as a function of the amplitude of the lateral flow fields[7].

[6] The choice of these numerical values is a way of specifying a threshold relative to some known nominal quantity.

[7] *Robee* accelerates if the lateral flow is small (meaning that the walls are far away), and slows down whenever the flow becomes larger (meaning that it is navigating in a narrow environment).

Additionally, a *sustaining mechanism* is implicitly embodied in the control loops to avoid erratic behaviours of the robot, as a consequence of localized (in space and time) absence of flow information. These aspects and the analysis of the different control loops will be presented in the following sections.

3.4.1 Navigation Loop

The analysis of the control loops is based on simplified dynamical models of the control loop components and on the use of linear systems theory[8].

The robot heading direction (*controlled variable*) is controlled by applying a rotation speed (*control variable*) superimposed on the forward motion. The simplest dynamic model of the system must account for two important terms : an integral term relating the input angular speed and the output angular position; the mechanical inertia of the system, which is modeled by a first order dynamic system. In continuous time, the transfer function relating the *control* and *controlled* variables, according to our simple model, is given by :

$$G_c(s) = \frac{a}{s\,(s+a)} \tag{3.17}$$

where a is the dominant low-frequency mechanical pole.

Since we are using digital control, we must determine how the computer (where the control algorithm is implemented) "sees" the system. As a sample-and-hold mechanism is being used, the *step invariant* method is appropriate to determine the discretized system [Astrom and Wittenmark, 1986]. Considering the low-frequency pole at 5 Hz, and a sampling period of $\tau = 0.7s$, we obtain :

$$G_d(z) = \frac{0.468z + 0.0318}{z^2 - (1 + 1.51e^{-7})z + 1.51e^{-7}}\ . \tag{3.18}$$

Again for simplicity, we may assume that the difference between the left and right flow vectors, provides the *error*, e, between the robot direction, ξ^{robot}, and the direction of the lateral walls, ξ^{walls}, with a delay of one sampling period introduced by the flow computation. This approximation is only valid for small course deviations, since the visual

[8] A more accurate control analysis/synthesis is undergoing.

process is, in fact, non linear. The sensor model is then given by :

$$e_{(k)} = \zeta^{walls}_{(k-1)} - \zeta^{robot}_{(k-1)} \, . \tag{3.19}$$

Qualitatively, $e(k)$ is positive if the left side flow is larger than the right side flow, which means that the left wall is closer than the right one. Hence, the appropriate control action would be turning to the right. The discrete time PID controller, that is used to close the navigation loop, performs the following control law :

$$u_{(k)} = K_p[\;\; e_{(k)} + K_i \sum_n e_{(k-n)} + K_d(e_{(k)} - e_{(k-1)}) \;\;] \tag{3.20}$$

where u is the control variable (rotation speed in degrees/s) and $e_{(k)}$ stands for the *error signal* observed at time instant k. The transfer function corresponding to the PID is given by :

$$G_{PID}(z) = K_p \frac{(K_i + K_d + 1)z^2 - (1 + 2K_d)z + K_d}{z(z-1)} \, . \tag{3.21}$$

By connecting all these models, we obtain a linear feedback loop as shown in Figure 3.5.

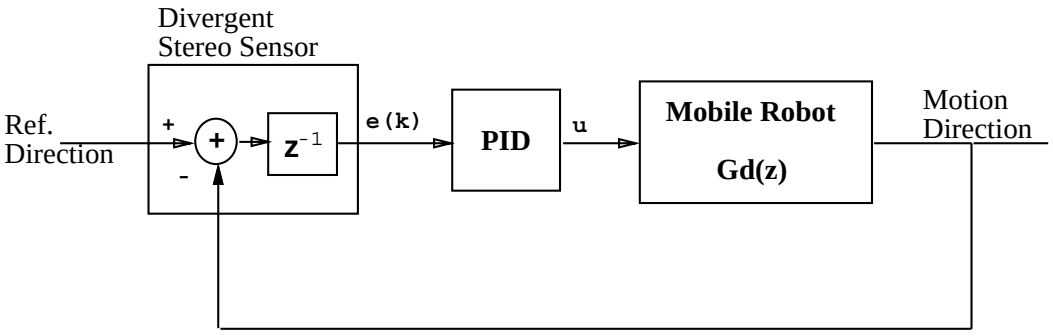

Figure 3.5: This block diagram illustrates a simple modelization of the robot navigation system

Although a thorough analysis of the system behaviour, with different control settings, can hardly be done based on approximate models, a discussion can still provide us some insight and understanding on the performance of several classes of controllers.

As a start, the loop could be closed simply by using a proportional gain (by setting the integral and derivative gains to zero). The root-locus corresponding to this situation is shown in Figure 3.6.

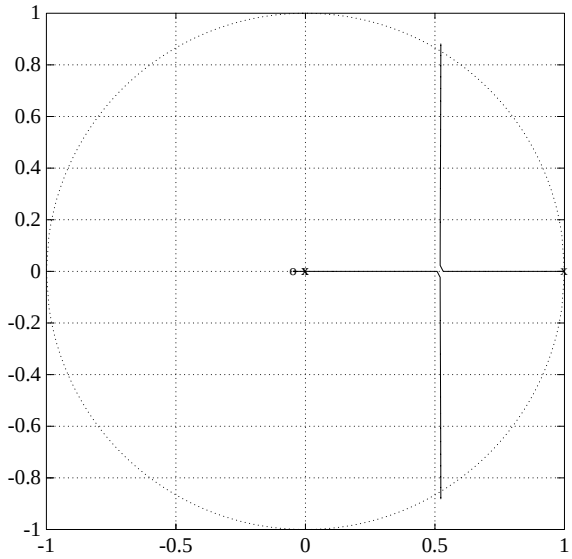

Figure 3.6: Root-Locus corresponding to the proportional controller. The unit circle is shown in dotted line for easier stability analysis.

Even though the proportional controller succeeds in stabilizing the system, fast responses can only be attainable with large values of gain which lead to significant oscillatory behaviour, as the dominant poles become complex conjugate.

By adding a derivative component to the controller (which basically works as a predictive term, thus coping better with the delay), one can expect to achieve faster responses, even for smaller gain values. Figure 3.7 shows the root-locus in this situation. The derivative component is fixed to $K_d = 1.5$. The effect of adding the extra zero is that the low frequency pole is attracted into the higher frequencies, thus improving the response time of the system. Also, for large gains the oscillatory behaviour of the system is reduced.

Finally, the effect of inserting the integral component in the controller can be analysed by considering the root-locus shown in Figure 3.8. The parameters used are $K_i = 0.1$ and $K_d = 1.5$. In this last situation, adding a discrete integrating effect (an extra pole in 1), decreases the stability margin of the system, as well as the response speed. In practice, the system may even become unstable due to various unmodeled components or perturbations, such as the non linearities in the visual processing.

This simplified analysis, allowed us to gain important insight and understanding on the

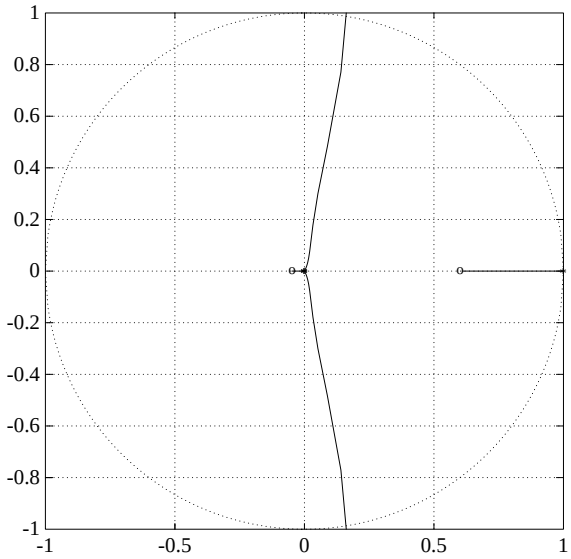

Figure 3.7: Root-Locus corresponding to the proportional-derivative controller. The derivative gain was fixed at $K_d = 1.5$.

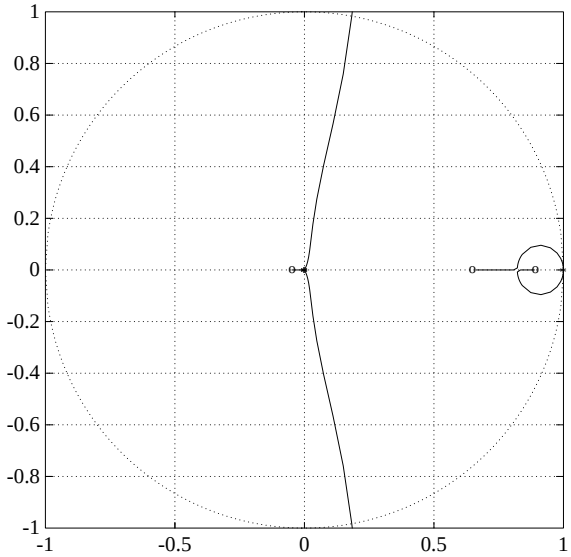

Figure 3.8: Root-Locus corresponding to the PID controller.

behaviour of the system under the different controller topologies. Further modeling is still necessary for a quantitative analysis or a more systematic control synthesis. Nevertheless, some of the ideas discussed in this section were later on verified in the experiments, as described in Section 3.5.

3.4.2 Velocity Control

This section presents the strategy proposed to control the robot forward speed based on the environment structure. The rationale is that, if the robot is navigating on a narrow environment, it is safer to decrease the forward speed; whereas if it is moving on a wide clear area, it is reasonable to increase its speed.

The mean flow vector on each side of the robot gives a qualitative[9] measurement of depth. By averaging these bilateral mean flow vectors, a qualitative measurement of width is obtained.

The control objective amounts to keeping the average flow close to some specified reference value. If the flow increases, the robot should slow down, thus reducing the observed flow. This behaviour, which can be implemented within the purposive approach described here, is not only coherent from the perceptual viewpoint (it agrees with what a human driver, for example, would do) but also increases the safety. Qualitatively, this corresponds to saying that the size of the environment is scaled by the robot speed.

Let T_o be the nominal speed, at which the robot should move, and let f_o be the corresponding nominal flow. For safety purposes, the robot speed, T, is constrained to the interval $[(1-\beta)T_o,\ (1+\beta)T_o]$, where $\beta \in [0,1]$ quantifies the permissible excursion. A sigmoid function is used as smooth saturation, as shown in Figure 3.9.

The velocity to be applied to the robot is given by :

$$T = T_o \left[1 - \beta + \frac{2\beta}{1 + e^{\delta(f - f_o)}} \right] \tag{3.22}$$

where f is the average between the left and right flows, and δ determines how fast the speed variation should be with respect to the flow variation. To determine δ, one can say, for example, that 90% of the total velocity excursion should be reached for a relative

[9] Qualitative in the sense that it is not a measure of depth, and, moreover, depends on the robot speed.

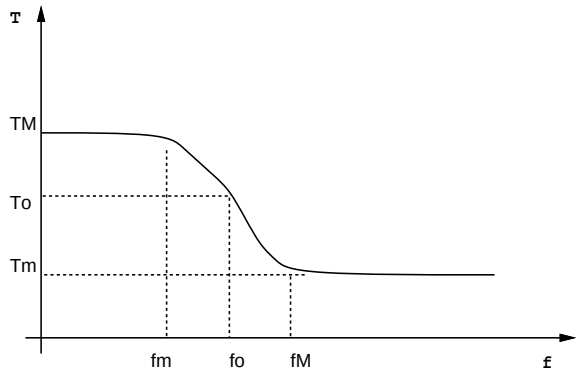

Figure 3.9: The sigmoid function, used in the velocity control loop, describes how the robot forward speed should change, with relation to the measured flow. It is used to introduce a smooth saturation on the robot velocity.

variation of ϕ around the nominal flow. By using equation (3.22), δ is given by :

$$\delta = \frac{\ln 19}{\phi f_o} \,. \tag{3.23}$$

In the current implementation, we have used $\beta = 0.5$, $\phi = 0.3$, and a reference flow of $f_o = 2.0$ pixels/frame, yielding $\delta = 4.95$.

3.4.3 "Sustained" behaviour

The navigation system, described in the previous sections, allows the robot to navigate by balancing the flow measurements on the left and right sides. Therefore, it can only be applied as long as there is texture on both sides of a corridor–like environment. This situation is not entirely satisfactory for two reasons :

- It would be nice if the reactive behaviour of *Robee* could be used in environments far more complex than corridors.
- In most mission scenarios, *Robee* would most certainly find environments with "walls" not uniformly covered with texture. This fact would cause an illusory perception of infinite distance and, therefore, elicit unappropriate behaviours (for example, if an open door is found while traveling along a corridor).

To overcome these problems, we have introduced in the control strategy a mechanism that is able to cope with unilateral lack of flow information. Whenever it occurs, the

control system uses a reference flow that should be sustained on the "seeing" camera (i.e. the camera still measuring reliable flow).

This mechanism monitors whether there is significant flow being measured on both sides of the vehicle, or if just one (or none) of the cameras is capturing significant flow. Three situations may arise :

- **Bilateral flow -** Optical flow is measured in both cameras. Consequently, the robot is locally navigating in a corridor, and the standard navigation strategy can be applied

- **Unilateral flow -** Only one camera is capturing flow information. Without the *sustaining* mechanism, the robot would simply turn towards the side without flow measurements, trying to balance lateral flows. A more appropriate behaviour, instead, would be keeping the unilateral flow constant, hence following the ipsilateral wall at a fixed distance. With such strategy, the robot may cross corridors with open doors, even with partially untextured walls and, when arriving to a room, follow the walls at a fixed distance.

- **Blind -** If none of the cameras is capturing flow information, the robot is virtually blind. This robot should either stop or follow straight ahead for a while, or even wander in a random exploratory way, until some texture is again found. Currently, *Robee* stops if this situation arises.

Let us analyse how, without any prior knowledge of the environment, the *sustaining* mechanism is implemented . During normal operation, the reference flows are estimated by filtering over time each of the lateral mean flow vectors. The time filtering takes into account the number of vectors that contributed to the mean computation :

$$\bar{f}_{(t)} = \frac{\alpha \bar{n}_{(t-1)} \bar{f}_{(t-1)} + (1-\alpha) n_{(t)} f_{(t)}}{\alpha \bar{n}_{(t-1)} + (1-\alpha) n_{(t)}} \tag{3.24}$$

$$\bar{n}_{(t)} = \frac{\alpha \bar{n}^2_{(t-1)} + (1-\alpha) n^2_{(t)}}{\alpha \bar{n}_{(t-1)} + (1-\alpha) n_{(t)}} \tag{3.25}$$

where $f_{(t)}$ is the mean flow, computed at time t, with $n_{(t)}$ flow vectors; $\bar{f}_{(t)}$, $\bar{n}_{(t)}$ are the corresponding *time filtered* values; and $\alpha \in [0, 1]$ is a time decay constant. Functionally,

α determines the amount of past flow information that should be "remembered" in the filtering process.

The system enters the *sustaining mode* whenever it is unable to estimate reliable flow vectors on one side. In the experimental tests, α was set to 0.6, which agrees with the time filtering parameter used for temporal smoothing.

As a final remark, we would like to stress that the proposed approach extends considerably the performance of the "reactive" behaviour and that the use of both the *corridor* and the *wall* following behaviours in task–driven navigation is straightforward. In particular, two potential applications are worth mentioning :

- The "reactive" control of *Robee* can be used to acquire information about environmental structure. In fact, odometric information, coupled with the sensory information of *Robee*, would allow to build a map of the environment in terms of *corridors* and *walls* which could be used by a planning system to drive navigation on a higher level.

- The planner of a robot moving in a known (but variable) environment could take advantage of "task–level" commands like : "navigate to the end of the corridor" or "follow a wall" without relying on geometric information which, ultimately, may be difficult to acquire and maintain in realistic environments.

3.5 Results

This section presents the results of a set of tests that clearly demonstrate several applications of the *Divergent Stereo* navigation approach. The goal of the experiments is the study of the closed loop behaviour of the visually guided robot. In order to test the performance in a wide range of environmental situations and to analyse in more detail the influence of the different controller settings, three experimental scenarios with increasing degree of difficulty have been considered. The final experiment illustrates the *sustaining* mechanism along with the velocity control loop. In all the results presented the trajectory of the robot was recorded from odometric data during real-time experiments.

3.5.1 Turn Experiment

On the first set of experiments, the robot was tested in a turning corridor setup, as shown in Figure 3.10.

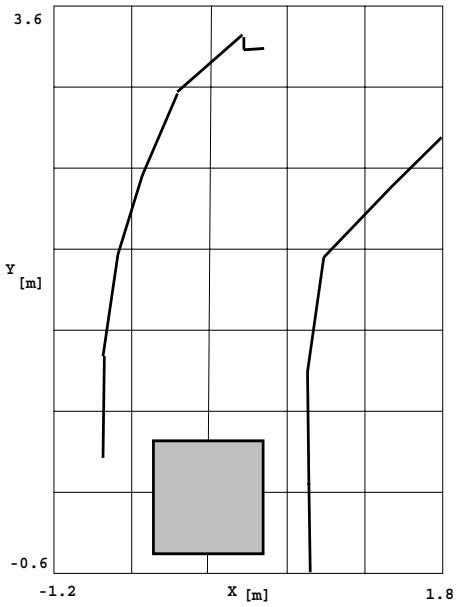

Figure 3.10: Setup used for the *turn* experiment. The vehicle is supposed to perform the turn, using the divergent stereo navigation strategy.

The whole navigation system has been tested with different controller settings. For these experiments, the integral gain K_i, was fixed to zero, as suggested by the discussion on the controller design.

In the first three experiments, the influence of the derivative gain on the navigation performance has been studied. Figure 3.11, shows the trajectories followed by the robot with a fixed proportional gain, $K_p = 1.5$, and using for K_d the values of 1.0, 1.2 and 1.4. The trajectories were recorded using the odometry information, and are shown superimposed on the experimental setup.

To evaluate the effect of the proportional gain on the controller, another set of trials was performed. The integral and derivative gains were kept constant ($K_i = 0$ and $K_d = 1.2$), while using for K_p the values of 1.25, 1.5 and 1.75. The results are shown in Figure 3.12.

The analysis of the trajectories show that, by increasing the value of K_d, the response

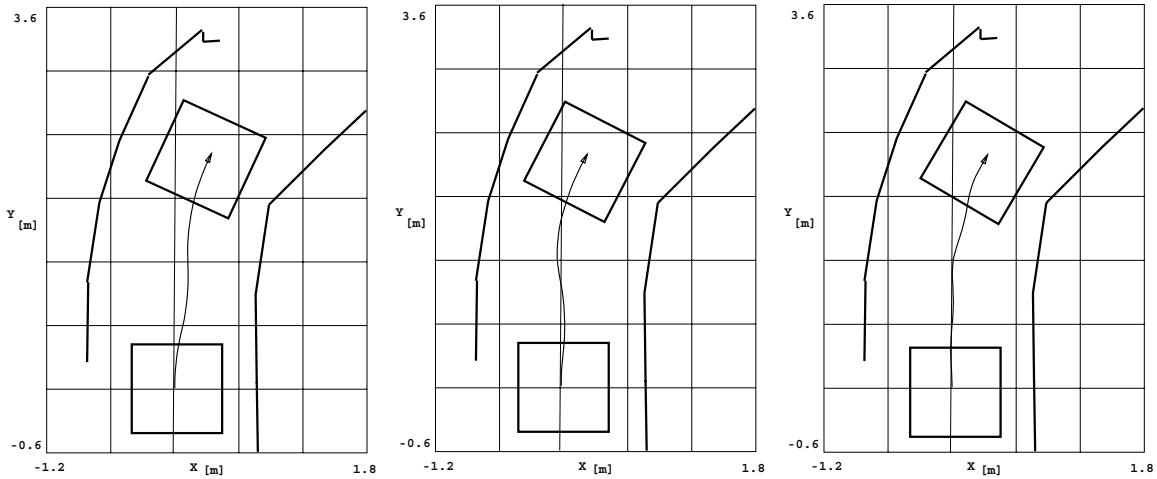

Figure 3.11: Results obtained in the closed loop operation in the turn experiment. The controller settings were $K_p = 1.5$, $K_i = 0$ and using for K_d the values (from left to right) of 1.0, 1.2 and 1.4. The trajectory was recovered using odometry.

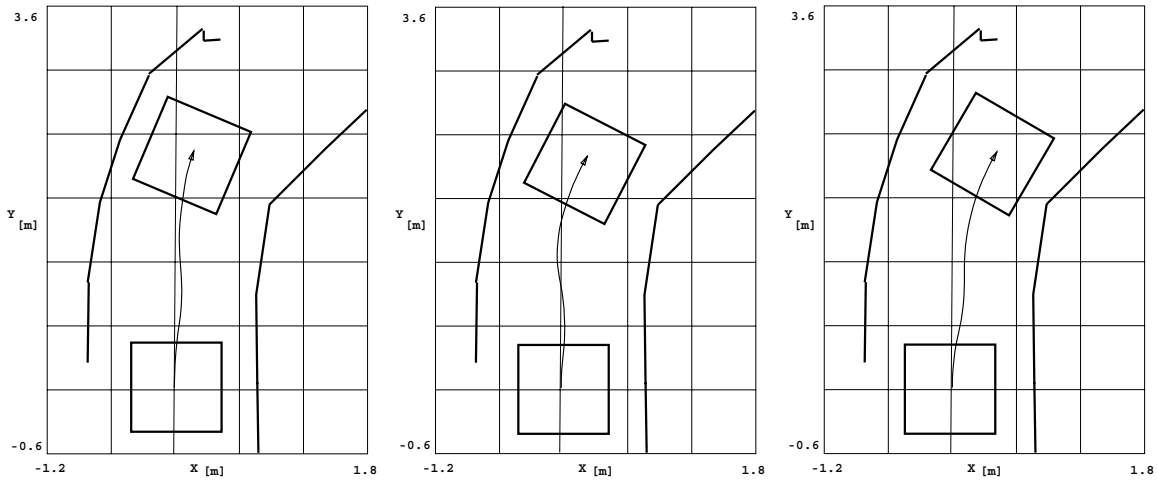

Figure 3.12: Results obtained in the turn experiment (closed loop operation). The controller settings were $K_d = 1.2$, $K_i = 0$ and using for K_p the values (from left to right) of 1.25, 1.5 and 1.75. The trajectory was recovered using odometry.

becomes faster, even though the trajectories may become less smooth. On the other hand, by increasing K_p, the response becomes faster but not as fast as when the derivative component is increased

3.5.2 Funnel

In another set of experiments, we used a funneled corridor with an obstacle. The setup is shown in Figure 3.13. When navigating in this environment, the robot must avoid the obstacle while trying to keep centered in the funneled corridor. This situation is particularly interesting because it forces the robot to react to sudden changes (the obstacle) as well as to smooth changes of the environment structure.

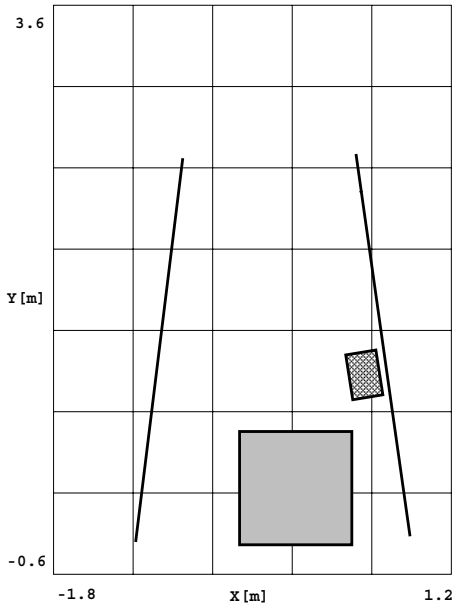

Figure 3.13: Setup used for the funneled corridor experiment. In this scenario, the robot has to avoid an obstacle while managing to keep the track in the funneled corridor.

Again, different settings of the PID controller have been used, in order to study the robot behaviour. Figure 3.14 shows the trajectories corresponding to three of those experiments. The robot trajectory, recovered from odometry, is superimposed on the setup layout.

The first trial represents an experiment with the controller tuned with $\{K_p = 1.5$, $K_i = 0.0$, and $K_d = 1.5\}$, which led to a nice trajectory. On the second trial, the integral

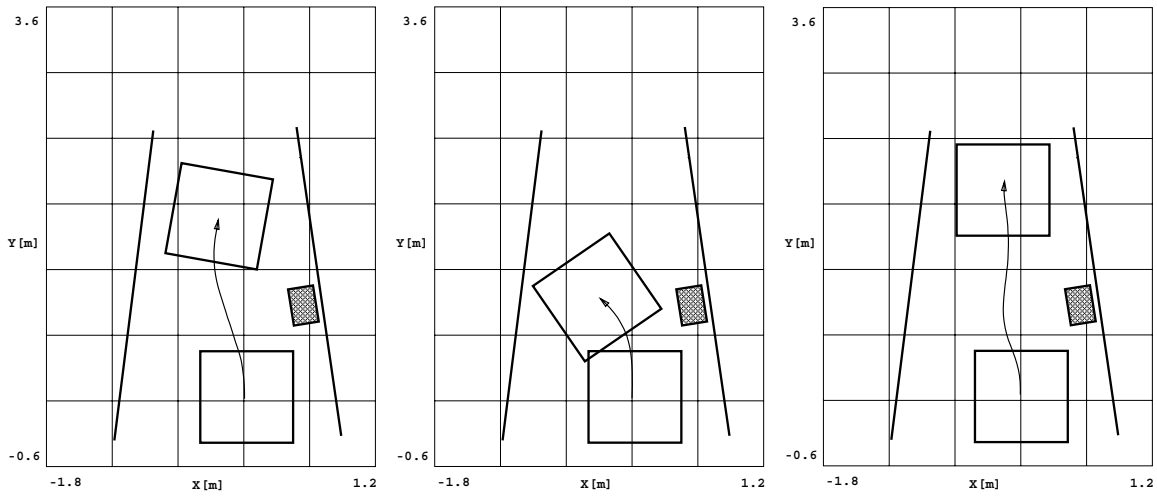

Figure 3.14: Funneled corridor experiment. The leftmost and rightmost plots show the results of increasing the proportional gain while decreasing the derivative gain. In the center, it is seen the unstable behaviour due to the insertion of the integral action.

component has been introduced using $\{K_p = 1.5,\ K_i = 0.1,\ K_d = 1.5\ \}$. As suggested by the discussion made in Section 3.4, an unstable behaviour was observed, with the vehicle moving towards the left wall. Finally, on the third experiment a nice performance was also obtained by slightly increasing the proportional gain, while diminishing the derivative gain $\{K_p = 1.8,\ K_i = 0.0,\ K_d = 1.2\}$. As expected, a somewhat smooth trajectory was obtained.

The tests performed, enhance the importance of suitable control system design, and clearly show the aim of our discussion of the modeling and control system design. Even if further improvements on the vehicle dynamics modeling are certainly necessary, it is worth noting that the behaviour of the robot does not depend upon critical values of the controller settings and, therefore, the robustness of the system is very promising.

3.5.3 Corridor

In this set of experiments, a corridor which is just slightly larger than the robot and with a sharp turn in the end has been used (see Figure 3.15) to test the performance of the system on a combination of different environmental situations.

Also in this situation, several trials were performed, changing the parameters of the

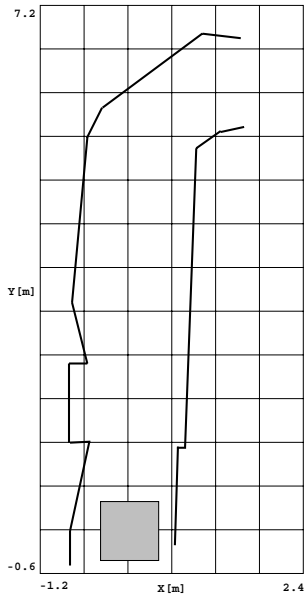

Figure 3.15: Setup used for the corridor experiment. The corridor is just slightly larger than the robot, and has a tight turn in the end.

PID controller. Figure 3.16 shows the trajectories obtained by increasing the values of K_p. The derivative gain is fixed in $K_d = 1.2$, while the proportional gain increases from left to right $K_p = 1.2,\ 1.4,\ 1.6$. The integral gain is not used.

On the second set of trials, the proportional gain is set to $K_p = 1.4$, and the derivative gain is increased from left to right $K_d = 1.0,\ 1.2,\ 1.4$. Again, the integral gain is not used. The trajectories are shown in Figure 3.17.

The results show that by increasing, both the proportional and derivative gains, the response becomes somewhat faster. In general, by increasing the derivative gain leads to a faster behaviour of the control system.

3.5.4 Velocity Control

In order to test the velocity control, the robot was navigating through a funneled corridor, where the width changes from 1.65m to 1.25m. The full length of the funneled corridor is about 2.25m.

Since the corridor is becoming narrower, the average flow increases. By introducing the velocity control mode, the robot velocity will decrease in order to keep a constant flow.

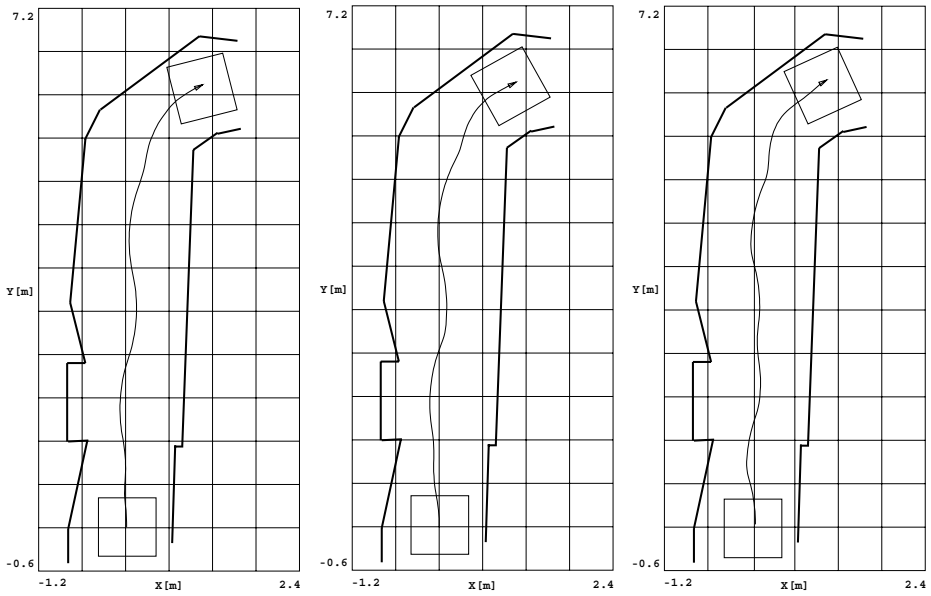

Figure 3.16: Corridor experiment. The derivative gain is fixed in $K_d = 1.2$, while the proportional gain increases from left to right $K_p = 1.2, 1.4, 1.6$. The integral gain is not used.

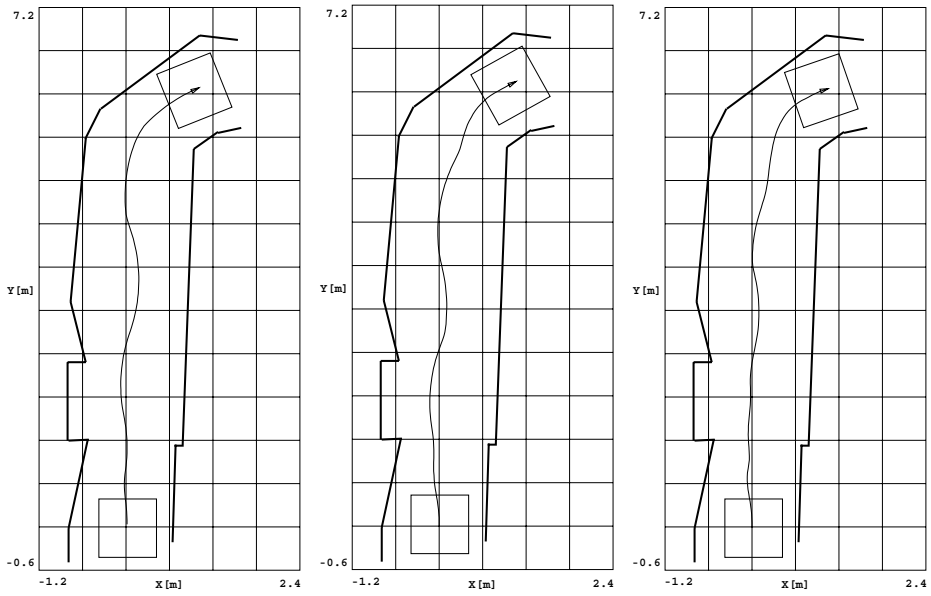

Figure 3.17: Corridor experiment. The proportional gain is fixed in $K_p = 1.4$, while the derivative gain increases from left to right $K_d = 1.0, 1.2, 1.4$.

Figure 3.18 shows the average between the left and right flows measured over time. The filled line was obtained without velocity control (showing increasing flow values), while the dotted line shows the action of the velocity control keeping image flow close to the desired value of 2 pixels/frame.

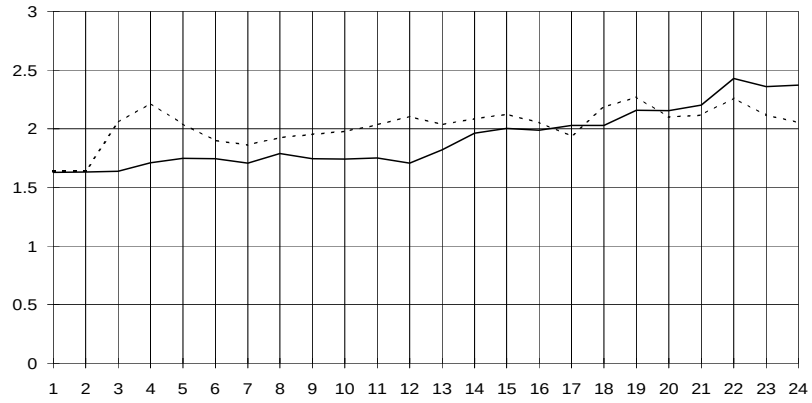

Figure 3.18: The average between the left and right flows obtained in an experiment with (dotted line) and without (filled line) velocity control. The nominal flow is 2.0 pixels/frame and the nominal velocity 80mm/s.

The evolution of the robot velocity along the path is shown in Figure 3.19. It is seen that, as the corridor is narrowing, the velocity decreases in order to keep the flow to a lower safer value.

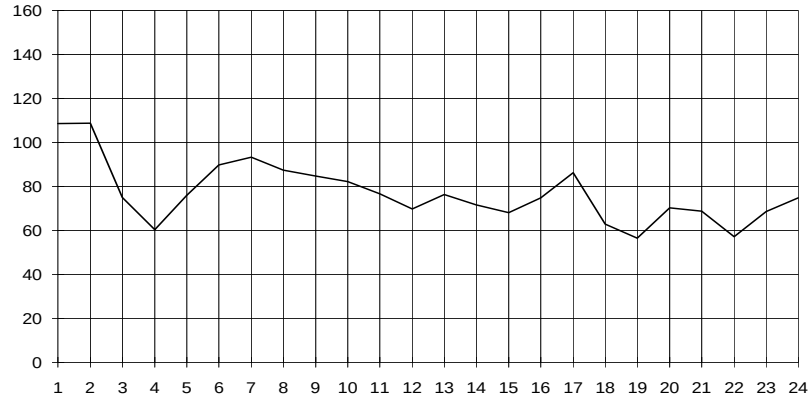

Figure 3.19: The robot speed in mm/s during the funneled corridor experiment. As the corridor narrows, also the velocity decreases for safer operation.

Other experiments were performed using the proposed strategy. Particularly, it is

interesting to see that, during the corridor experiment, the final turn is done at a reduced speed enabling the robot to make a softer, safer turn.

3.5.5 Sustained behaviour

This experiment was designed to show the influence of the sustaining mechanism upon the robot behaviour and how it can deal with different environment structures. The first experiment consisted in using a corridor which exhibits a lack of texture on the right side, and whose left wall suddenly finishes in a room. Figure 3.20 shows the trajectories obtained by activating (center) and de-activating (left) the sustaining mechanism. In the latter case, the robot turns left trying to balance both lateral flows while, in the former, a reference flow value is sustained leading the robot to follow the right wall.

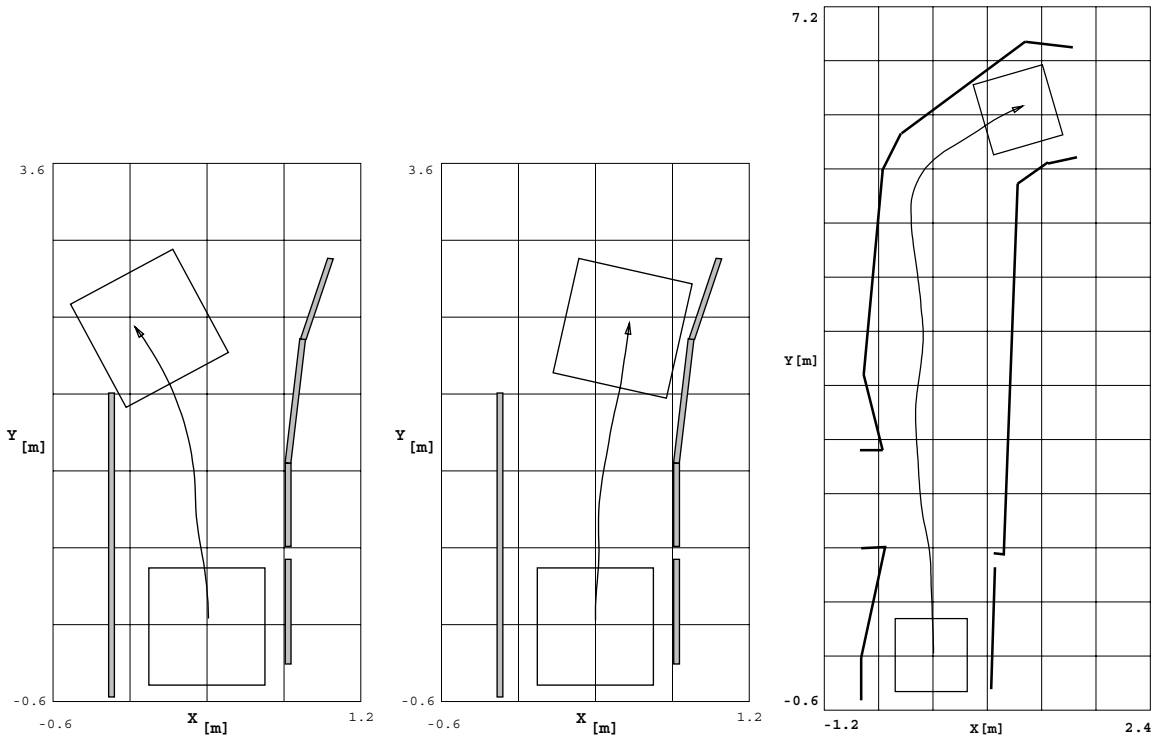

Figure 3.20: Sustained mechanism experiment. The left diagram corresponds to the behaviour of the robot without the sustaining mechanism. At the center, due to the sustaining behaviour, the robot manages to follow the right wall. The rightmost diagram corresponds to an experiment along the corridor with an open door on the left and a lack of texture on the right.

Another experiment was made using the corridor setup, where some texture was removed from the corridor walls, and a door, roughly located midway in the corridor, was left opened. The results are documented in Figure 3.20.

To conclude, we would like to claim that the proposed approach, both on the control and visual perception facets, led to good results and proved the feasibility of a navigation system based on these principles. Furthermore, it should be noticed that with the introduction of the sustained behaviour, the robot is able to navigate in a much wider set of environments. In fact, only one textured wall is needed for the navigation strategy to work.

It is worth noting that, the design principles adopted to eliminate the influence of the rotational component of the optical flow, did prove successful in our experimental conditions and, therefore, even if a stabilization mechanism is desirable or even mandatory in other situations, it is not strictly necessary to produce the kind of behaviours described in this thesis.

3.6 Conclusions

A qualitative approach to visually-guided navigation based on optical flow has been presented motivated by studies and experiments performed on freely flying honeybees.

The approach is based on the use of two cameras mounted on a mobile robot with the optical axes directed in opposite directions such that the two visual fields do not overlap (*Divergent Stereo*). Range is perceived by computing the apparent image speed on images acquired during robot motion.

A real-time computation of optical flow is presented, based upon the constraints imposed by the geometry of the cameras and by the navigation strategy. Furthermore, some suggestions have been presented on how to select some design variables in order to make the disturbances due to the rotational motion, irrelevant to the described reactive behaviours.

A PID controller was used to close the visuo-motor control loop. The closed loop behaviour was studied, based on models of the different control system components. The analysis of the control system design led to a suitable configuration for the PID controller.

The approach has been tested using real-time experiments to accomplish different navigation tasks, like performing a tight turn or navigating through a funneled corridor. The influence of the control parameters on the system behaviour was studied and the results confirmed, for the assumptions made, the discussion on the control system design.

A controller for the robot forward velocity was also studied and implemented. Experiments have been made to show the improvement achievable by including this control loop in cluttered environments.

Finally, through the insertion of a sustained behaviour, the robot is able to navigate in environments rather more complex than a simple corridor, showing the capability of operating in sparsely textured corridors, and following unilaterally textured walls.

All the experiments were performed without the need for accurate depth or motion estimation, nor requiring a calibration procedure (besides the manual positioning of the two cameras).

The main features of Robee can be summarized as follows :

- **Purposive** definition of the sensory apparatus and of the associated processing. In fact, the approach proposed cannot be considered general but, with limited complexity, solves a relevant problem in navigation : the control of heading direction in a cluttered environment.

- Use of **qualitative and direct** visual measures. In our opinion this is not only a "religious" issue but, more importantly, a way to achieve a reasonable autonomy with limited computational power. Successful examples of this approach have recently appeared in the literature both with respect to reflex-like behaviours for obstacle avoidance [Enkelmann, 1990, Ferrari et al., 1991, Fossa et al., 1992, Gaspar et al., 1994, Sandini and Tistarelli, 1990] and in relation to more "global" measures of purposive navigation parameters [Fermüller, 1993a].

- **Continuous use of visual measures**. A further aspect worth mentioning is the attempt made at developing a sensory system providing a continuous stream of environmental information. A first advantage is the increased robustness implicit in the use of repeated measures (no single mistake produces catastrophic errors) and

a secondary, and potentially more important, advantage is the possibility of implementing sensory-motor strategies where the need for a continuous motor control is not bounded by an "intermittent" flow of sensory information. This paradigm is, in our opinion, a non trivial evolution of some active vision implementations where the motion of the (active) observer is seen "only" as a way of taking advantage of the stability of the environment (e.g. by moving the vehicle along preprogrammed, known trajectories to reduce uncertainty). The use of vision **during** action [Sandini et al., 1993a], on the contrary, may be a very powerful extension of the concept of active observer by exploiting the use of dynamic visual information not only at the "reflexive" level of motor control.

- **Simplicity**. This feature is often regarded as an engineering and implementation aspect and, as such, is not explicitly considered a scientific issue. This view, in our opinion, must be changed if reasonable applications of computer vision are addressed. The issue of simplicity, however, should not be considered, within specific aspects of intelligent actor's design (such as, sensory systems, mechanical design, computational architecture etc.) but must be considered at system level. Robee is an example of such an holistic view of simplicity where the purpose is achieved by a comprehensive analysis and integration of visual processing, sensor design, sensor placement, control law and vehicle structure. In this respect low-level animals (and insects in particular) are extremely interesting examples of "simple" actors where all engineering aspects are mixed exploiting not only "computational" issues but, more importantly, the cooperation of "intelligent" solutions which, if considered separately, may look like interesting implementational tricks but, once acting together, produce intelligent behaviours.

Before concluding, it is worth mentioning the fact that *Robee* is blind in the direction of motion (it will bump against an obstacle just in front of it). This problem will be addressed in the next chapter.

On the other hand, for the navigation behaviour, we are using only a very limited portion of the lateral visual field. A more frontal part of the visual field, can be used to extract other motion-derived measures (e.g.time-to-crash) to control, for example, the

docking speed and the heading direction of a mobile robot. These aspects will be focused and presented in Chapter 5.

Chapter 4

Visual Obstacle Detection

As discussed in the previous chapter, the *Divergent Stereo* approach that has been proposed, allows *Robee* to manoeuvre in cluttered environments, but it is blind in the direction of motion, thus being unable to avoid obstacles located ahead. In this chapter, we propose a visual obstacle detection behaviour to overcome this limitation.

4.1 Introduction

Traditional systems for robot navigation tend to perform some kind of map building and use this map to determine a safe trajectory avoiding all the obstacles. As discussed previously (see Chapter 2), this approach is quite sensitive and computationally too demanding to be used in real-time for navigation or obstacle avoidance in mobile robotics, particularly when dealing with dynamic environments. Instead, an alternative solution, within the framework of *purposive vision* [Aloimonos, 1990], may use the specificity of the task under consideration, to propose simpler and more robust approaches for vision guided robotics.

The major hypothesis considered in the approach herein proposed, consists in assuming that the robot is moving on a flat ground plane. This constrains the methodology proposed to indoor scenes. With this assumption, and for the sole purpose of obstacle detection, we propose a system which is fast, robust and independent of a variety of motion or camera parameters.

The visual information used is the normal flow vector field computed over an image sequence acquired by a single camera. A point to be noticed is the assumption that the normal flow (or equivalently the first order temporal-spatial derivatives of an image sequence) is the only flow information available (as discussed in Section 3.3). There is no need, whatsoever, to use extra constraints for the flow field in order to overcome the *aperture problem* as it is usually done in a variety of other methods.

Conceptually, we try to characterize the apparent motion of the ground plane globally, and detect violations to this coherent motion pattern, which can only correspond to points lying outside the ground plane. To determine the presence of the obstacles we use inverse projection techniques as suggested in [Zielke et al., 1990, Mallot et al., 1991]. However, instead of calibrating the extrinsic and intrinsic parameters of the camera as in [Zielke et al., 1990, Mallot et al., 1991], we use the image measurements directly to estimate the projective transformation between the image plane and the ground plane. This transformation is then used to inverse project the flow field onto the ground plane, highly simplifying the interpretation of the flow pattern.

The motion of the ground floor perceived in the image plane, can be fully described by a second order polynomial in the image coordinates [Subbarao and Waxman, 1986]. This parameterization captures the motion of the ground plane as a whole. However, estimating the second-order coefficients of this polynomial leads to highly unstable algorithms [Negahdaripour and Lee, 1992]. Instead, we approximate the motion of the planar surface as an affine transformation and derive a robust parameter estimation procedure uniquely based on the normal flow information.

In an initialization stage, the affine parameters are used to estimate the projective transformation between the image plane and the ground plane (which roughly contains information about the ground plane orientation with respect to the image plane). This transformation is constant since the camera is rigidly attached to the robot. During the obstacle detection phase, the computed normal flow field is inverse projected onto the horizontal plane, and analysed.

For pure translational motion, the obstacle detection analysis is very simple since all points on the ground plane should have the same flow vectors, whereas points lying above or below the ground plane, will have respectively larger or smaller flow values.

Contrasting with previous approaches [Carlsson and Eklundh, 1990, Enkelmann, 1990, Sandini and Tistarelli, 1990], the method we propose here, does not rely on the knowledge of the camera motion and, for pure translational motion, it is also independent of the camera intrinsic parameters. A geometric, intuitive explanation is given.

Different experiments were performed to test the various steps of the whole method. Finally, a real-time system was implemented on a robot, to detect obstacles laying on the ground plane. The system is fast and the performance robust. The results obtained are presented and discussed.

In the following sections, we describe the problem of motion analysis of a planar patch. Then, the affine model approximation for the motion field is introduced and we present the estimation procedure to recover the model parameters uniquely based on the normal flow. We then describe the inverse projection method, and show how to recover the needed image plane/ground plane transformation using the affine model parameters. Finally, a variety of tests are presented, and conclusions drawn.

4.2 Planar surfaces in motion

Many authors have addressed the problem of mission parameters estimation from global flow field data, as in [Nelson and Aloimonos, 1988], [Guissin and Ullman, 1989] and [Hummel and Sundareswaran, 1993]. Also, our first concern is about the characterization of the flow field perceived in the image plane. Particularly, in our approach, it is assumed that the robot is moving on a flat ground and that the camera is facing the ground plane, hence observing a planar surface in motion. With this assumption, it is possible to obtain a globally valid parameterization for the corresponding optical flow field. These parameters can be robustly estimated based on the normal flow field, as it is shown later in this chapter.

Let us assume a rigid body motion model for the robot, with general linear velocity $\mathbf{T} = [T_x \ T_y \ T_z]^T$ and general angular velocity $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$. Assuming a pinhole model for the camera (see Section 2.2), the projection of a 3D point with coordinates

$\{X,\ Y,\ Z\}$, into the image plane is given by :

$$\begin{cases} x &= f_x \frac{X}{Z} + c_x \\ \\ y &= f_y \frac{Y}{Z} + c_y \end{cases} \tag{4.1}$$

where, as usual, f_x, f_y denote the camera focal length expressed in pixels and c_x, c_y denote the image center coordinates. The projection coordinates x, y are given in pixels. The motion perceived in the image plane by the moving camera is then given by the well known equations [Subbarao and Waxman, 1986, Sundareswaran, 1991, Cipolla and Blake, 1992], [Cipolla et al., 1993] :

$$u(x,y) = f_x \left[\frac{\frac{x}{f_x} T_z - T_x}{Z(x,y)} + \omega_x \frac{xy}{f_x f_y} - \omega_y (1 + \frac{x^2}{f_x^2}) + \omega_z \frac{y}{f_y} \right] \tag{4.2}$$

$$v(x,y) = f_y \left[\frac{\frac{y}{f_y} T_z - T_y}{Z(x,y)} + \omega_x (1 + \frac{y^2}{f_y^2}) - \omega_y \frac{xy}{f_x f_y} - \omega_z \frac{x}{f_x} \right] \tag{4.3}$$

where u and v are the x and y components of the flow field.

Considering that the mobile robot is moving on a ground plane and that the camera is pointing at the ground floor, a global model for the motion field can be found. The plane equation can be given by :

$$Z(X,Y) = Z_0 + \gamma_x X + \gamma_y Y \tag{4.4}$$

where γ_x, γ_y are the surface slopes along the horizontal and vertical directions (slant and tilt) and Z_0 is the distance measured along the optical axis. By introducing the perspective equations, it is possible to describe the ground plane surface as a function of the image pixel coordinates, instead of the 3D coordinates :

$$Z(x,y) = \frac{Z_0}{1 - \gamma_x \frac{x}{f_x} - \gamma_y \frac{y}{f_y}} \tag{4.5}$$

Finally, using equation (4.5) together with equations (4.2), (4.3), we obtain the quadratic equations describing the flow of a planar surface in motion [Negahdaripour and Lee, 1992, Subbarao and Waxman, 1986].

$$u(x,y) = u_0 + u_x x + u_y y + u_{xy} xy + u_{xx} x^2 \tag{4.6}$$

$$v(x,y) = v_0 + v_x x + v_y y + v_{xy} xy + v_{yy} y^2 \tag{4.7}$$

which is a globally valid description of the optical flow, with the parameters being given by:

$$\begin{array}{llll} u_0 & = & -f_x\left[\frac{T_x}{Z_0}-\omega_y\right] & v_0 = f_y\left[-\frac{T_y}{Z_0}+\omega_x\right] \\ u_x & = & \frac{T_z+\gamma_x T_x}{Z_0} & v_x = \frac{f_y}{f_x}\left[\frac{T_y\gamma_x}{Z_0}-\omega_z\right] \\ u_y & = & \frac{f_x}{f_y}\left[\frac{T_x\gamma_y}{Z_0}+\omega_z\right] & v_y = \frac{T_z+\gamma_y T_y}{Z_0} \\ u_{xy} & = & v_{yy} & v_{xy} = \frac{-1}{f_x}(\frac{\gamma_x T_z}{Z_0}+\omega_y) \\ u_{xx} & = & v_{xy} & v_{yy} = \frac{1}{f_y}(-\frac{\gamma_y T_z}{Z_0}+\omega_x) \end{array} \tag{4.8}$$

At this point, a straightforward approach would consist in directly estimating the 8 parameters of the flow model (4.8). However, it has been shown analytically and experimentally in [Negahdaripour and Lee, 1992], that the estimates of the second-order parameters are often affected by noise which is higher, up to several orders of magnitude, than the first order parameters estimates, even in the case of perfect planar motion. If the angle of view is small, and the depth range restricted [Bergen et al., 1992, Koenderink and van Doorn, 1991], then the second-order parameters of the flow model can be discarded and the motion of the surface can be approximated as an affine transformation. The modeling error (particularly at the image periphery) is still usually smaller than the estimation error of the full second order model.

Then, to improve robustness, it is preferable to neglect the second order terms and approximate, instead, the motion field by an affine motion model :

$$\begin{array}{lll} u(x,y) & = & u_0+u_x x+u_y y \\ v(x,y) & = & v_0+v_x x+v_y y \end{array} \tag{4.9}$$

In the next section, we will present a robust method to estimate the affine model parameters uniquely based on the normal flow and, finally, we will show how to recover the image plane orientation relative to the ground plane.

4.2.1 Affine motion parameters estimation using the normal flow

We have established a model that captures the optical flow of a planar surface in motion with respect to a camera. The problem now is the estimation of the affine flow parameters. Particularly, we would like to constrain ourselves to the use of just normal flow information. The parameters will be used later on for a number of tasks such as in the obstacle detection mechanism, to recover the ground plane orientation, or to control the docking manoeuvre of a mobile robot, as described in Chapter 5.

Let us suppose that only the normal component of the flow field is available. This assumption is important since the normal flow is the single component which can be reliably estimated [Fermüller, 1993a, Fermüller, 1993b, Horn, 1986], due to the well known *aperture problem*, without having to assume further constraints on the motion field (such as smoothness) as it is usually done. The optical flow constraint equation, assuming image brightness constancy over time [Horn and Shunck, 1981] (see Section 3.3), is given by :

$$uI_x + vI_y = -I_t, \tag{4.10}$$

where I_x, I_y and I_t stand for the partial derivatives of the image with respect to x, y, and time. According to affine model equations (4.9), the first term of equation (4.10) can be written as :

$$uI_x + vI_y = v_o I_y + v_x I_y x + v_y I_y y + u_o I_x + u_x I_x x + u_y I_x y \tag{4.11}$$

Therefore, the temporal derivative over the planar patch can be expressed as a linear combination of the parameters to estimate:

$$\begin{bmatrix} I_y & xI_y & yI_y & I_x & xI_x & yI_x \end{bmatrix} \boldsymbol{\theta} = -I_t \tag{4.12}$$

where $\boldsymbol{\theta}$ is given by:

$$\boldsymbol{\theta} = [v_0 \;\; v_x \;\; v_y \;\; u_0 \;\; u_x \;\; u_y]^T \tag{4.13}$$

To estimate the affine parameters $\boldsymbol{\theta}$, it is sufficient to use just 6 measurements of spatial-temporal image derivatives or, equivalently 6 measurements of the normal flow. The least squares solution to this problem can be obtained by considering an over-determined

system of equations. Let n be the number of measurements available, and define :

$$\mathcal{D} = [-I_{t1} \ -I_{t2} \ \ldots \ -I_{t_n}]^T \tag{4.14}$$

now let

$$\mathcal{M} = \begin{bmatrix} I_{y1} & x_1 I_{y1} & y_1 I_{y1} & I_{x1} & x_1 I_{x1} & y_1 I_{x1} \\ I_{y2} & x_2 I_{y2} & y_2 I_{y2} & I_{x2} & x_2 I_{x2} & y_2 I_{x2} \\ & \vdots & & & \vdots & \\ I_{yn} & x_n I_{yn} & y_n I_{yn} & I_{xn} & x_n I_{xn} & y_n I_{xn} \end{bmatrix} \tag{4.15}$$

The least squares solution for the estimation problem is given by the pseudo-inverse solution :

$$\hat{\boldsymbol{\theta}} = (\mathcal{M}^T \mathcal{M})^{-1} \mathcal{M}^T \mathcal{D} \tag{4.16}$$

The direct application of the least squares estimation procedure is usually quite sensitive to outliers which usually lead to an ungraceful degradation of the estimates. To overcome this problem, we devised a recursive estimation procedure, aiming at eliminating the effect of outliers. The algorithm works as follows :

1. Choose randomly a set of data points, $\{I_x, \ I_y, \ I_t\}$, to get an initial estimate, $\boldsymbol{\theta_0}$. Set $k = 1$.

2. Choose randomly a new set of data points, $\{I_x, \ I_y, \ I_t\}$, such that the modeling error in equation (4.12), evaluated with the current parameter estimate, is small.

3. Estimate $\boldsymbol{\theta_k}$ based on the new data set. Set $k = k + 1$.

4. Return to step (2) until $\boldsymbol{\theta}$ remains unchanged or k exceeds a given number of iterations.

The rationale behind this algorithm is that by getting an approximate initial estimate, one can successfully improve this estimate by selecting the data that are most coherent with the model. In this way, outliers will be discarded in the point selection mechanism and the estimate improved, in subsequent steps.

A wide variety of tests were performed in order to evaluate the robustness of the estimation procedure. Figure 4.1 shows an image sequence acquired by a camera moving towards a slanted poster.

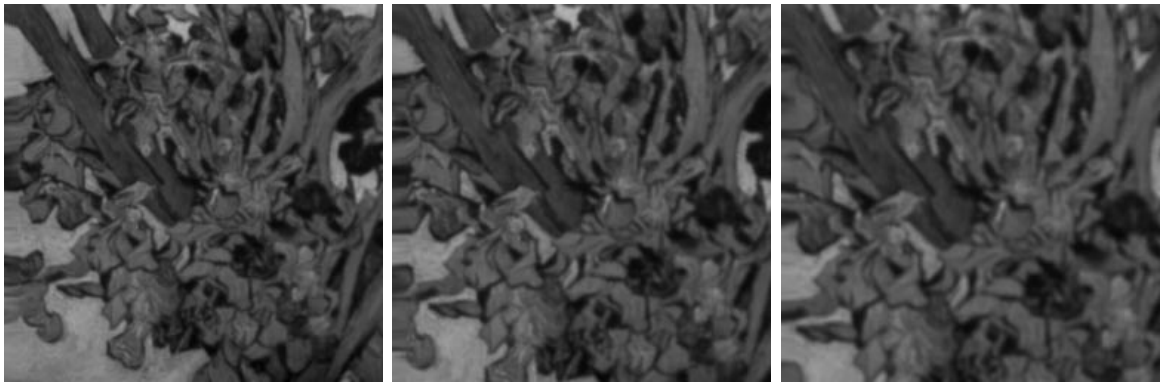

Figure 4.1: An image sequence used for the estimation tests, acquired by moving a camera towards a slanted poster.

Figure 4.2 shows an example of the optical flow field computed based on this image sequence. These data were used to estimate the parameters of the affine motion model as described previously. We have used these parameters to generate a "reconstructed" flow field, which is shown for comparison in Figure 4.2. As shown, the estimation procedure

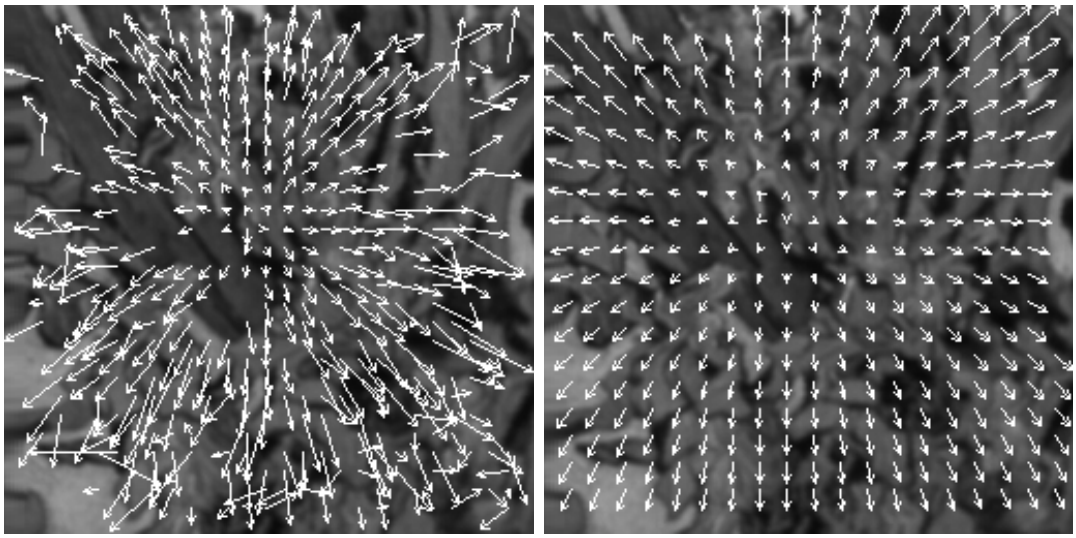

Figure 4.2: The left image shows the optical flow estimated for the poster sequence. The right image shows the reconstructed flow field based on the estimated affine motion model parameters .

succeeds in discarding the outliers and the "reconstructed" optical flow is, at least from a qualitative point of view, consistent with the input data. Notice, for instance, the location of the Focus of Expansion in both cases.

4.2.2 Plane coefficients estimation - the intrinsic parameters

We have already presented a procedure to estimate the set of parameters which describe the affine model of the planar surface motion field. Once $\boldsymbol{\theta}$ has been estimated, one can determine γ_x and γ_y by referring to the equation set (4.8). However, these parameters can only be estimated up to a scale factor, which is the focal length expressed in pixel coordinates :

$$\begin{cases} \frac{\gamma_x}{f_x} = -\frac{v_x}{v_0} \\ \\ \frac{\gamma_y}{f_y} = \begin{cases} -\frac{u_y}{u_0}, & \text{if } u_0 \neq 0 \\ \\ \frac{u_x - v_y}{v_0}, & \text{otherwise} \end{cases} \end{cases} \tag{4.17}$$

Consequently, if the camera intrinsic parameters are known, the $\gamma_{x,y}$ coefficients can be obtained directly, thus solving the problem. However, even in the absence of the camera parameters, f_x and f_y, it is still possible to proceed with the method, as we shall now show. If the mobile robot motion is assumed to be purely translational, the affine model parameters can be simplified to :

$$\begin{aligned} u_0 &= -\frac{f_x T_x}{Z_0} & v_0 &= -\frac{f_y T_y}{Z_0} \\ u_x &= \frac{T_z}{Z_0} + \frac{\gamma_x}{f_x}\frac{fxT_x}{Z_0} & v_x &= \frac{f_y T_y}{Z_0}\frac{\gamma_x}{f_x} \\ u_y &= \frac{f_x T_x}{Z_0}\frac{\gamma_y}{f_y} & v_y &= \frac{T_z}{Z_0} + \frac{\gamma_y}{f_y}\frac{f_y T_y}{Z_0} \end{aligned} \tag{4.18}$$

These equations have been rewritten to emphasize the ambiguity between $\{f_x,\ f_y\}$ and $\{\gamma_x,\ \gamma_y,\ T_x,\ T_y\}$. In all the equations above, if we scale up f_x or f_y by a given factor, ξ, divide the corresponding T_x or T_y by ξ and multiply the corresponding $\{\gamma_x,\ \gamma_y\}$ by ξ again, then all the flow parameters remain unchanged. This amounts to saying that, if a camera with different intrinsic parameters is used, it is possible to find a different orientation and velocity (suitably scaling $\gamma_x,\ \gamma_y,\ T_x,\ T_y$) such that the camera observes exactly the same flow field.

Having this ambiguity in mind and as we are not interested in the absolute camera speed, nor on the absolute camera orientation, we can suppose that a *virtual* camera with

unitary f_x and f_y, is being used. The simplified parameterization then follows :

$$\begin{aligned} u_0 &= -\tfrac{T_x}{Z_0} & v_0 &= -\tfrac{T_y}{Z_0} \\ u_x &= \tfrac{T_z}{Z_0} + \gamma_x \tfrac{T_x}{Z_0} & v_x &= \tfrac{T_y}{Z_0}\gamma_x \\ u_y &= \tfrac{T_x}{Z_0}\gamma_y & v_y &= \tfrac{T_z}{Z_0} + \gamma_y \tfrac{T_y}{Z_0} \end{aligned} \tag{4.19}$$

where one should keep in mind that T_x, T_y, γ_x and γ_y do no longer convey information neither on the absolute orientation of the ground plane, nor on the absolute camera speed. They correspond to the speed and orientation, with respect to the ground floor, that a camera with unitary intrinsic parameters, f_x and f_y, should have in order to perceive the same flow field. Finally, γ_x and γ_y are simply determined by using equation (4.17) with unitary f_x, f_y.

In order to evaluate the performance of the estimation procedure to recover the plane orientation, we have generated a synthetic flow pattern corresponding to a robot moving forwards with a speed of 25cm/s equipped with a camera pointing down at 45^o :

$$\gamma_x = 0, \quad \gamma_y = 1$$

The camera intrinsic parameters, we used, are those of a standard CCD (6.4mm by 4.8mm) sensor with a 4.8 mm lens and 256×256 image resolution :

$$f_x = 192, \quad f_y = 256, \quad c_x = 128, \quad c_y = 128$$

The flow vectors were computed according to these camera settings and motion and corrupted with zero-mean Gaussian noise with a given variance (the noise is added independently to each component). Figure 4.3 shows the synthetic flow field in the absence of noise.

Table 4.1 shows the slant and tilt angles estimate obtained using the synthetic flow data with increasing noise levels. These results show that the estimates of the plane orientation are fairly robust and accurate even in the presence of significant levels of noise.

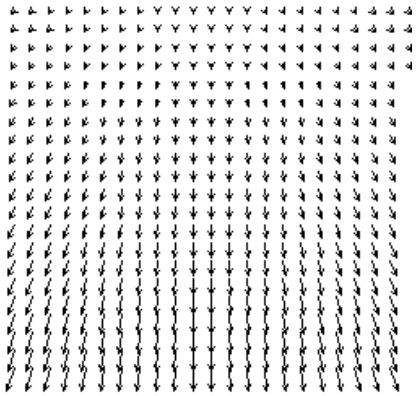

Figure 4.3: Flow field generated for the simulation study, which corresponds to a robot moving forward on a flat ground, with a camera pointing down.

Noise σ	# points	slant (deg.)	tilt (deg.)
0.0	200	0.0	-45.0
1.0	200	-0.83	-46.97
1.5	200	-0.30	-42.79
2.0	200	-0.20	-46.63

Table 4.1: Tilt and Slant estimates for different values of the noise standard deviation. The results show the robustness of the estimation procedure. The flow fields with different noise levels are shown in Figure 4.5.

4.3 Inverse Perspective Flow Transformation

The method we propose for ground plane obstacle detection, is based on an inverse perspective mapping of the flow field. The main idea is that by re-projecting the flow onto the horizontal plane, the analysis is much simplified. We will show that the obstacle detection algorithm is very simple for pure translational motion and can also cope with rotation by fitting simple models to the transformed flow.

Similarly to what has been proposed in [Zielke et al., 1990, Mallot et al., 1991], the rationale of this method is that if it is possible to inverse project the flow field perceived on the image plane, (π_C), onto the horizontal plane, (π_H), then the camera translation becomes parallel to the ground floor and the rotation is solely around the vertical axis, which greatly simplifies the flow pattern as illustrated in Figure 4.4.

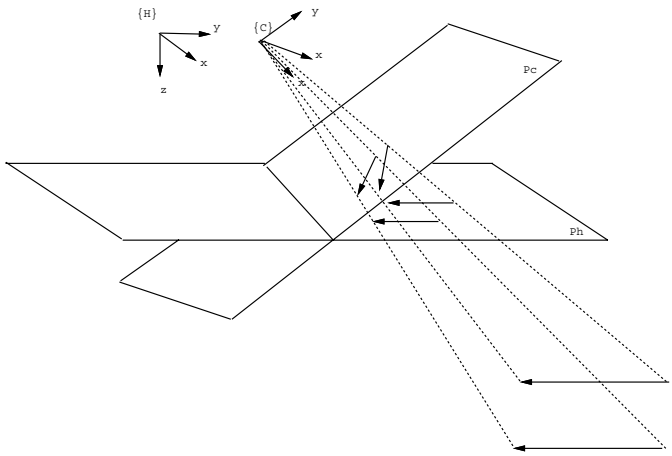

Figure 4.4: Inverse perspective mapping. The coordinate systems (C) and (H) share the same origin even though in the picture they have been drawn separately for the sake of clarity. While on (C) the motion of the ground floor is perceived as a complex vectorial pattern, in (H) all the vectors have equal length and orientation under translational motion.

In all the subsequent analysis, we will include the camera intrinsic parameters, which could be considered unitary and discarded in the case of translational motion according

to the discussion in Section 4.2.2. Let

$$C = \{X_c,\ Y_c,\ Z_c\ \}$$
$$H = \{X_h,\ Y_h,\ Z_h\ \} \tag{4.20}$$

be the coordinate frames associated to the camera plane and the horizontal plane. As both frames share a common origin, the coordinate transformation relating both systems is just a rotation term, ${}^{H}\mathbf{R}_{C}$:

$${}^{H}\mathbf{P} = {}^{H}\mathbf{R}_{C}\ {}^{C}\mathbf{P} \tag{4.21}$$

where ${}^{C}\mathbf{P}$ is a point in the 3D space expressed in the camera coordinates, and ${}^{H}\mathbf{R}_{C}$ results from rotating a tilt angle, ψ, around the camera x axis and a pan angle, ϕ, around the camera y axis (which corresponds to a camera pointing down in front of a mobile robot). The rotation matrix will have the following structure however, in most practical systems the pan angle, ψ, is small and could even be neglected) :

$${}^{H}\mathbf{R}_{C} = \begin{bmatrix} \cos\phi & -\sin\phi\sin\psi & -\sin\phi\cos\psi \\ 0 & \cos\psi & -\sin\psi \\ \sin\phi & \cos\phi\sin\psi & \cos\phi\cos\psi \end{bmatrix} \tag{4.22}$$

Let the perspective projection of a point in a plane β, be defined :

$${}^{\beta}\mathbf{P}' = \mathcal{P}_{\beta}({}^{\beta}\mathbf{P}) \tag{4.23}$$

$$\begin{bmatrix} s\ x'_{\beta} \\ s\ y'_{\beta} \\ s \end{bmatrix} = \begin{bmatrix} f_x X_{\beta} \\ f_y Y_{\beta} \\ Z_{\beta} \end{bmatrix} \tag{4.24}$$

where $\mathcal{P}_{\beta}$ denotes the projection operator, and x', y' are image points expressed in pixel coordinates. The set of points in the 3D space that project on a given image pixel (x'_c, y'_c) is given by :

$${}^{C}\tilde{\mathbf{P}} = [\lambda\frac{x'_c}{f_x} \quad \lambda\frac{y'_c}{f_y} \quad \lambda]^T \tag{4.25}$$

which describes a beam passing through the projection center and the projection point in the image plane. As any other 3D point expressed in the camera coordinate system, $\tilde{P}$ can be expressed in the frame attached to the horizontal plane :

$$^{H}\tilde{\mathbf{P}} = {}^{H}\mathbf{R}_{C}\, {}^{C}\tilde{\mathbf{P}} \tag{4.26}$$

Finally, this point can be projected into the horizontal plane, π_H, combining equations (4.22) to (4.26) :

$$\begin{bmatrix} x'_H \\ \\ y'_H \end{bmatrix} = {}^{H}\mathcal{P}_{C}(x'_c, y'_c) \tag{4.27}$$

where ${}^{H}\mathcal{P}_{C}(x'_c, y'_c)$ denotes the operator projecting from the image plane to the horizontal plane. This plane-to-plane projective transformation [Mundy and Zisserman, 1992] can be rewritten as :

$$\begin{bmatrix} s\, x'_H \\ s\, y'_H \\ s \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi\sin\psi & -sin\phi\cos\psi \\ 0 & \cos\psi & -\sin\psi \\ \sin\phi & \cos\phi\sin\psi & \cos\phi\cos\psi \end{bmatrix} \begin{bmatrix} x'_c/f_x \\ y'_c/f_y \\ 1 \end{bmatrix} \tag{4.28}$$

This equation determines how to inverse project, onto the horizontal plane, a point projected in the image plane pixel (x'_c, y'_c). To obtain the inverse projection of a flow vector, we have to calculate the time derivatives of (x'_H, y'_H) :

$$\begin{bmatrix} u'_H(x'_c, y'_c) \\ \\ v'_H(x'_c, y'_c) \end{bmatrix} = \frac{\partial {}^{H}\mathcal{P}_{C}(x'_c, y'_c)}{\partial x'_c}\, u + \frac{\partial {}^{H}\mathcal{P}_{C}(x'_c, y'_c)}{\partial y'_c}\, v \tag{4.29}$$

which can be written in homogeneous coordinates as :

$$\begin{bmatrix} s\,u'_H(x'_c, y'_c) \\ s\,v'_H(x'_c, y'_c) \\ s \end{bmatrix} = \begin{bmatrix} f_x\,[(\cos\psi + \sin\psi\,\frac{y'_c}{f_y})\frac{u}{f_x} - \sin\psi\,\frac{x'_c}{f_x}\frac{v}{f_y}] \\ f_y\,[\sin\phi(\sin\psi - \cos\psi\,\frac{y'_c}{f_y})\frac{u}{f_x} + (\cos\psi\sin\phi\,\frac{x'_c}{f_x} + \cos\phi)\frac{v}{f_y}] \\ (\sin\phi\,\frac{x'_c}{f_x} + \cos\phi\sin\psi\,\frac{y'_c}{f_y} + \cos\phi\cos\psi)^2 \end{bmatrix} \quad (4.30)$$

Remains to be considered the estimation of ψ and ϕ using the affine model parameters that were estimated initially.

4.3.1 Recovering the Slant and Tilt parameters

During an initialization phase, we must determine the rotation matrix, preferably without explicitly calibrating the system. The process simply consists in determining γ_x, γ_y as explained in the previous sections and assuming that both f_x and f_y are unitary (for the case of pure translation).

Since any point in the camera coordinate system can be expressed, according to equation (4.21), in the horizontal plane coordinate system, we can examine the Z component of such term:

$$Z_H = \sin\phi X + \cos\phi\sin\psi Y + \cos\phi\cos\psi Z \quad (4.31)$$

However, in terms of the camera coordinates, Z is a function of X and Y according to the ground plane constraint, given in equation (4.4). Combining these two equations, and knowing that, in the coordinate system of the horizontal plane, all the points in the ground floor have a constant depth, Z_H, we get:

$$\begin{aligned} \psi &= -\arctan\gamma_y \\ \phi &= -\arctan(\gamma_x\cos\psi) \\ Z_H &= \cos\psi\cos\phi\,Z_0 \end{aligned} \quad (4.32)$$

To summarize, once γ_x and γ_y have been estimated according to the procedure described in Section 4.2.1, the inverse perspective transformation can be applied, to inverse project the optical flow onto the horizontal plane, where the analysis is simplified. The

values of ψ and ϕ are estimated only once during an initialization phase and remain constant provided that the camera is in a fixed position relative to the robot.

4.3.2 Obstacle Detection

In this section we will analyse the problems that have to be addressed after inverse projecting the optical flow, in order to detect the presence of obstacles.

Using the horizontal plane coordinate frame, the linear velocity is constrained to be parallel to the "*new image*" plane (hence $T_z = 0$), the angular velocity component is solely around the vertical axis and the distance to the ground floor is constant. Therefore, the optical flow equations are given by :

$$u_h(x_H, y_H) = \left(-\frac{T_x}{Z_H} + \omega_z y_H\right) \tag{4.33}$$

$$v_h(x_H, y_H) = \left(-\frac{T_y}{Z_H} - \omega_z x_H\right) \tag{4.34}$$

In many situations, it is reasonable to assume that the rotation component is neglectable when compared to the translational component. On other cases, we can avoid computing the flow during fast rotations, which would correspond to a saccadic suppression mechanism observed in many biological vision systems. For pure translational motion, the transformed flow vectors are constant for every point on the ground plane and a simple test can be performed to check if there are any obstacles above or below the ground plane. The detection mechanism simply relies on the fact that the optical flow should be globally constant no matter what the motion direction and speed might be. Having just computed the normal flow component, at this point the normal flow vectors can be projected onto the direction of motion, which is constant all over the image.

$$u_h(x, y) = -\frac{T_x}{Z_H}$$

$$v_h(x, y) = -\frac{T_y}{Z_H} \tag{4.35}$$

In the case of general motion, the main difficulty consists in separating the rotation component from the purely translational component [Negahdaripour and Lee, 1992, Sundareswaran, 1992, Hummel and Sundareswaran, 1993]. In our simplified reference frame, we can simply apply an estimation procedure to the inverse projected flow in order to recover the translational (constant all over the image) and rotational components (which depend on the x or y coordinates). The same kind of estimation techniques that were used to estimate the affine flow parameters can again be used to separate the rotation and translation components of motion. Once this has been done, the translational component can be checked as in the case of purely translational motion.

4.4 Results

In this section we present some results obtained using the proposed method. Initially, we will show some results using the synthetic data and finally some results obtained with a real robotic application.

For the synthetic tests, we have used the synthetic flow field generated according to the procedure described in Section 4.2.2. The synthetic flow fields were corrupted with increasing levels of noise to evaluate the performance of the affine flow parameters estimation. The inverse projection procedure was performed assuming that the intrinsic parameters are unknown. Figure 4.5 shows these experiments. On the left column we have the synthetic flow field after being degraded with noise of different intensities. The right column shows the corresponding inverse projected flow fields.

From the figure, we can see that even for severely corrupted data, the flow becomes relatively constant after the inverse projection. The degradation is more noticeable in the image areas corresponding to the far part of the visual field. In these areas, the flow amplitude is rather small and the signal to noise ratio is poor. The method seems to be considerably more accurate at short range, instead.

The system has been tested in real time on a mobile robot. The mission consisted in moving around in a room and stopping whenever an obstacle was detected. The experimental setup is composed of a camera with a 8mm lens, installed on a TRC Labmate mobile platform. The camera was placed in the front part of the robot facing the ground

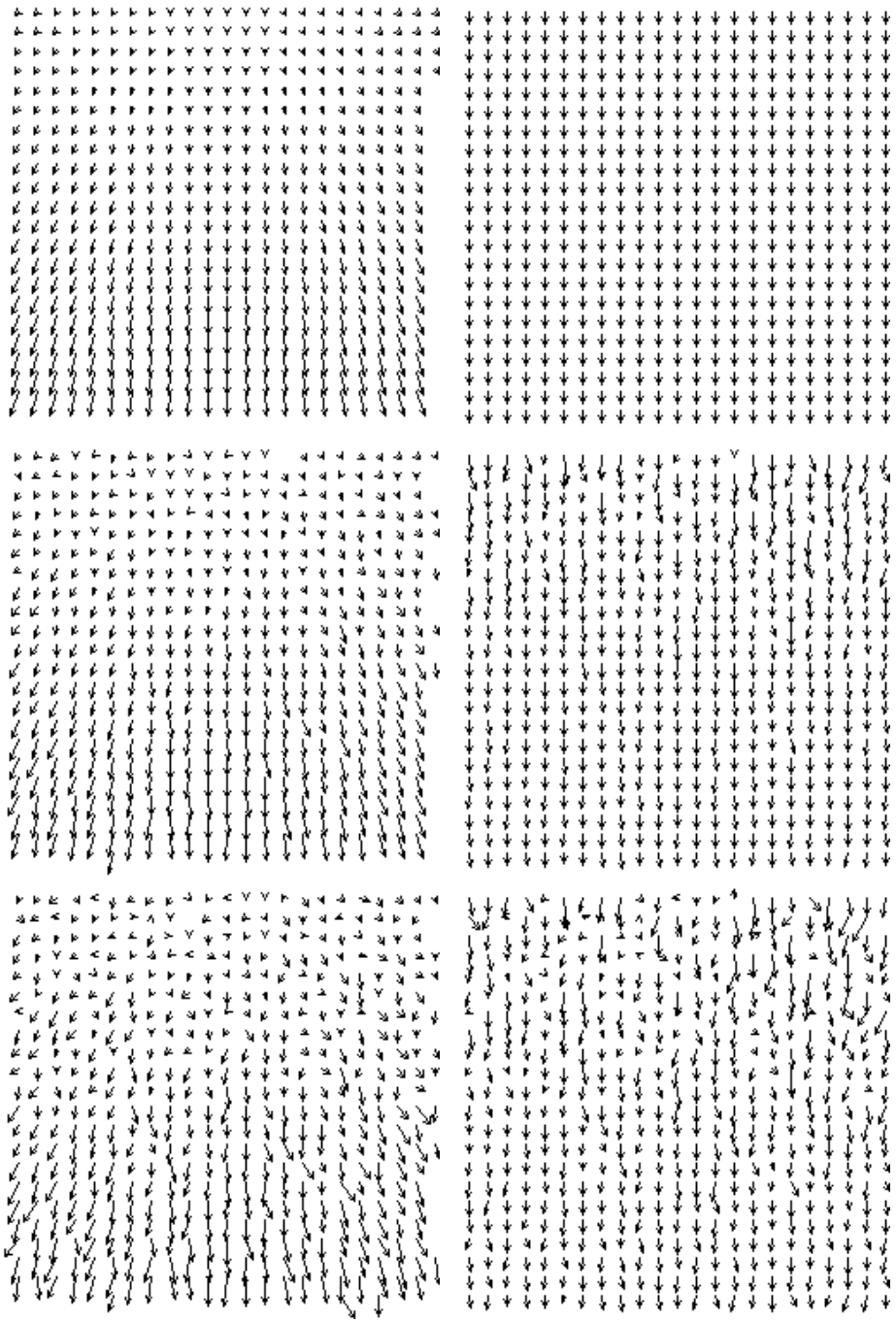

Figure 4.5: The left column shows the synthetic flow fields corrupted by noise with $0.0, 1.0$ and 2.0 pixels/frame of standard deviation. On the right column we show the flow inverse projected onto the horizontal plane which becomes constant, and can be used for obstacle detection.

plane with an angle of about 65 degrees.

In all the experiments performed, the robot speed was set to 10cm/s. The images are acquired with a resolution of 128×128 pixels and a central window of 80×80 pixels is used for the normal flow computation and the obstacle detection process. This resolution represents a tradeoff between the necessary detail for the flow computation and analysis, and the computing speed. At the current level of implementation, the system is running at a frequency of about 1 H_z on an VDS Eidobrain image processing workstation.

Figure 4.6 shows an example of the normal flow field measured while the robot is moving over the ground plane, in the absence of obstacles. Notice how the flow is along the image gradient and how its magnitude increases from the top to the bottom. These normal flow vectors are then used in the initialization stage to estimate the ground plane parameters. The same figure also shows the result of inverse projecting the ground plane flow field. The result is an approximately constant flow field over the whole image.

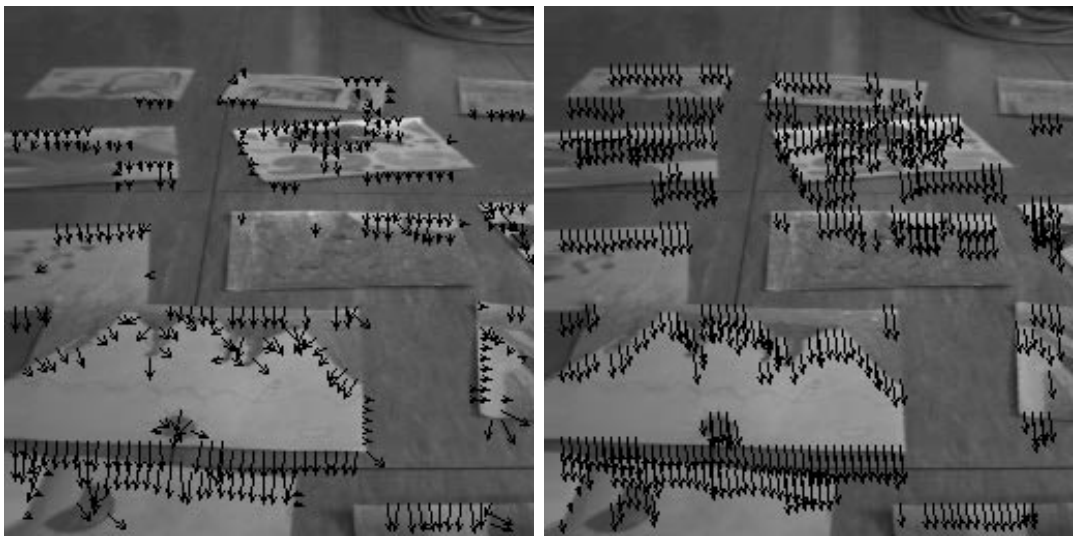

Figure 4.6: The left image shows a sample of the ground plane normal flow field measured during the robot motion. The right image shows the flow field that results from the inverse projection.

Several tests were performed using different obstacles placed on the ground floor. An example is shown in Figure 4.7. On the left we have the normal flow observed during the trajectory with an obstacle in the visual field. Using the flow information alone, it is difficult to detect the obstacle. On the right side of Figure 4.7, we show the result of the

inverse projection mechanism, where the obstacle flow becomes clearly larger than the ground flow.

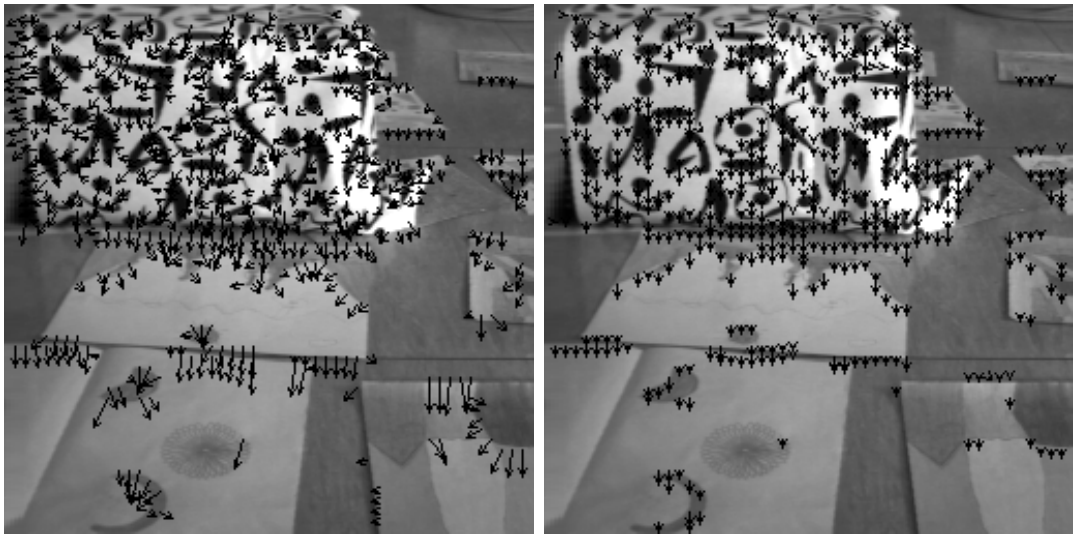

Figure 4.7: The left image shows a sample of the normal flow field measured during the robot motion. The right image shows the resulting inverse projected flow, where the obstacle can be easily detected.

The result of the detection process is shown in Figure 4.8. The white areas correspond to obstacle points, while the dark areas represent free space.

This result shows that the obstacle has been detected, particularly the object top regions. This is due to the sensitivity constraints imposed by the image resolution. It should be noted that the object contours are reasonably defined and that an acceptable segmentation could be achieved. However, a precise reconstruction of the obstacle position is not the main goal of the methodology we propose. The goal, instead, is to provide a simple, fast and robust reflex-type behaviour for obstacle detection. These characteristics were achieved in all the tests performed where the system behaved robustly detecting various obstacles and safely stopping the robot. Several simple strategies can be used in order to circumvent the obstacles detected. The robot can, for example, rotate a given angle away from the image side where the object is located, or we can determine the amount of rotation needed to remove the obstacle from the robot visual field.

Figure 4.8: Points detected as obstacles above the ground plane.

4.5 Conclusions

We have described a method for fast obstacle detection for mobile robots. The basic assumption is that the robot is moving on a ground floor and any object not laying on this plane is considered an obstacle. The method is based on the inverse projection of the flow vector field onto the ground plane, where the analysis is much simplified as, for the case of pure translational motion, the flow vectors become constant all over the image with the obstacles having a larger flow than every other point laying on the pavement. The detection is then much simplified.

The flow induced by the motion of the ground plane, is described by the affine model. The model parameter estimation procedure relies exclusively on the first order space and time derivatives, or equivalently, on the normal flow. Based on the affine model parameters, the tilt and slant angles of the image plane with respect to the ground are estimated and used for the inverse projection operation. No explicit calibration of the camera intrinsic or extrinsic parameters is needed.

It is also shown that in the absence of rotational motion, the method is independent of the camera intrinsic parameters. It is not necessary to know the robot motion, as the detection strategy is based on the constancy of the flow vectors. Several tests were presented to illustrate the robustness of the approach. A running implementation is available using a mobile platform. Results of the real time experiments have been presented.

Therefore, we have proposed a strategy for obstacle detection to overcome some of the limitations of *Robee*. It should be noticed that both approaches share some common points. They use the same input data, first order spatial and temporal derivatives of the images, and different specialized parts of the visual field, peripheral and central, according to the purpose.

Chapter 5

Visual Behaviours for Docking

There is often the need for a mobile robot to approach a specific point in the environment in a controlled way, for instance to recharge batteries, logging in to a computing center, grasp objects, etc. This docking procedure is then an important functionality of a mobile robot.

This chapter describes visual-based behaviours for docking operations in mobile robotics. We consider two different situations : in the *ego-docking*, each robot is equipped with a camera and the egomotion controlled when docking to a surface, whereas in the *eco-docking*, the camera and all the necessary computational resources are placed in a single external docking station, which may serve several robots. In both situations, the goal consists in controlling both the orientation, aligning the camera optical axis with the normal to the surface, and the approaching speed (slowing down during the manoeuvre).

These goals are accomplished without any effort to perform 3D reconstruction of the environment or any need to calibrate the setup, in contrast with traditional approaches. Instead, we use image measurements directly to close the control loop of the mobile robot.

In the approach we propose, the robot motion is directly driven by the normal flow field, similarly to the procedure in the previous chapter. The docking system is operating in real time and the performance is robust both in the *ego-docking* and *eco-docking* paradigms. Experiments are described.

5.1 Introduction

The intimate relationship between perception and action has been discussed and presented in different forms in the last years. From the seminal paper of Ruzena Bajcsy describing the peculiarity of Active Perception [Bajcsy, 1988] through the papers on Active and Animate vision [Ballard, 1991, Aloimonos et al., 1988] and, more recently, to the concept of Purposive Vision [Aloimonos, 1990]. Along its evolution research about the perception/action relationship has suggested, at least, three major advances. The first is linked to the concept of "exploratory actions" and to the fact that an active observer can acquire more information about the world by controlling his own position and kinematic parameters (including optics). The second is the observation that action may help in simplifying perceptual processes some of which are, in general, ill-posed. The third is the observation that action is tightly linked to purpose and that purposive actions provide a natural and powerful constraint to perceptual processing allowing, among other things, the use of qualitative perceptual information.

From the control point of view the evolution has gone from an "exploratory approach", which, in some sense is linked to the problem of motor planning (e.g. move around the object to acquire more information or move the finger around the rim of a cup to acquire its shape) through an "utilitarian" phase where action is driven by the need to improve the perceptual process, to arrive, more recently, to the concept of visual servoing and visual behaviours where action is eliciting and simplifying the perceptual processes which, in turn, drive the action itself (the "vision during action" approach [Fermüller, 1993b, Sandini et al., 1993a, Santos-Victor et al., 1994a]). In this case the control loop becomes tighter and, if direct visual measures are used, motor control is directly driven by iconic information (I.e. data which are directly computed from the images). The simplest instance of this kind of sensory/motor coordination is represented by visual reflexes where the action cannot be purposively controlled but is a direct consequence of a sensory input. The experiment presented here, in spite of the fact that it is based on direct, iconic, visual information, is based on the powerful assumption that we are able to define the purpose of the motor action. We define such a motor action, solely driven by direct visual information and purpose, as a visual behaviour. In this sense, the observation that the goal of action

is to perceive has evolved to a different one: the goal of perception is to act.

In particular, the visual behaviours we address here, are docking strategies for indoor mobile robots. The robot desired behaviour consists in approaching a surface, along its normal, with controlled forward speed and then stop. Such behaviour can be used by a mobile vehicle to dock to a computing center, recharge its batteries in a battery charging station, approach a work cell to manipulate various objects or, with minor changes, follow another robot at fixed distance.

We will consider two distinct situations for the docking problem. In the first situation, that we call *ego-docking*, the camera is mounted on board of the vehicle, and the robot egomotion is controlled during a docking manoeuvre to a particular surface in the environment. The second scenario, that we call *eco-docking*[1], the camera and computational resources are installed on a single external docking station with the ability to serve multiple robots. Both scenarios are depicted in Figure 5.1.

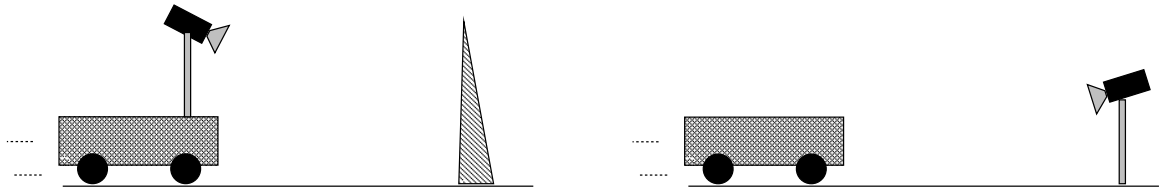

Figure 5.1: The left diagram shows the *ego-docking* where a robot, equipped with a camera and computing resources, docks to a specified surface. Instead, in the *eco-docking*, shown on the right, the camera is attached to a single docking station which may serve multiple robots.

From the perceptual point of view, both situations are quite similar since the important issue is the relative motion between the camera and the docking surface. However, a careful analysis reveals some formal differences between both cases. In the *ego-docking*, the camera position with respect to the robot is fixed, whereas in the *eco-docking* it is changing continuously, thus posing new problems for the visuo-motor control loop. However, we show that, by proper formulation of the problem, exactly the same control architecture can be used in both cases.

The behaviours and framework we describe can have multiple important applications

[1]From *oikos*, the Greek word for environment or external world.

in mobile robotics. In the *ego-docking* case, each robot can be controlled to dock to any particular point in the environment, thus offering large flexibility. In the *eco-docking* concept, a single docking station with a camera and the computing resources can be used to serve a large number of robots. The robots can be commanded to approach the docking station using odometric information alone (hence with limited precision) and once in the neighbourhood of the docking station, the control system would take over and perform the manoeuvre.

Similarly to the approaches for the navigation and obstacle detection behaviours, described in Chapters 3 and 4, we assume that only a partial description of the optical flow field is available (See Section 3.3 for a discussion on the optical flow estimation and constraints).

As mentioned before, the normal flow alone conveys sufficient information to accomplish many perception problems and can be estimated robustly and fast. The solution proposed here relies, again, exclusively on the use of the normal flow information to drive the robot motion, without imposing any type of smoothness constraints on the flow field.

Section 5.2 is devoted to the problem sensory-motor coordination, where the robot (motor) and camera (sensor) coordinate frames are related in both the *ego-docking* and *eco-docking* situations. This analysis establishes the link between the measured visual parameters and the appropriate motor actions in closed loop.

Section 5.3 shows that it can be assumed that the camera is observing a planar surface in motion and, therefore, the analysis used in Section 4.2 still holds and the normal flow is used to extract the affine motion parameters.

In Section 5.4, we use the sensory-motor coordination to relate the affine motion parameters expressed in the camera coordinate frame (sensor frame) and the navigation commands expressed in the robot coordinate system (motor frame). A closed loop strategy is proposed to control the robot heading and speed, directly integrating the visual measurements in the motion controller.

In Section 5.5, we present a real-time implementation of the behaviours described, showing a robust performance in various experiments and, finally, in the last section, we draw some conclusions and establish further directions of work.

It should again be stressed that the use of visual measurements directly in the mo-

tion controller leads to improved robustness as the system is continuously monitoring its performance [Espiau et al., 1992, Santos-Victor et al., 1993, Santos-Victor et al., 1994a, Sundareswaran et al., 1994], there is no need to calibrate the camera and no reconstruction of the environment is performed.

5.2 Sensory-motor coordination

In this section we will address the problem of sensory-motor coordination. Both in the *ego-docking* or *eco-docking* problems, the visual information captured by the camera is being used to drive the motor control loops of the mobile platform. Therefore, one could refer to the *sensor frame* where the visual measurements are computed and the *motor frame*, where the commands to the robot are defined. In order to design the closed-loop controllers for the robot, it is necessary to determine the coordinate transformation between both frames, which we designate by the *sensory-motor* coordination problem.

All the experiments were carried out using a TRC Labmate mobile platform with the motion degrees of freedom described in Figure 5.2. The robot motion is constrained to a

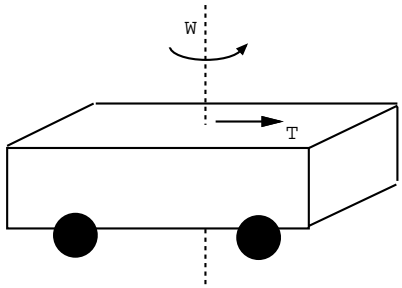

Figure 5.2: Mobile platform motion degrees of freedom.

forward speed T and a rotation speed ω around the vertical axis. In the following sections we will analyse the sensory-motor coordination for both the *ego-docking* and *eco-docking* problems.

5.2.1 Ego-docking

For the *ego-docking* behaviour, each robot is equipped with an on-board camera and all the computational resources needed for the perception process. Whenever this behaviour is launched, the mobile robot is supposed to align itself perpendicularly to a specified docking surface, controlling the forward speed, until it stops.

We assume that the camera coordinate frame is translated with respect to the origin of the motor frame and rotated by a tilt angle, ψ, around the x axis, and a pan angle, ϕ, around the y axis. It is further assumed, for simplification, that the translation is solely along the y (forward) and z (upward) directions, as shown in Figure 5.3.

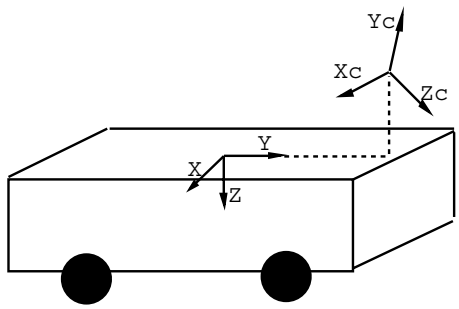

Figure 5.3: Sensor and motor coordinate frames.

In many situations the camera points in the forward direction, coincident with the motion direction for pure translation and, therefore, the pan angle is approximately zero. However, we address the more general situation as it allows the use of an active pan/tilt camera mount which may be advantageous for more complex visual behaviours. The constraint regarding the translation between sensor and motor frames can be easily achieved in practice without the need for any specific calibration procedure, and is by no means critical.

Let {R} and {C} be respectively, the robot and camera coordinate frames or, in other words, the motor and sensor coordinate frames. The motion of {C} expressed in the sensor coordinate frame [Yioshikawa, 1990] is given as a function of the motion of {R} according to :

$$\begin{aligned} \mathbf{T}_C &= {}^C\mathbf{R}_R(\mathbf{T}_R + \boldsymbol{\omega_R} \times {}^R\mathbf{P}_{OC}) \\ \boldsymbol{\omega_C} &= {}^C\mathbf{R}_R\, \boldsymbol{\omega_R}\,, \end{aligned} \tag{5.1}$$

where, as usual, $\mathbf{T}$ and $\boldsymbol{\omega}$ stand for translation and angular velocities of the frame identified by the subindex. The term ${}^{R}\mathbf{P}_{OC}$ is the position vector of the origin of the camera coordinate frame expressed in the robot coordinate system, assumed to have zero x component :

$$\mathbf{P}_{OC} = [0 \quad dy \quad dz]^T . \tag{5.2}$$

As explained, the rotation matrix relating both frames, is decomposed by a pan (ϕ) and tilt (ψ) contributions, and can be written as :

$${}^{C}\mathbf{R}_{R} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ -\sin\phi\sin\psi & \cos\psi & \cos\phi\sin\psi \\ -\sin\phi\cos\psi & -\sin\psi & \cos\phi\cos\psi \end{bmatrix} . \tag{5.3}$$

The motion degrees of freedom of the mobile platform are constrained, according to Figure 5.2, to a pure rotation around the Z axis and a pure translation along the forward direction :

$$\begin{aligned} \mathbf{T}_R &= [0 \quad T_{ry} \quad 0]^T \\ \boldsymbol{\omega_R} &= [0 \quad 0 \quad \omega_{rz}]^T . \end{aligned} \tag{5.4}$$

Finally, we can express the sensor egomotion as a function of the motor frame motion, by combining equations (5.1) to (5.4). It yields :

$$\begin{aligned} \mathbf{T}_C &= \begin{bmatrix} -\omega_{rz} dy \cos\phi \\ T_{ry}\cos\psi + \omega_{rz} d_y \sin\phi\sin\psi \\ -T_{ry}\sin\psi + \omega_{rz} d_y \sin\phi\cos\psi \end{bmatrix} , \\ \boldsymbol{\omega_C} &= \begin{bmatrix} \omega_{rz}\sin\phi \\ \omega_{rz}\sin\psi\cos\phi \\ \omega_{rz}\cos\psi\cos\phi \end{bmatrix} . \end{aligned} \tag{5.5}$$

In many situations, the camera does not have any mechanical degrees of freedom and, therefore, the rotation matrix is constant over time. Assuming, as we mentioned before, that in this case the camera is pointing in forward direction, ϕ is set to zero and the camera motion equations simplify to :

$$\begin{aligned} \mathbf{T}_C &= [-\omega_{rz} dy \quad T_{ry}\cos\psi \quad -T_{ry}\sin\psi]^T , \\ \boldsymbol{\omega_C} &= [0 \quad \omega_{rz}\sin\psi \quad \omega_{rz}\cos\psi]^T . \end{aligned} \tag{5.6}$$

An interesting extension to the *ego-docking* approach, consists in having an active camera mount, and control the docking point, just by fixation. That is to say that the robot would dock to the world point being fixated, without the need to specify the docking point in terms of odometry or any other metric system. This behaviour could be achieved by combining a fixation behaviour for gaze control (the gaze control problem is addressed in Chapter 6) and the docking behaviour herein described.

5.2.2 Eco-docking

In the *eco-docking* scenario, we have a single docking station carrying a camera (eventually with an active mount) and all the necessary computational resources. This station may serve several robots, which in turn do not need particular visual capabilities nor do require any specific computing power. Typically, a robot could be commanded to go to the docking station based on odometric information alone, which is known to be of limited precision and, once in the vicinity of the docking station, the local docking behaviour would take over and conduct the manoeuvre.

Even tough this problem is still very similar to the *ego-docking* case, particularly from the perceptual point of view, the sensory-motor coordination differs significantly since the geometric transformation between the motor frame (robot frame) and the sensor frame (docking station frame) depends on the instantaneous position and orientation of the mobile platform with respect to the docking station, as shown in Figure 5.4.

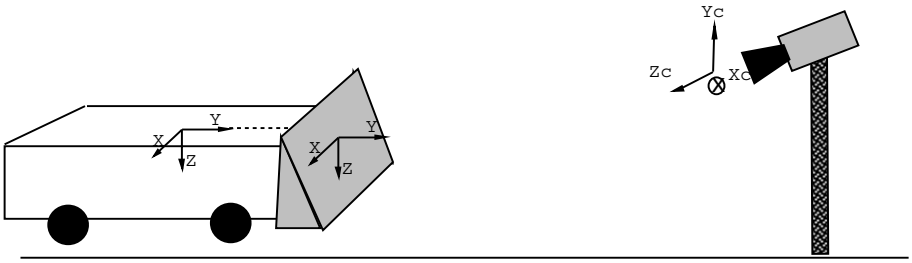

Figure 5.4: Coordinate systems involved in the eco-docking problem.

Let us now analyse how these different coordinate systems are related. Let {R}, {S} and {C} denote respectively the robot, robot front panel and camera coordinate systems. The linear velocity of any point in the robot front panel, with respect to a fixed frame

and expressed in the robot coordinate frame [Yioshikawa, 1990], is given by :

$$ {}^{R}\mathbf{T}_S = \mathbf{T}_R + \boldsymbol{\omega_R} \times {}^{R}\mathbf{P}_S \, , \tag{5.7} $$

where ${}^{R}\mathbf{P}_S$ denotes the position vector of a general point in the robot front panel expressed in the robot coordinate frame. By introducing the coordinate transformation between the camera and robot coordinate systems,

$$ {}^{C}\mathbf{P} = {}^{C}\mathbf{R}_R \, {}^{R}\mathbf{P} + {}^{C}\mathbf{P}_{OR} \, , \tag{5.8} $$

and expressing the linear velocity of $\mathbf{P}_S$ in the camera coordinate frame, we can rewrite equation (5.7) as :

$$ {}^{C}\mathbf{T}_S = {}^{C}\mathbf{R}_R \, (\, \mathbf{T}_R + \boldsymbol{\omega_R} \times [\, {}^{R}\mathbf{R}_C \, (\, {}^{C}\mathbf{P}_S - {}^{C}\mathbf{P}_{OR} \,) \,] \,) \, , \tag{5.9} $$

where ${}^{C}\mathbf{P}_{OR}$ is the position vector of the origin of frame {R}, expressed in the camera coordinate system. Comparing with the ego-docking case, the difference stems mainly from the rotation around an object centered coordinate frame instead of a camera centered coordinate frame. The motion perceived for each point in the robot front panel does not depend on the absolute distance to the camera alone, ${}^{C}\mathbf{P}_S$, but also on the distance to the robot coordinate system. When the robot is not too close to the docking station, this distance is generally smaller than the distance to the camera. As a consequence, the influence of the rotation in the motion perceived in the image plane, is smaller in the *eco-docking* case, when compared to the *ego-docking*.

In the absence of rotation, both the *ego-docking* and *eco-docking* are identical as from a perceptual point of view one cannot distinguish the case where the camera is approaching the surface or vice-versa. For the examples tested, we assume that the robot moves in a piecewise-linear trajectory, so that both docking problems become exactly equal.

The major change, however, is that ${}^{C}\mathbf{R}_R$ is no longer constant, as it was in the *ego-docking* case with the camera fixed to the robot itself and assuming that there are no mechanical degrees of freedom. Particularly, the pan angle changes continuously as the robot evolves. In practice all these terms can be seen as perturbations to the model assumed in the *ego-docking* case. Therefore, the use of feedback control strategies becomes important as it reduces the effect of the external perturbations or unmodeled terms, on the global performance of the closed loop system.

5.3 Planar surfaces in motion revisited

In both docking situations described, the *ego-docking* and the *eco-docking*, it can be assumed that the camera is observing a planar surface in motion. In the *ego-docking*, the camera is viewing the docking surface, whereas in the *eco-docking* the camera is observing the robot front panel. Therefore, the hypotheses used in Section 4.2 are still valid. It is then possible to approximate the flow field of the planar surface in motion with an affine model which parameters can be estimated using the normal flow, according to what has been detailed in Section 4.2.

5.4 Visual Based Control

Within this section, we will show how to use the parameters of the affine flow model to control the docking manoeuvre. The emphasis will be on the direct use of visual measurements, the normal flow in this case, to control motion and accomplish a given task. The coordination of perception and action results in an improved performance as the visual information is continuously being used to monitor the robot behaviour. The objective of the control system, both in the *ego-docking* and the *eco-docking* problems is twofold :

Heading control - The goal of the heading control is to align the camera axis and the docking surface normal, during the docking manoeuvre. In this way, the robot approaches the surfaces perpendicularly.

Time to crash - The robot forward speed is controlled depending on the time to contact, thus slowing down when approaching a wall. Several authors have shown the importance of this parameter in the control of locomotion in various animals [Gibson, 1958, Lee, 1976, Wann et al., 1993].

A point worth mentioning is that the control strategy we propose for the docking behaviour is such that while moving in open space (say for the *ego-docking* case), the control loop will only adjust the robot forward speed to a cruise speed and the heading direction remains unchanged (thus moving on a straight path). Only when the docking

surface appears, will the control generate changes in the heading. One can say that the behaviour is elicited by the visual information without any need for "higher order" computations, to launch the behaviour.

5.4.1 Ego-docking behaviour

Once the affine optical flow parameters have been estimated, we have to establish the control laws to command the robot. The first step consists in translating the motion parameters (angular and linear velocities) of the camera into the control variables (robot forward speed and angular speed) or, in other words, relate the sensor (camera) and control (robot) coordinate frames.

Using equation (5.6) together with the affine flow parameter equations (4.8) leads to :

$$\begin{aligned} u_0 &= f_x \left[\tfrac{dy}{Z_0} + \sin\psi\right] \omega_{rz} & \qquad v_0 &= -f_y \tfrac{T_{ry}\cos\psi}{Z_0} \\ u_x &= -\tfrac{T_{ry}\sin\psi + \gamma_x \omega_{rz} dy}{Z_0} & \qquad v_x &= \tfrac{f_y}{f_x}\left[\tfrac{T_{ry}\gamma_x\cos\psi}{Z_0} - \omega_{rz}\cos\psi\right] \\ u_y &= \tfrac{f_x}{f_y}\left[\cos\psi - \tfrac{\gamma_y dy}{Z_0}\right]\omega_{rz} & \qquad v_y &= \tfrac{\gamma_y\cos\psi - \sin\psi}{Z_0} T_{ry} \end{aligned} \quad . \tag{5.10}$$

The term v_0 is inversely proportional to the *time to crash*, which is the time left before a collision occurs, provided that the robot keeps the same speed. In fact, v_0 consists in a ratio between the forward robot speed and the distance measured along the optical axis. Therefore, to control the docking speed of the robot, v_0 should be kept constant by controlling the forward speed T_{ry}. To control the heading direction, instead, we can use the visual parameter v_x. To align the camera axis perpendicularly to the visualized surface the controller must regulate γ_x to zero.

To keep the time to crash constant, by controlling the robot speed, we define the error signal as the difference between the observed v_0 and a nominal desired v_o^{ref} :

$$e_v = v_0^{ref} - v_0 \ ,$$

while the robot speed is incrementally adjusted (thus introducing an integrator in the

control loop) :

$$\Delta T_{ry}[n] = G_{cv}(e_v) \,, \tag{5.11}$$

where G_{cv} is at the moment a PID controller.

To control the robot orientation, let us then consider a controller with the following structure:

$$\omega_{rz} = -K\gamma_x \,, \tag{5.12}$$

where K may denote a simple gain or some filtering mechanism. Using this control structure in the equation of v_x, yields :

$$v_x = -\frac{f_y \cos\psi}{f_x}\left(\frac{T_{ry}}{Z_0 K} + 1\right)\omega_{rz} \,,$$

which, in turn, can be rewritten as

$$\begin{aligned} \omega_{rz} &= -\frac{f_x}{f_y}\frac{Z_0 K}{(T_{ry} + Z_0 K)\cos\psi} v_x \\ &= -\frac{G_\omega}{\cos\psi} v_x \,. \end{aligned} \tag{5.13}$$

Therefore, directly regulating v_x to zero accomplishes the goal of regulating γ_x, thus orienting the robot perpendicularly to the surface. Note how $\cos\psi$ is a sort of sensitivity coefficient. If the camera is pointing straight ahead, then we cannot determine the qualitative orientation of the docking surface and the heading control is no longer possible. It is however a structural parameter that can be easily set by pointing the camera slightly downwards. The rotation velocity controller works with an error signal given by :

$$e_\omega = v_x \,,$$

with the speed being controlled by :

$$\omega_{rz}[n] = G_{c\omega}(e_\omega) \,. \tag{5.14}$$

Figure 5.5 illustrates the overall structure of the controller.

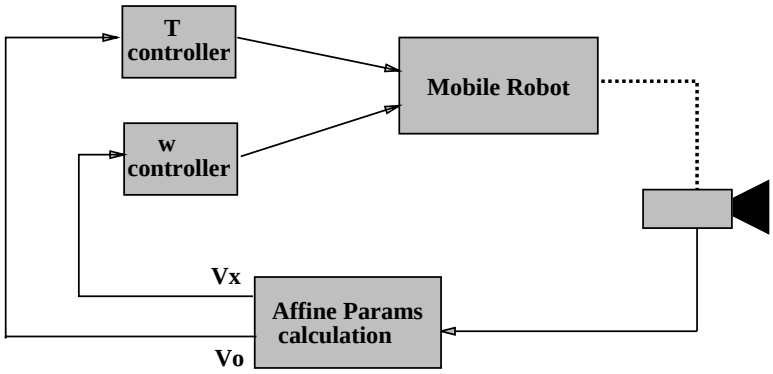

Figure 5.5: Structure of the overall heading and docking speed control system.

5.4.2 Eco-docking

In the absence of rotation, the visual parameters used in the control system are given by :

$$\begin{aligned} v_0 &= -f_y \frac{\cos\psi \, T_{ry}}{Z_0} \\ v_x &= -\frac{f_y}{f_x} \frac{\gamma_x \, \cos\psi \, T_{ry}}{Z_0} \, , \end{aligned} \tag{5.15}$$

which is **exactly** the same situation as in equations (5.10) except that there is an inversion on the rotation direction. Note that, by simply changing the sign of the rotation control loop, the same strategy is able to cope with both docking problems which, from a perceptual point of view, are in fact very similar. In the presence of small rotations, the differences regarding the changes in ${}^{C}\mathbf{R}_{R}$ are balanced by the control system during operation, while disturbances due to rotation are less noticeable that in the ego-docking case.

5.5 Results

The system has been tested in real time on a TRC Labmate mobile platform and a camera with a 8mm lens. For the ego-docking problem, the camera was placed in the front part of the robot facing the ground plane with an angle of about 60 degrees and roughly aligned with the center of the robot as shown in Figure 5.6, while for the eco-docking, the camera was also slightly pointing downwards.

Figure 5.6: Experimental setup used for the *ego-docking* behaviour.

The images are grabbed with a resolution of 128×128 pixels and a central window of 80×80 pixels is used to compute the normal flow. The normal flow is then used to determine the affine parameters and the control signals are synthesized. The system is running approximately at 1 Hz on a Eidobrain image processing workstation. Figure 5.7 shows an image of the normal flow measured for the ground plane, which is used to estimate the affine parameters.

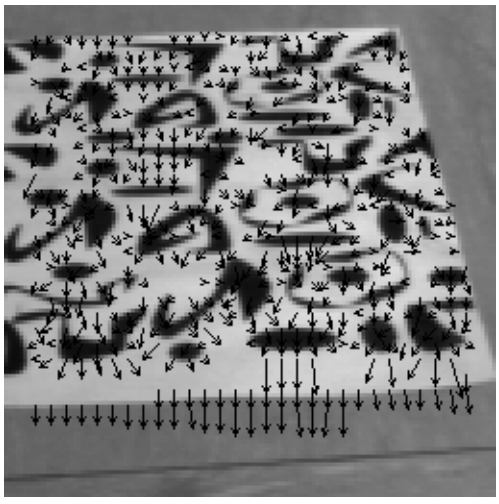

Figure 5.7: Sample of the normal flow field used to estimate the affine motion parameters.

Several tests were made using both the *ego-docking* and *eco-docking* behaviours. Fig-

ure 5.8 shows the trajectory of the robot during an ego-docking manoeuvre. Initially there is an angular difference of approximately 45^0 between the robot heading and the surface normal. During the manoeuvre the robot visually aligns itself with the direction normal to the surface, while controlling the forward speed.

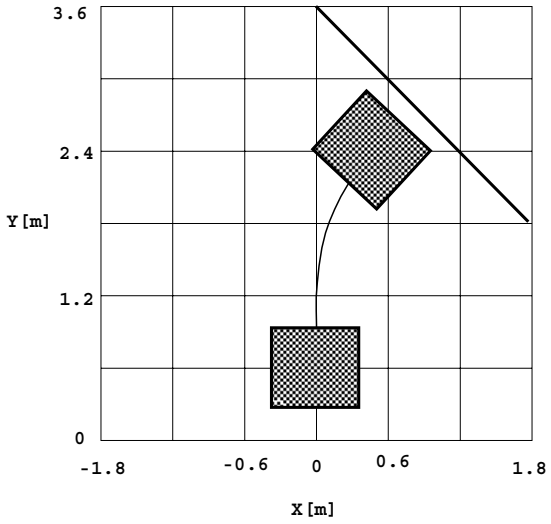

Figure 5.8: Trajectory during a real *ego-docking* manoeuvre. The trajectory is recovered using odometry.

Figure 5.9 shows the evolution of the forward speed and angular position (heading direction) of the robot during operation. Note how the velocity and orientation vary smoothly as the robot approaches the goal.

Finally, several tests were made for the case of the eco-docking concept, using the same controller apart from a sign inversion in the rotation control law. Figure 5.10 shows a plot of the robot trajectory during an *eco-docking* procedure. The evolution of the robot forward speed and heading direction is shown in Figure 5.11. It is seen that while the robot is far from the docking station, the speed control loop originates an increase of the robot velocity and, when the robot gets closer to the docking station, the speed decreases to a final stop.

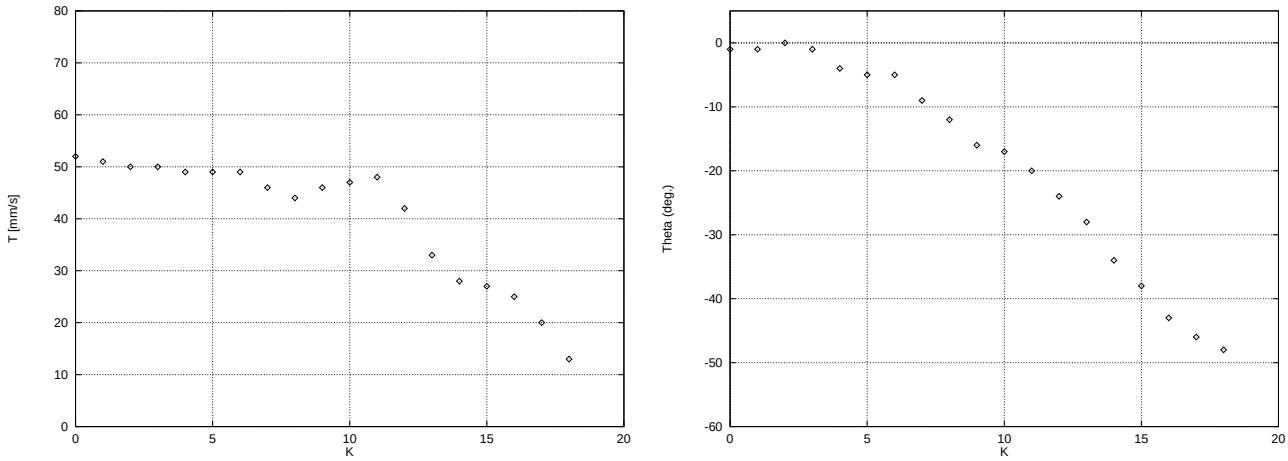

Figure 5.9: Ego-docking example. The left plot shows the evolution of the robot forward speed in $[mm/s]$, while the left plot shows the evolution of the heading direction, in degrees, during the manoeuvre.

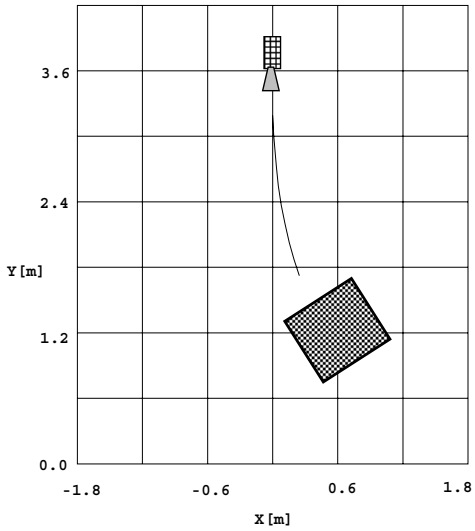

Figure 5.10: Trajectory during the *eco-docking* manoeuvre to a static docking station. The trajectory is recovered using odometry.

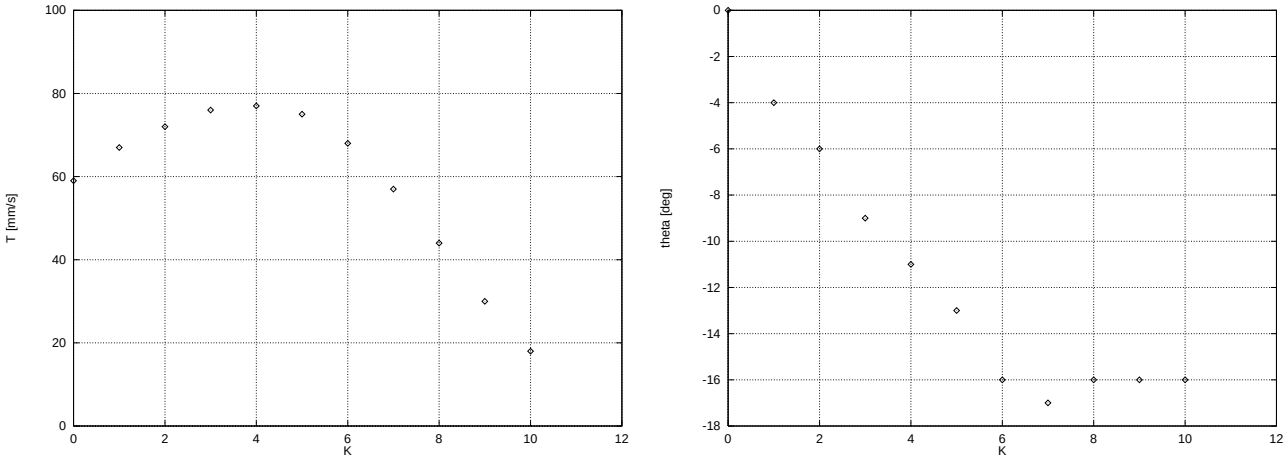

Figure 5.11: Eco-docking example. The left plot shows the evolution of the robot forward speed in $[mm/s]$, while the left plot shows the evolution of the heading direction in degrees, during the docking manoeuvre.

5.6 Conclusions

The docking behaviours described here, illustrate the intimate relation between perception and action. On one hand, the visual measurement used (normal flow) is elicited by the motion of the robot. On the other hand, the perception/action loop is not decoupled in the sense that the performance of the perceptual processes is also a function of the control parameters.

The robot motion is uniquely controlled by the direct link between the normal flow estimation and the motor commands generated by the controller and, as a consequence, no matter what the robot "sees" it will end up in front of the "docking wall" and perpendicular to it. There is no need to have "consciousness" of the situation since the behaviour is elicited directly by the purpose and the conditions in the environment.

More specifically, an active vision approach for the problem of docking has been presented in two different situations : the *ego-docking* and the *eco-docking*.

In the *ego-docking*, each robot is equipped with an on board camera and the egomotion controlled during docking manoeuvre to a given surface. In the *eco-docking*, instead, the camera and all the necessary computing resources are placed on a single external docking station, able to serve several robots.

In both situations, the goal consists in controlling both the robot orientation, aligning

the camera optical axis with the surface normal, and the approaching speed (slowing down during the manoeuvre). These goals are accomplished without any effort to perform 3D reconstruction of the environment or any need to calibrate the setup, contrasting with traditional approaches.

Our approach is based on the direct use of image measurements to drive the motion controller, without any intermediate reconstruction procedure. Again, we use the normal flow field as the visual input data.

An affine model is fitted to the measured motion field, as already explained in Chapter 4. The affine parameters are expressed as a function of the robot motion and directly used to close the motor control loop. The closed loop strategy proposed uses direct visual measurements to control the robot forward speed (based on time to crash measurements) and heading direction. The same control structure is used for the *ego-docking* or *eco-docking* cases, in spite of some differences which have been discussed.

A real time implementation was done and a robust docking behaviour achieved, with examples given both in the *ego-docking* and *eco-docking* problems. There is no need to calibrate the camera intrinsic or extrinsic parameters nor is it necessary to know the vehicle motion.

Chapter 6

Gaze Control

We have highlighted the fact that perception and action are so tightly connected that they cannot be considered separately. This link between action and perception is present in many animals in nature, as we have already mentioned. Particularly, many animals exhibit eye movements which are continuously used to perceive the surrounding space. Humans, for instance, make use of a large variety of controlled eye movements, which are well studied and described in the literature [Robinson, 1968].

Also, in computer vision, there are many advantages in using agile camera systems. For example, we have mentioned in the previous chapter, that more complex docking manoeuvres could be accomplished by actively controlling the gaze direction of a camera while we are moving throughout a scene.

In this chapter, we discuss the problem of gaze control in general terms, as for object tracking. A stereo head system **Medusa**, with four mechanic degrees of freedom (two independent vergences, tilt and pan), designed for active vision applications is described. A control system for the head is proposed and tested. Some examples on visual tracking are shown and discussed.

6.1 Introduction

As referred previously, computer vision systems have been traditionally designed disregarding the observer role (camera motion, stereo rig geometry, lens parameters, etc) in

the perceptual process. Actually, most systems were designed to process images which had been prerecorded in some sense or, at least, acquired independently of the perception process itself. However, most biological vision systems do not just *see* the surrounding space but they actively *look* at it and, very often, their visual processing is related to a specific task or set of tasks [Pahlavan and Eklundh, 1992, Blake and Yuille, 1992].

Within this new framework, the ability to perform controlled eye movements, driven by visual stimuli, has several advantages on improving perception [Bandopadhay et al., 1986, Ballard, 1991, Grosso, 1994]. The advantage of fixating a point in the environment is discussed in [Fermüller and Aloimonos, 1992] for various tasks in navigation. The problem of establishing and maintaining a given orientation, between two stereo cameras and a static or moving target, is addressed in [Grosso and Ballard, 1993, Grosso, 1993]. By fixating and tracking an object, we can segment the object from the background (motion blur) and establish an object centered coordinate system, which is more suitable for recognition purposes [Ballard, 1991].

To further understand these processes, different research laboratories have developed active vision systems which incorporate cameras with various mechanical and/or optical degrees of freedom (see [Christensen et al., 1994] for an overview). These systems have agile "eyes" (cameras) that can verge to maintain a fixation point on an interesting object and track it over time; then switch the attention to a different object, bringing it to the image center, and so on.

Some of the existent heads include symmetric vergence movements for both eyes [Krotkov, 1989, Ferrier, 1991] allowing the separate control of the direction of gaze (often referred to as *version*) and vergence. Other prototypes make use of independent vergence degrees of freedom and allow very fast saccadic rotations [Coombs, 1991, Murray, 1992]. The stereo heads described in [Coombs, 1991, Crowley et al., 1992] are mounted on top of robotic manipulators. The stereo head described in [Pahlavan, 1993] was designed to exhibit most of the human oculomotor system degrees of freedom, thus incorporating 13 degrees of freedom [Pahlavan and Eklundh, 1992].

Besides the eye movements, there is another important process in many biological vision systems : the accommodation [Bruce and Green, 1985, Carpenter, 1988], which

adjust the optics of the eyes to focus objects at various distances[1].

Some stereo heads have optical degrees of freedom [Krotkov, 1989, Christensen, 1991], [Crowley et al., 1992, Pahlavan, 1993] as well as mechanic degrees of freedom. In fact, vergence movements are driven by simultaneous stimuli of accommodation and disparity [Carpenter, 1988]. Disparity and accommodation can be put together to improve the perception of depth, as in [Abbott and Ahuja, 1988, Krotkov, 1989, Ahuja and Abbott, 1993].

A key issue in all these problems is the sensorimotor coordination or, in the particular case of computer vision, visuo-motor coordination. What kind of visual information can be used for active control of the camera head systems, how can it be done and with what purpose. Ideally, one should directly use some "simple" image measurements to provide visual feedback for motor control, as it seems to be the case in many biological vision systems, rather than performing exhaustive calculations for scene reconstruction.

To further understand these problems, we have developed a stereo head with 4 degrees of freedom, for active vision experiments [Trigt et al., 1993, Santos-Victor et al., 1994b]. The head is composed of two B&W video cameras, which can verge independently (hence fixating closer or further objects), a common tilt unit (up/down) and a common pan unit (left/right). To some extent, it corresponds to an anthropomorphic design and provides the main degrees of freedom available in the human ocular system and in many other living creatures [Yarbus, 1967, Robinson, 1968, Land, 1975, Carpenter, 1988].

The head control allows the performance of basically two kind of movements : saccadic movements to rapidly switch the attention from one point in the scene to a different one, and smoother movements to track moving objects in the environment. As mentioned before, accommodation is an important perception mechanism in biological vision systems [Bruce and Green, 1985]. Accommodation will be possible in the future by introducing active lens control, namely focusing.

This chapter describes the design and construction of the stereo head, and addresses some early experiments and problems of active control of eye movements.

[1] Accommodation can be achieved in vertebrates either by moving the eye lens backwards or forwards (as in fish, amphibians and snakes), or by changing the lens shape (other reptiles, birds and mammals), thus altering its power.

6.2 System Description

The stereo head described here, is composed of two black and white cameras mounted on two independent vergence axes. Additionally, the cameras have a common tilt degree of freedom and can also pan around a "neck" structure. The inter-ocular distance, the baseline, can be manually set between 60 and 200 mm. This feature is important when working with different depth ranges. A schematic design of the head is shown Figure 6.1.

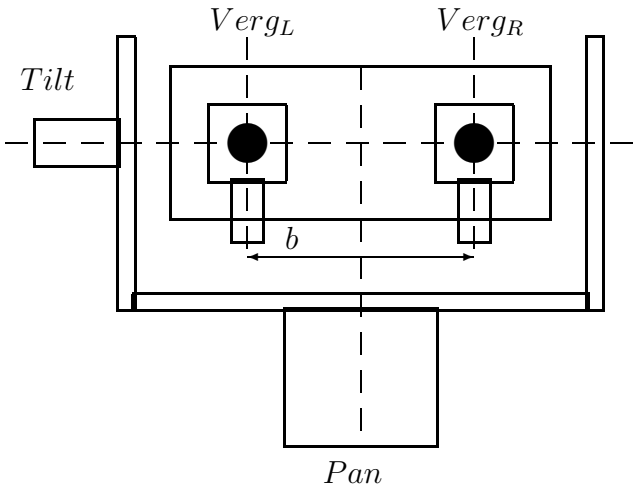

Figure 6.1: Schematic representation of the stereo head. The degrees of freedom are two independent vergences and common tilt and pan. The baseline, b, can be manually adjusted.

Moreover, the camera mounting system is such that it can be adjusted to ensure that both vergence and tilt axes intersect in the cameras optic centers, if desired. If the rotation axes do not pass through the optic centers, each camera rotation necessarily leads to a small translation component. As a consequence, the target motion in the image plane varies with depth as well as the amount of rotation. The depth dependence can then be used to constrain stereo correspondences [Yuille and Geiger, 1990, Francisco, 1994].

However, by adjusting the rotation axes to pass through the cameras optic center, the kinematics of the global structure and, as a consequence, the control system design are significantly simplified . Of course, this condition can only be met approximately as, for

instance, the location of the cameras optic centers is extremely hard to determine, even with complex calibration methods. Also, in an active configuration it would be necessary to continuously update the head calibration parameters as they change over time. Again, it is hardly likely that many natural vision systems are that well "calibrated" and yet they have outstanding performances. Our approach here, is then to ensure that the optical axes and the vergence and tilt axes do intersect approximately, while when designing the various visual behaviours, one must guarantee that these errors do not degrade significantly the visuomotor capabilities.

In order to accurately track moving objects, we have specified an angular resolution of 0.01^0. This value is well suited to the resolution of a good CCD chip and a large focal length, which represents a worst case scenario [Pahlavan and Eklundh, 1992]. The maximum angular speed and angular acceleration were also set as design specifications. Regarding the different joints rotational speed, the maximum value was set at $180^0/s$ which is roughly half the speed of human saccades [Pahlavan and Eklundh, 1992, Robinson, 1968], thus enabling the head to rapidly switch attention between different targets. The maximum value for the angular acceleration was set at $1080^0/s^2$.

The mechanical structure was carefully designed in order to prevent vibrations while performing fast movements. The required motor torques were determined by calculating the moments of inertia of the different parts of the head, and by the set of angular speed and acceleration specifications. We have used DC motors with harmonic drives (with negligible backlash) coupled with encoders for vergence, tilt and pan axes.

The main characteristics of the different joints of the stereo head are summarized in Table 6.1. We have also included the minimum achievable speed by each joint, which depends on the encoder resolution and sampling frequency. This is an important limit when tracking slowly moving objects (or very far from the cameras).

The stereo head host system is a 486/50MHz PC computer, equipped with a DT2851 frame grabber and a DT2859 video multiplexer. The motors are driven by Advanced Motion Controls MC3X series PWM servo amplifiers and we are using two Omnitech Robotics MC3000 axes control boards. Each of these boards controls up to 3 axes and, therefore there are two unused control channels that will be used in the future for active lens control. Additionally, each joint is equipped with switches for limit detection during

	Resolution (degrees)	Min. speed (deg/s)	Max. speed (deg/s)	Accel. (rad/s^2)
Verg.	0.0031	1.55	180	6 π
Tilt.	0.0018	0.9	180	6 π
Pan.	0.0036	1.8	180	6 π

Table 6.1: Medusa main characteristics. This table shows the angular resolution, minimum and maximum speed and angular acceleration for each joint.

operation and with an inductive position sensor for fine homing during startup. Figure 6.2 shows the mechanical designs of the stereo head. It can be seen how the cameras have been mounted in order to be able to move the optical center to the point where the vergence and tilt axes intersect. The designs include also the switches for limit detection and the inductive sensors for fine homing. A picture of Medusa is also shown on Figure 6.2.

6.3 Control Architecture

The main problem of gaze control is that of determining what kind of visual cues can be used to control the motion of our eyes, how it is done and with what purpose. This has been an interesting topic of research in psychology, psychophysics and psychophysiology for many years [Doorn and Koenderink, 1983, Gibson, 1950, Pobuda and Erkelens, 1993, Robinson, 1968, Warren and Hannon, 1990] and provides an important source of information and useful ideas for the active control of camera heads.

The procedure of tracking a moving object can be seen as a cycle of the following steps : image acquisition; calculation of the target object position; generation of new motor commands to reposition the target in the center of the retina. The calculation of the target position should be understood in the broader sense of determining its position, and/or velocity, binocular disparity or whatever variables might be relevant to compute. In our artificial stereo head, this sequence of steps is described by the block diagram shown in Figure 6.3.

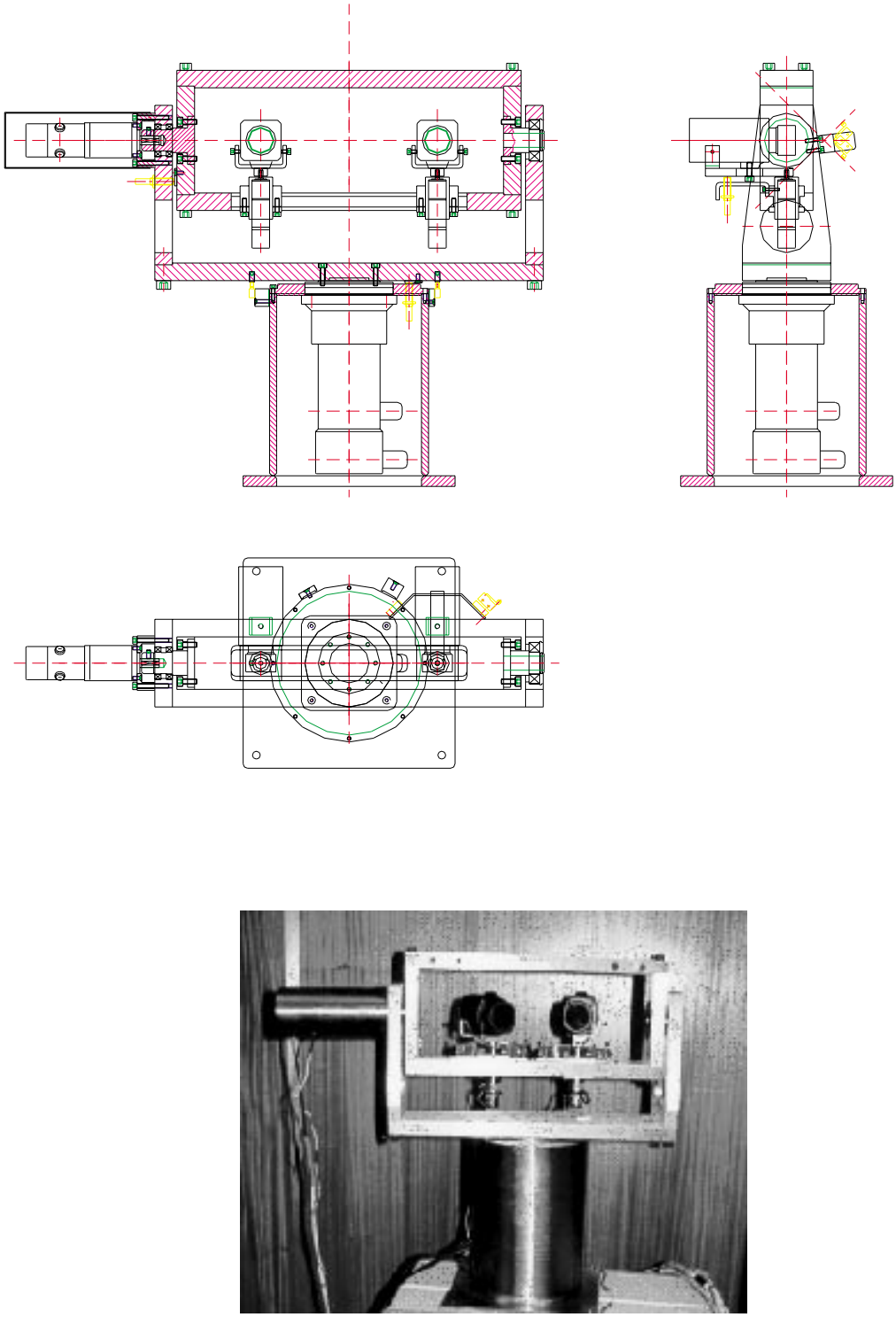

Figure 6.2: Detailed mechanical designs of the stereo head. The bottom image shows a picture of Medusa.

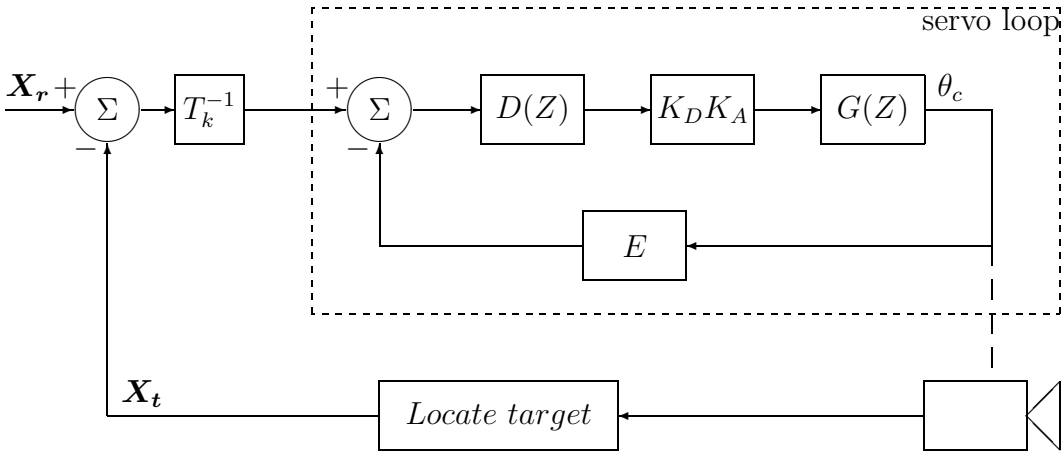

Figure 6.3: Complete loop for gaze control. This figure shows the cycle of image acquisition, locating the target and generating new motor commands to keep the object centered in the retina.

The variables $\boldsymbol{X_r}$ and $\boldsymbol{X_t}$ represent respectively the target reference position (position, velocity, etc) and the observed target position. Note that these quantities are measured in the retinal (image) plane. The position of the motor shaft, holding the camera, is denoted by θ_c. The term T_k^{-1} represents the inverse kinematics of the head (or an inverse jacobian), relating the error measured in the image plane into the appropriate joint angles or velocities. Naturally, this process can be an accurate description of the head kinematic chain or some simplified version. In the figure, D(z) denotes the digital servo controller, K_D is the DAC conversion factor, K_A is the amplifier gain, G(z) is the transfer function of the motor and E is the encoder function.

Therefore, one can consider that the control system is composed of an outer loop and an inner servo loop, as shown in the picture. The inner loop operation consists in generating new motor commands, based on the new desired joint positions and feedback from the encoders. The outer loop is responsible for determining new joint positions, using the visual information as feedback, and providing input reference values for the low level control system. Each of these control levels will be addressed separately in the following sections

6.4 Servo Loop

This section is devoted to the problems of the low level control, while the visual based control will be detailed in Section 6.5. The advantage of considering these two problems separately stems from the fact that, while the high level control system is difficult to model, the servo loop can be carefully modeled and analysed. Since the gaze controller runs at a much lower frequency than the servo loops, we can assume that the motors transient responses vanish after each sampling period. Hence, we can decouple both control loops and focus our attention on the visual control system.

For the servo loop, we performed extensive modeling of the different subsystems in the control loop, and a suitable controller was designed based on specifications, both for position and velocity control. Position control is suitable for fast movements between different targets, appropriate for saccadic motions, and the velocity control mode is suitable for the smooth tracking of a moving target.

The low level control makes use of the digital filter of the MC3000 control board. This filter provides programmable compensation of the closed loop system to improve response and stability. The discrete filter transfer function is given by :

$$D(z) = K\frac{z - A}{z + B} \tag{6.1}$$

The compensation filter, together with the sampling time of the control board, affects the dynamic response and stability of the servo system. The filter zero, A, pole, B, gain, K and the sampling time, t_s, can be set during the controller design process. The parameters A and B can vary between 0 and 1, t_s varies between 64 and 2048 μs and the gain factor K has a value between 0 and 64.

The axes control board can operate in 4 different modes: position control, proportional velocity control, integral velocity control and trapezoidal control mode. For all these operating modes, we will now design a suitable controller for closed loop operation.

6.4.1 Position Control

In the position control mode, the board reads the encoder pulses and compares the observed position to the desired position. The resulting position error is input to the control

filter (equation (6.1)), and used to move the motor shaft. This operation mode is shown in Figure 6.4.

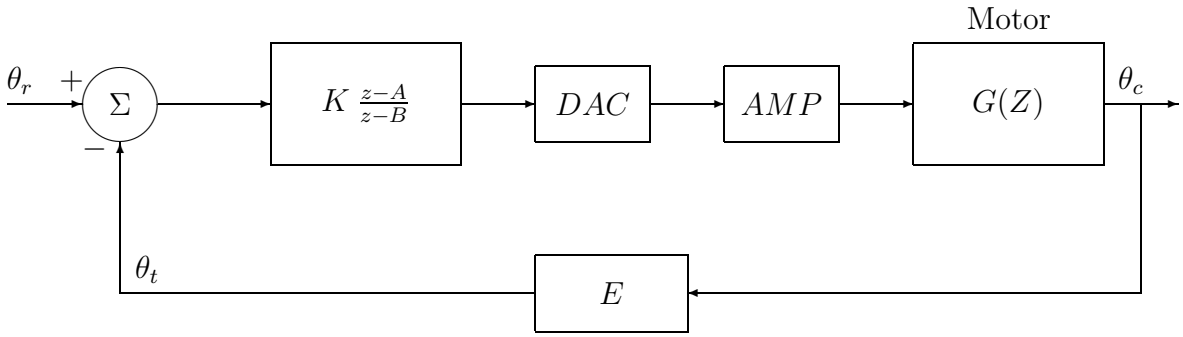

Figure 6.4: Block diagram of the complete system in position control mode.

The motor transfer function [Omnit. Robotics, 1989], relating the motor input voltage and output angular position in continuous time, $G_c(s)$, is given by :

$$G_c(s) = \frac{1/K_E}{s(sT_M + 1)} \tag{6.2}$$

where K_E is the motor voltage constant and T_M is the mechanical time constant. The motor electrical pole can be neglected when compared to the dominant mechanical pole. To find the discrete transfer function, as it is seen by the computer, and assuming that the system output is sampled by a zero order hold mechanism, we can use the step invariant method [Astrom and Wittenmark, 1986, Franklin et al., 1986]. The discrete transfer function, G(z), follows from :

$$G(z) = (1 - z^{-1})\mathcal{Z}(\mathcal{L}^{-1}\frac{G_c(s)}{s}) \tag{6.3}$$

in which $\mathcal{Z}(\mathcal{L}^{-1}F(s))$ is the z transform of the time series whose Laplace transform is given by $F(s)$. The general expression for G(z) is given by :

$$G(z) = \frac{1}{K_E}\frac{(T - T_M + T_M e^{-\frac{T}{T_M}})z + (T_M - T_M e^{-\frac{T}{T_M}} - Te^{-\frac{T}{T_M}})}{(z-1)(z - e^{-\frac{T}{T_M}})} \tag{6.4}$$

The choice of the sampling time is critical for the system. Since the motors position sensor is discrete (an encoder) there will be a minimum velocity that can be detected by the control board and, therefore, that can be used to command the motors. This velocity is one encoder pulse per sampling time. For a sampling period of $1ms$, and for the vergence motors this minimum speed amounts to $3^o/s$. For this reason, we have chosen the sampling time, T, to be 2.048 ms, the maximum value allowed by the board. Hence, the minimum velocity will be reduced to $1.5^o/s$.

Considering that, presently, the system is running at a frequency of about 6 to 7Hz and that, we expect in the future to reach a speed of about 15-25Hz, the sampling period is still small enough in order to guarantee the vanishing of all the transients responses between successive image acquisitions.

The mechanical time constant T_M is derived to incorporate the effects of the total inertial moment for the joint [Omnit. Robotics, 1989].

$$T_M = \frac{R\ J_{tot}}{K_E\ K_T} \tag{6.5}$$

where R is the motor armature resistance, J_{tot} is the total inertial moment for the joint and K_T is the motor's torque constant. The output range of the 8 bit DAC from the control board is 20 V and, therefore, the conversion factor K_D can be determined as:

$$K_D = \frac{20}{2^8} \quad [\text{V/counts}] \tag{6.6}$$

The amplifier gains, K_A, have been measured (see [Trigt et al., 1993] for a detailed description). For the vergence, a value of 1 was obtained while for tilt and pan we obtained 1.95 and 3.56, respectively. The encoder function is given by :

$$E = \frac{4NG}{2\pi} \tag{6.7}$$

In this expression, N is the number of encoder lines and G the gear ratio. The factor 4 is present because the control boards use quadrature counts, thus increasing the resolution of the closed loop. Combining the different amplification factors, a new transfer function $P(z)$, is obtained according to the following expressions:

$$P_{verg(z)} \quad = \quad 0.155\frac{z + 0.96}{(z-1)(z-0.88)} \tag{6.8}$$

$$P_{tilt(z)} = 0.149\frac{z+0.95}{(z-1)(z-0.86)} \tag{6.9}$$

$$P_{pan(z)} = 0.106\frac{z+0.92}{(z-1)(z-0.77)} \tag{6.10}$$

The root locus for the uncompensated vergence system, $P_{verg(z)}$ is shown in figure 6.5. It shows that if the loop is closed using a simple proportional controller, then the system will become unstable.

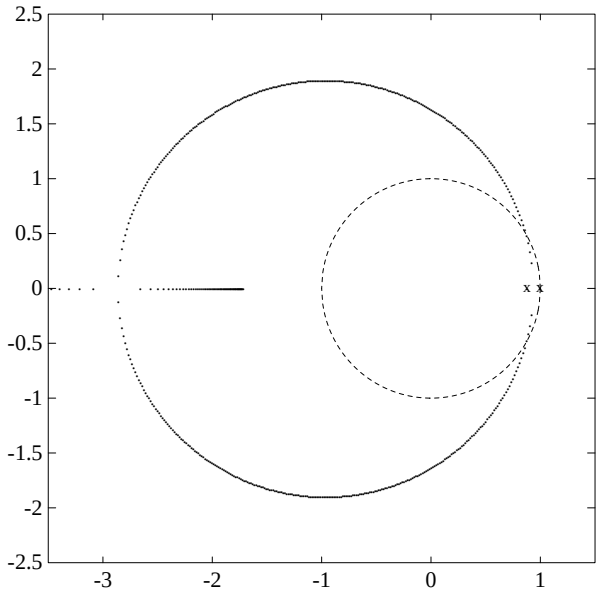

Figure 6.5: Root Locus of uncompensated system for vergence. The unit circle is shown in dotted line. The system becomes unstable with a proportional controller.

The digital filter, expressed in equation (6.1) was used to improve the system damping and stability by suitably placing the controller zero and pole. As a method, the controller zero, A, was chosen in order to cancel one of the poles of the vergence, tilt and pan dynamic systems (equations (6.8) through (6.10)). The controller pole is chosen to be at the origin. Basically this strategy introduces a derivative action to improve the system damping.

According to this strategy, the Root Locus of the compensated vergence system

D(z)P(z) is shown in Figure 6.6. It is now seen that the system is stabilized for a set of

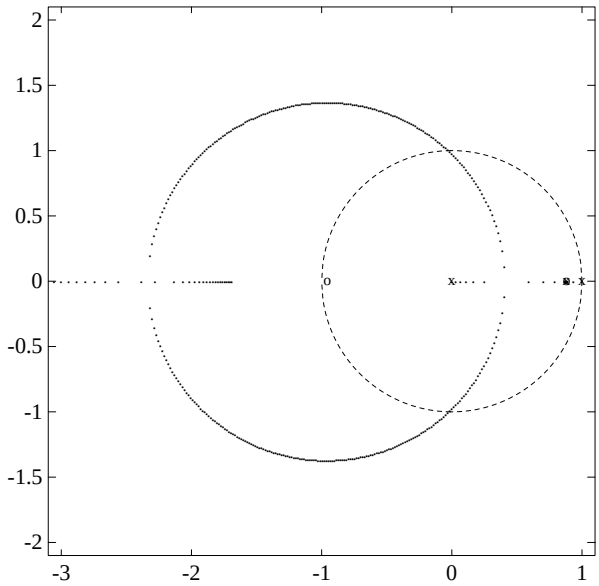

Figure 6.6: Root Locus of PD compensated system for vergence.The unit circle is shown in dotted line.

values of the controller gain, K, . The controller gain was chosen in order to have double real poles in the closed loop system. This criterion ensures that the system response will not overshoot. The closed loop poles are the roots of the following polynomial :

$$D(z)P(z) + 1 = 0 \tag{6.11}$$

The value of the control gain is obtained by solving the roots of equation (6.11) and setting the imaginary part of the roots to zero. For the three different axes the following controllers were calculated :

$$D_{verg(z)} = 1.14\,\frac{z - 0.88}{z} \tag{6.12}$$

$$D_{tilt(z)} = 1.20\,\frac{z - 0.86}{z} \tag{6.13}$$

$$D_{pan(z)} = 1.71\,\frac{z - 0.77}{z} \tag{6.14}$$

We have simulated the system behaviour using the filter values in equations (6.12) to (6.14). Figure 6.7. shows the simulated step response of the vergence motors with load.

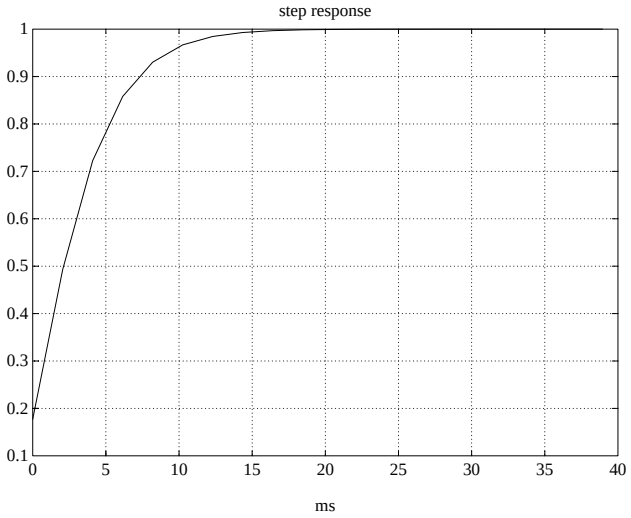

Figure 6.7: Simulated step response for the vergence motor.

It is seen that, due to the presence of a discrete integrator in the control chain, the system has zero steady state error to a position step command. Moreover, with the criterion we used for the control system design, the closed loop system shows a fast response, without overshoot. Notice how the transient response lasts for roughly 15 ms, which is still much faster than the vision loop.

6.4.2 Proportional Velocity Control

In the proportional velocity control mode, the control error is determined by the difference between the desired command velocity and the actual motor velocity. The controller is a simple proportional gain, K. Since the digital filter is disabled in this control mode, the closed loop behaviour is mainly determined by the system dynamics. The block diagram of the closed loop system under this control mode is shown in Figure 6.8

In this mode, the motor transfer function (given that the output is now the motor velocity instead of the shaft position) is given by :

$$G_c(s) = \frac{1/K_E}{sT_M + 1} \tag{6.15}$$

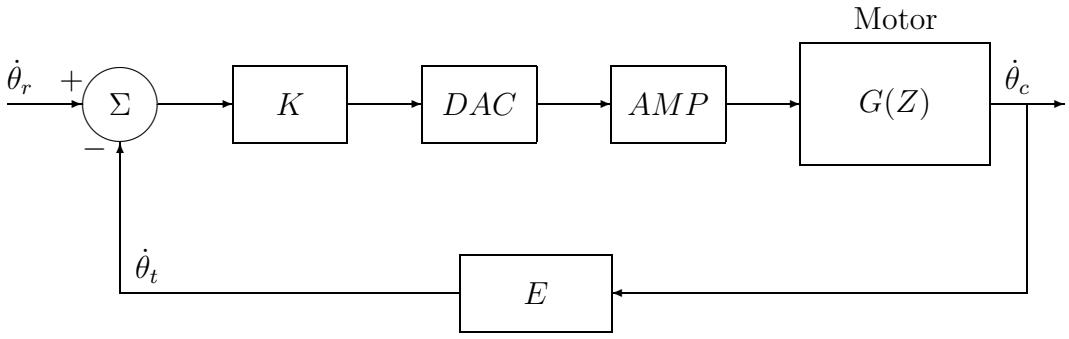

Figure 6.8: The control system block diagram when operating in Proportional Velocity mode.

As it was done in the position control mode, a discrete version of this system can be obtained by using the step invariant method. We have :

$$G(z) = \frac{1}{K_E} \frac{1 - e^{-\frac{T}{T_M}}}{z - e^{-\frac{T}{T_M}}} \tag{6.16}$$

The complete transfer function for the open loop system, considering all the different subsystems, is given by :

$$P(z) = \frac{K_D K_A E T}{K_E} \frac{1 - e^{-\frac{T}{T_M}}}{z - e^{-\frac{T}{T_M}}} \tag{6.17}$$

By using a proportional controller, K, the closed loop poles will be the roots of the following equation :

$$z - e^{-\frac{T}{T_M}} + \frac{K K_A K_D E T (1 - e^{-\frac{T}{T_M}})}{K_E} = 0 \tag{6.18}$$

For a given location of the closed loop pole, the corresponding value of the gain, K, can be calculated. There are a variety of control system design procedures that can be used at this point. We have analysed three different approaches :

(i) To guarantee the system stability, the maximum value for K will occur when the discrete loop pole is located in $z = -1$, just before becoming unstable. Using this

constraint, we have un upper bound for the controller gain :

$$K < \frac{K_E(1 + e^{-\frac{T}{T_M}})}{K_A K_D E T (1 - e^{-\frac{T}{T_M}})} \tag{6.19}$$

(ii) Another design constraint that can be used is related to the steady state error. Since none of the dynamic subsystems in the action loop has an integrator term, the steady state error to a step command (a sudden velocity change) will not be zero. There will always be a velocity error, and constant perturbations will not be rejected. However, by suitably setting the value of K, this error can be bounded by design specification. The transfer function relating the error signal and the plant input is given by :

$$H(z) = \frac{E(z)}{U(z)} \tag{6.20}$$

$$= \frac{K_E(z - e^{-\frac{T}{T_M}})}{K_E(z - e^{-\frac{T}{T_M}}) + K_E K_D K_A E T (1 - e^{-\frac{T}{T_M}})} \tag{6.21}$$

To keep the error smaller than a bounding value, err_{lim}, when time tends to infinity (and z tends to 1), the following condition must be satisfied :

$$K > \frac{K_E(1 - err_{lim})}{K_A K_D E T err_{lim}} \tag{6.22}$$

(iii) A third design criterion consists on imposing that the pole should be positive, in order to ensure that the motor velocity will not exceed the command velocity. The limit situation is reached for $z = 0$, and yields another criterion for the choice of K.

Table 6.2 summarizes the control gains to be used considering all the design constraints for the different degrees of freedom. In the implementation, we have chosen the gains corresponding to the $z = 0$ constraint, even though these gains do not meet the steady state error criterion. However, in the next section, we will analyse another velocity control mode with better performance regarding the steady state error.

6.4.3 Integral Velocity Control Mode

This control mode provides velocity control with controlled acceleration and deceleration, at a user defined maximum rate. The controller compares the desired velocity and the

Criterion	**Vergence**	**Tilt**	**Pan**	**Bound**
(i) $z > -1$	6.2	6.4	8.7	max
(ii) $err_{lim} < 0.1$	3.5	4.3	10.2	min
(iii) $z = 0$	2.9	3.0	3.8	max

Table 6.2: Bounding values for the controller gain, in order to fulfill the different design criteria in the proportional velocity control mode. The *Bound* column specifies whether the constraint is a maximal or minimal value for K

actual motor velocity, and the desired motion is achieved by incremental position moves, constrained by the maximum acceleration/deceleration.

A point worth stressing is that the position controller is being used in order to achieve the desired velocity command. Therefore, regarding the dynamics and controller parameters, the same analysis that was done for the position control mode, is still valid for the integral velocity control mode.

The main difference, when compared to the proportional velocity control mode, is that since the compensation filter is now used, the steady state velocity error is zero. This characteristic of the integral velocity control mode is suitable for smooth tracking of a moving target. We have set the a(de)acceleration to 1.3 π $rad\, s^{-2}$.

6.4.4 Trapezoidal Profile Control Mode

The trapezoidal profile control mode, provides position moves while profiling the velocity and thus controlling the acceleration.

The controller starts at the actual position and generates a profile to the final position by accelerating at the specified constant acceleration until the specified maximum velocity is met, or half the position move is complete. Then, either it remains at the maximum velocity until the deceleration point or it immediately enters in the deceleration phase until it stops. After issuing the last position command, the board enters in position mode to hold to the final position.

Again, the basic mechanism being used is position control and, accordingly, the anal-

ysis made, for the closed loop behaviour in the position control mode, is again valid in this control mode.

For the different joints, step responses have been obtained and suitable values for the acceleration and maximum velocity were determined for the different joints, according to Table 6.3:

	vergence	**tilt**	**pan**
$\dot{\theta}$ (rad/s)	1.5 π	0.75 π	1.5 π
$\ddot{\theta}$ (rad/s^2)	6 π	3 π	6 π

Table 6.3: Values for the acceleration and maximum velocity to be used in trapezoidal profile control mode for the different degrees of freedom.

The values for the tilt are half as big as the values for both vergence and pan. Those values have shown the best results for the step responses.

6.5 Gaze control : the oculomotor control system

Having analysed the servo loop designs, we can now concentrate on the problem of visual control of the stereo head. In the control architecture proposed for **Medusa**, the gaze control system is responsible for determining the location and/or speed of the target and to calculate new command positions/velocities, for the vergence, tilt and pan motors.

As the servo loop is running at a much higher frequency than the visual feedback loop, from the point of view of the gaze control system, we can consider that the servo loop transients are extinct once a new image is acquired. Hence, the servo loop appears as a delay, h, when seen by the slower gaze control loop. This structure is depicted in Figure 6.9.

For many years, there has been a large interest in the study of the oculomotor control system in biological systems. Its function consists in acquiring visual targets rapidly and, once acquired, stabilizing their images on the retina in spite of the relative movements between the target and the observer [Yarbus, 1967, Robinson, 1968, Land, 1975]. Many

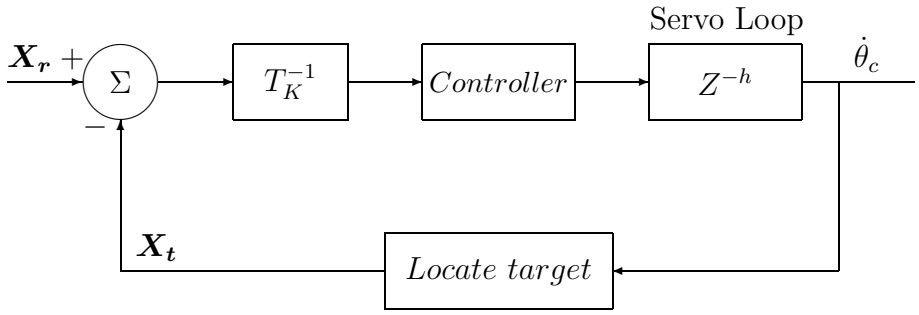

Figure 6.9: Gaze control loop. At the sampling rate of the visual feedback loop, the inner servo loops can be considered as simple delays.

animals exhibit different kinds of eye movements. In general, these movements stabilize the image in the retina, therefore minimizing the velocity blur [Carpenter, 1988]. Also, these movements allow the increase of the field of view observable by the animal. The retina in many animals, as in birds and mammals, is composed of a relatively large peripheral area, with low acuity and a much smaller central segment called the fovea with high acuity. Eye movements are used to shift this foveal region continuously so that acute vision over a wide field of view can be achieved [Bruce and Green, 1985].

We can identify five basic kinds of eye movements [Robinson, 1968] made by awake, frontal eyed, foveal animals : saccadic, smooth pursuit, vergence, vestibular and the physiological nystagmus.

Saccadic eye-movements consist of very rapid relocations of the direction of gaze to switch attention between different targets. Since the animal does not see well during these movements, due to the saccadic-suppression mechanism, these eye-movements are not controlled by visual feedback during their execution, therefore minimizing the execution time.

The smooth pursuit movements correspond to slow and accurate movements of the eyes, for example, while tracking a target by the fovea. Another kind of smooth pursuit movement is the optokinetic nystagmus, the involuntary following by the eyes of any large moving target. The optokinetic nystagmus occurs, for example, while looking through the

window of a moving train.

The vergence mechanism is used whenever the distance to the target changes, to keep the target fixated by the foveas of both eyes. If the target comes very close, further convergence is impossible, and "double vision" occurs. The main characteristic of this movement is that both eyes make equal movements in opposite directions, thus keeping the object in the fovea. Vergence movements are the slowest type of eye movements. As mentioned before, the accommodation stimulus is used for the vergence control system [Carpenter, 1988].

The vestibular movements are induced by stimuli of the vestibular system semicircular canals, and have the function of compensating the head rotations by counter-turning the eyes in order to stabilize the image on the retina. The vestibular nystagmus (also known as the vestibular-ocular reflex) is usually interrupted by saccadic movements during head rotations of large amplitude.

The physiological nystagmus are extremely small movements consisting of drifts, high frequency tremor and microsaccades that are continuously present during fixation. However this kind of movements is not involved in the larger tracking eye movements and, therefore, will not be further discussed.

The kind of eye movements implemented in the stereo head are the saccadic movements and smooth pursuit mechanism. Basically, for the saccadic movement, the eyes operate in position control, thus allowing very fast movements to relocate the direction of gaze direction from one point to another. For the smooth pursuit mechanism, the integral velocity control mode is used, thus allowing accurate tracking of the target. The first problem to be addressed, then is the determination of the target position on the retina.

6.5.1 Target detection

For the time being, we kept the visual processing to a low complexity level. The reason to do so is that we can concentrate on the problems of visuo-motor coordination without excessive computation time for visual processing.

The first goal we established for the stereo head was following a bright spot moving throughout the scene. The images are acquired with a resolution of 512 by 512 pixels

and uniformly subsampled down to 64 by 64 pixels. Then, each image is thresholded to segment the target from the background. The target position, in the visual field, is estimated by the center of mass of the image, computed at the coarse resolution.

However, our goal in the future, is to carry on this kind of visual control, using general gray-level images, and yet keeping the complexity of the visual processing to a low level. For this purpose, the images can be sampled using a space variant method, keeping a higher resolution at the image center, the fovea, and having a coarser resolution towards the periphery [Schwartz, 1977, Tistarelli and Sandini, 1993], [Wallace et al., 1994, Nielsen and Sandini, 1994]. With respect to the target detection, we can use the normal flow computed over the retinal image [Aloimonos and Duvic, 1994, Sinclair et al., 1994] to determine the target foveal speed [Martinuzzi and Questa, 1993] or determine the binocular disparity by means of correlation techniques or *cepstral* filtering as described in [Yeshurun and Schwartz, 1989].

6.5.2 Inverse Kinematics and Control

We have already described the procedure used to determine the target position in the image. The problem now is how to convert these measurements on the image plane into the appropriate joint angles. Basically, we have to determine the head kinematics and jacobian[2], which relate the target position and velocity with the appropriate joint positions and velocities :

$$\boldsymbol{\theta} = \boldsymbol{T_K^{-1}}(\boldsymbol{X_l}, \boldsymbol{X_r}) \tag{6.23}$$

where $\boldsymbol{\theta}$ denotes the joint angles, T_K^{-1} the inverse kinematics and X_l, X_r the target position in the left and right image planes.

The kinematics can be simplified if we assume that the vergence and tilt axes intersect in the cameras optic centers. In fact, this was the reason why the head design accounted for this possibility. Still, the kinematics are complex and, above all, depend largely on a number of parameters of the stereo head/cameras geometry which are hard to calibrate

[2]The kinematic and jacobian relate 2D image positions and velocity to the joint-space angular positions and speeds. In this sense, they differ from the usual robotics applications, which relate 3D coordinates and velocities to the joint space.

in practice.

Alternatively, we prefer to use a simple approximation of the kinematics since any inaccuracy can be compensated for in the feedback control loop. During smooth pursuit movements, the target position is stabilized in the center of the retina. When the target deviation from the image center is small, the relationship between the vertical and horizontal disparities and the corresponding tilt and vergence angles, can be decoupled.

In this situation, the relationship between the vergence and tilt angles and the target position in the left and right image planes, (x_l, y_l, x_r, y_r), results from simple geometric considerations as illustrated in Figure 6.10.

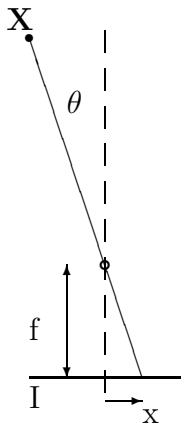

Figure 6.10: Simplified kinematics. The relationship between the target position in the image and its angular position, with respect to the vergence and tilt joints.

For the vergence, we have

$$\theta_{l,r} = \arctan \frac{x_{l,r}}{f_x} \tag{6.24}$$

where f_x is the camera focal length expressed in pixels. The tilt error can be calculated similarly, by averaging the two vertical target projections :

$$\theta_t = \arctan \frac{y_l + y_r}{2f_y}. \tag{6.25}$$

It was mentioned before that the saccadic and smooth pursuit eye movements have been implemented in the stereo head control system. Saccades are performed using position control in the servo loop with the joint angles described in equations (6.24) and (6.25).

As a consequence, the direction of gaze is redirected as rapidly as possible towards a new target or point of interest.

Alternatively, to track a slowly moving target, the smooth pursuit mechanism is accomplished using velocity control (integral velocity control mode). The required angular velocity command, to be applied to a specific joint, is estimated by dividing the position error by the sampling time of the overall visual loop (image acquisition and processing), and used to command the motors.

The visual controller (see Figure 6.9) is a simple PID controller. We have performed tests using a variety of parameters. Naturally, the pure delay existing in the loop, due to the visual processing poses a challenge to the control system. More sophisticated control strategies are now under consideration and development.

6.5.3 Coordination

There is a coordination problem to address regarding the pan and vergence degrees of freedom. In fact, the horizontal component of the disparity can be compensated either by verging/diverging the cameras, or by performing a pan movement. We have envisaged three main possibilities, to combine vergence and pan movements :

(i) The target is followed using the vergence and tilt motors and, whenever the cameras reach an uncomfortable position, a fast pan movement is performed to redirect gaze to a more comfortable position, while vergence movements compensate the neck rotation; This idea is shown in Figure 6.11. The amplitude of the pan saccade[3] can be determined based on the vergence angles :

$$\theta_g = \arctan\left[\frac{2\tan\theta_{vl}\tan\theta_{vr}}{\tan\theta_{vl}+\tan\theta_{vr}}\right] \tag{6.26}$$

The gaze angle, θ_g, is brought back to 90^0 and the new vergence angles, θ'_{vl} and θ'_{vr}, are set symmetrically equal using the equations :

$$\theta'_{vl} \;=\; \arctan\left[\frac{\sin\theta_{vl}}{\sin(\theta_g-\theta_{vl})}\right]$$

[3]Even though the term saccade is usually applied to the eye movements, this pan movement can be seen as saccadic in the sense that it is performed as fast as possible without visual feedback during the movement. Simultaneously, we have saccadic and vergence eye movements.

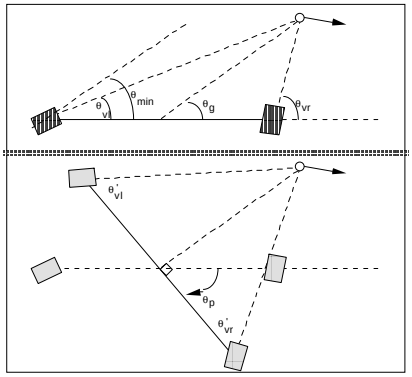

Figure 6.11: Fast movement (saccadic type) of the pan motor, followed by the appropriate vergence compensation.

$$\theta'_{vr} \quad = \quad \pi - \theta'_{vl} \tag{6.27}$$

(ii) The gaze angle is continuously kept at 90 degrees, by controlling the pan and tilt angles, while left and right vergence angles are kept symmetric, moving the fixation point closer or further from the camera;

(iii) A combined motion of the pan and vergence axes, based on the control effort (energy) and the comfort of the camera positions, is responsible for horizontally repositioning the target in the image center.

6.6 Results

An experiment has been carried out to test the overall response of the stereo head. The experiment setup is shown in Figure 6.12. We have used a light source moving at constant velocity in front of the stereo head. For this experiment, we have used a single axis (camera vergence) to track the moving spot.

When this experiment was made, the running frequency of the global visual loop is around 1.5 Hz, while the current setup runs at about 6Hz using all the degrees of freedom.

The camera is controlled in integral velocity mode, with the digital filter parameters, K and A, set according to the design analysis. The acceleration is 1.3 $\pi rad.s^{-2}$. See

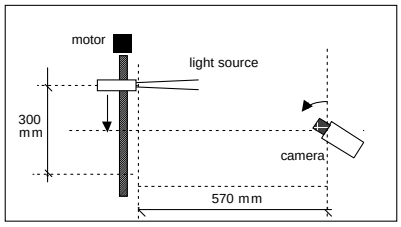

Figure 6.12: Experimental setup for the gaze control system.

section 6.4 for further details. We have tried different gaze controller parameters settings, as specified in Table 6.4. In order to illustrate the importance of the the minimum velocity that may be used, we have initially set the servo loop sampling period, T_s, at $1ms$. Therefore, the minimum velocity is expected to be $3^0/s$ in the vergence case.

Experiment	i	ii	iii	iv	v	vi
K_P	0.5	0.75	1.0	0.75	0.75	0.75
K_I	0.0	0.0	0.0	0.1	0.1	0.0
K_D	0.0	0.0	0.0	0.0	0.2	0.2

Table 6.4: PID settings for the experiments, with $T_s = 1\ ms$

Figure 6.13 shows different responses of the vergence system. The plots on the left side correspond to an increase of the proportional gain (from the top to the bottom). On the right hand side plots, we have introduced the integral action of the controller (top), then added a derivative term (center), and finally removed the integral part (bottom). The solid line shows the evolution of the target angular position, while the dotted line represents the evolution of the camera angular position. Both measurements are taken in the initial camera coordinate frame. It should be noticed that the target angular position is determined based on the image processing and therefore depends on the image resolution.

In general, the tracking capabilities shown are satisfactory. It is seen that the error is

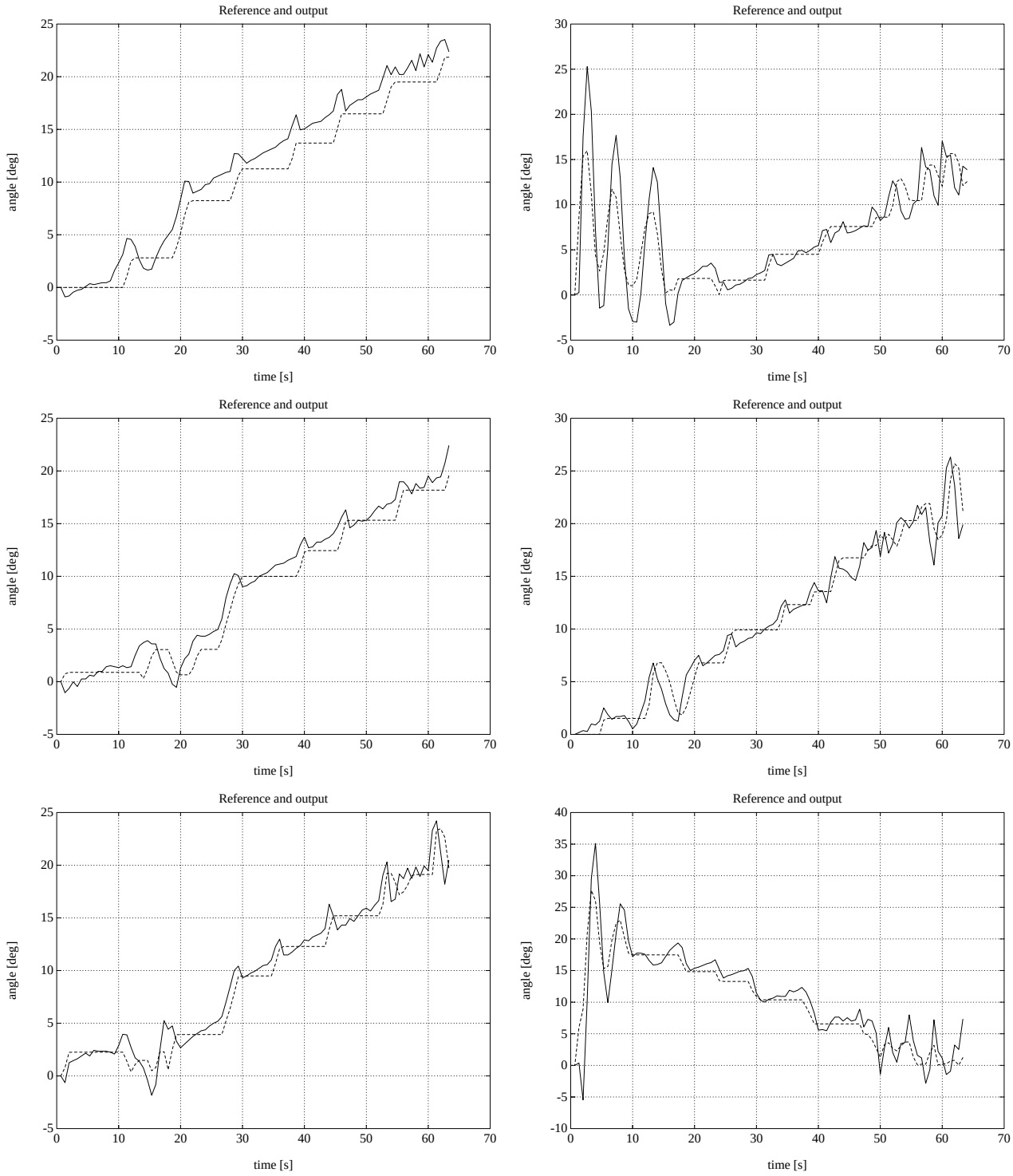

Figure 6.13: On the left column (experiments i, ii and iii), we see the effect of increasing the proportional gain of the gaze controller. On the right column, (experiments iv, v and vi) we start by adding some integrative action, then some derivative action, and finally remove the integral action.

reduced as the proportional gain increases. By introducing the integral effect, the error tends to zero (in average), although increasing the transient effects. This is mainly due to the addition of extra delay in the loop. By increasing the derivative gain, the system becomes more responsive and predictive, thus improving the overall system response, by increasing damping.

The existence of a pure delay of one sampling period for the vergence angles computation, is also clearly noticeable in the plots, and it is one of the effects that degrades the system response.

Another observation is that, for some periods, the camera is stalled in the same position while the error increases. This effect is mainly due to the fact that when the velocities are very small (under $3^0/s$) the encoder resolution is not sufficient to measure them, as explained in Section 6.4.

By increasing the sampling period to the proposed value of 2.048 ms, this minimum velocity is halved down to about $1.5^o/s$. The experiment was repeated with these settings. The two top plots on Figure 6.14 show two experiments with the same control settings but with different sampling frequencies. The tracking performance is clearly improved on the second plot, as the minimum velocity is lower.

To further improve these new results, one may check if the commanded velocity is less than the minimum achievable velocity. In this case, the vergence is controlled in position mode, thus achieving extremely high precisions, in a kind of microsaccades. If, otherwise, the command velocity is large enough, the controller uses the velocity control mode. The experiment was again repeated with this combined strategy and the results are shown on the bottom plot of Figure 6.14. The PID controller parameters are set at $K_p = 0.75$, $K_i = 0.1$ and $K_d = 0$. Note that a very accurate tracking behaviour was achieved. However, under this control strategy, the phase of gaze acquisition may be more difficult, due to the effect of pure position control.

Even though we have used a simple gaze control system, at the current stage, it was possible to illustrate the basic working mechanisms involved in a active gaze control applications. Particularly, much can be gained by a careful analysis of the tracking errors and choosing alternative control strategies, as it was done with the microsaccades. Accurate tracking capabilities were achieved. Further improvements in the control system are,

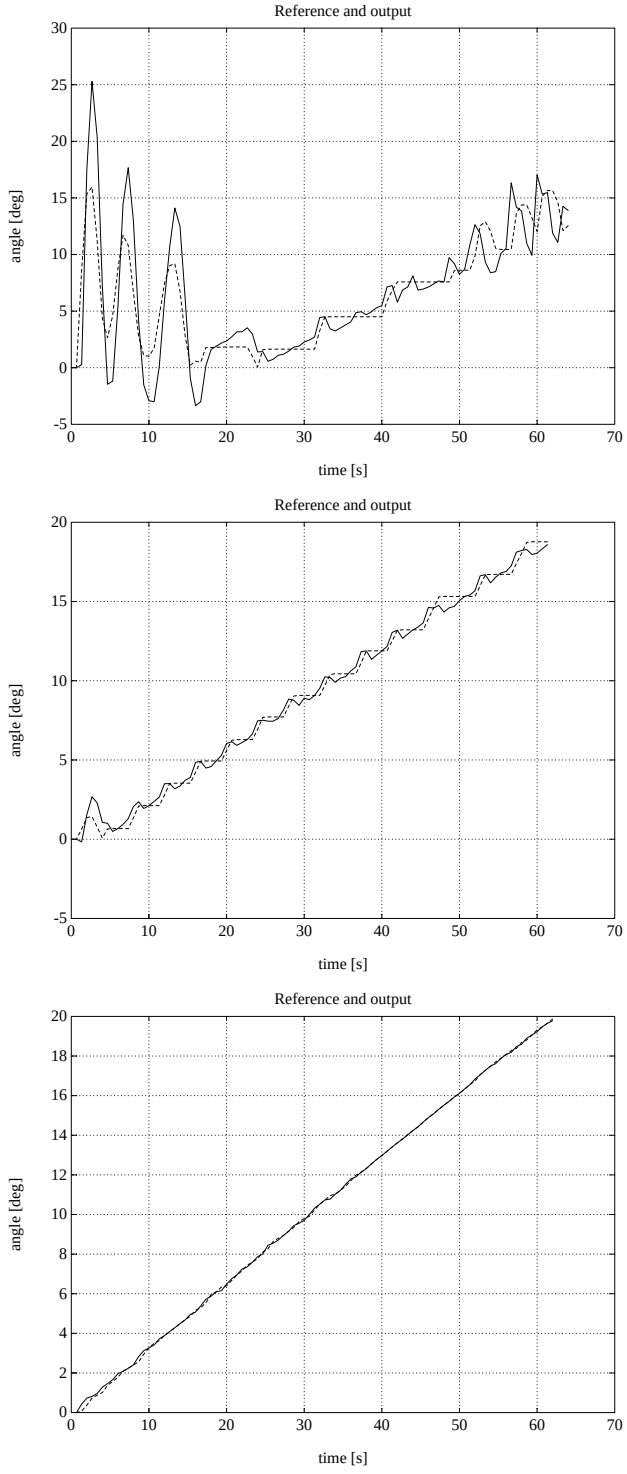

Figure 6.14: Tracking trajectories. From the first experiment to the second, we have doubled the sampling period, thus halving the minimum detectable speed to $1.5^0/s$. On the final experiment, the control strategy combines smooth pursuit and microsaccadic movements.

naturally, being envisaged.

6.7 Conclusions

We have presented **Medusa**, a stereo head for active vision applications, equipped with 4 degrees of freedom : independent vergence for each camera and a common tilt and pan. The inter-ocular distance can be manually adjusted. The camera position can be adjusted so that the tilt and vergence axes intersect (approximately) in the optical centers. We have set a number of specifications regarding the system performance, namely resolution, angular velocity and angular acceleration constraints.

The head control was split into two parts. The servo level was designed to ensure that each of the motors follows the position and velocity references, provided by the external loop. Particular care was taken in the design of these loops to ensure appropriate dynamic responses.

The visual information is then linked to the motor action by the gaze control system. In this outer loop, the visual processing identifies the target position and provides references to the servo loops in order to track the target. We have presented and discussed the main characteristics of biological oculomotor systems which provide a valuable help to the development of artificial stereo heads.

Finally, we have presented an example that illustrates the tracking capabilities of the system. The analysis of this experiment has revealed some control problems that suggested directions of improvement. The final control configuration led to low tracking errors.

Even though the gaze control system was kept to a simple design, good tracking performance was achieved. Further improvements are now being envisaged, namely in the gaze control system to cope with the pure delay introduced by the visual processing; and addressing the problem of gaze acquisition, which may require more advanced control design, than the problem of gaze holding.

Chapter 7

Summary and Conclusions

7.1 Summary

We have addressed in this thesis the problem of visual perception in the context of mobile robotics.

Chapter 1 was devoted to a general introduction to the problem of visual perception, discussing several approaches. Particularly, we have focused on the Gibson's approach to the psychology of visual perception, according to which perception is mainly a process of interacting with the environment in order to extract, active and selectively, the information relevant to a given purpose. On the other hand, in Marr's theory there is an important role played by internal representations of the external world. In this sense, visual perception consists on retrieving information from the world to build internal models which are then used for higher level cognitive procedures.

Marr's ideas led to the so called reconstructive approach in computer vision, aiming at recovering 3D information about scenes using images as the input. Gibson's ideas, instead, led to the purposive and qualitative active vision approach. Both methodologies are discussed in detail.

In **Chapter 2** we describe a vision system for 3D reconstruction using an image sequence as the input. It assumes a single camera moving in the environment (installed on a mobile robot or on a manipulator). The images are matched and the resultant disparities used to recover depth. In order to overcome the ill-posed character of the

matching process, we use prior models to constrain the disparity vector field to smooth solutions, in a regularization approach. The uncertainty on the disparity estimates is characterized and estimated. A kalman filter is used to integrate multiple measurements over time. We have presented several results in both land and underwater environments.

Chapter 3 describes an approach to autonomous navigation inspired on insect vision, *Robee*. Two lateral cameras are installed on top of a mobile robot and used to drive the vehicle along corridor like environments, by comparing partial descriptions of the peripheral flow fields. This approach does not require any reconstruction of the environment nor calibration. The robustness arises from the fact that visual measurements are directly coupled to the control of action, in a way similar to Gibson's direct perception approach. Another behaviour was implemented in which the robot can follow walls. Also, the vehicle speed is controlled based on visual input. Many examples were shown.

As *Robee* is blind in the direction of motion, in **Chapter 4** we have presented an approach to detect obstacles located ahead of the robot. The main assumption is that the robot is moving on a flat ground floor and, therefore, the camera is observing a planar surface in motion. This fact allows the description of the optical flow field by a parametric model. The information of the normal flow is used to robustly estimate the parameters of a simplified affine model. These parameters define an inverse perspective transformation between the image plane and the horizontal plane. Transforming the flow field in this way, simplifies the detection of obstacles lying above or below the ground floor. Real experiments are documented.

Docking is an important functionality for a mobile robot. It consists in approaching a specific point in the environment in a controlled way. We describe in **Chapter 5** two docking behaviours : ego-docking and eco-docking. In the ego-docking the robot carries an on board camera and docks to an external point in the environment. In the eco-docking, instead, the camera is installed on a docking station which controls the vehicle manoeuvres. In both cases the vehicle should approach the surface along the perpendicular direction and slow down until it stops. These behaviours are accomplished, again by using the normal flow to estimate the parameters of an affine motion model. These visual measurements are then used to drive the docking manoeuvres.

The active control of eye movements is an important feature in the visual system

of many animals. Also in robotic applications, we could profit from having a mobile vehicle equipped with an agile camera system, to actively explore the environment, follow moving targets, locate landmarks, etc. In **Chapter 6** we have described the design and control of an active camera head. This head has two cameras and four mechanical degrees of freedom. We have discussed the different types of eye movements and proposed a control system to implement them. Examples of tracking visual targets are presented and discussed.

7.2 Discussion

The work described in this thesis can be divided in two main parts. The reconstruction system produces depth maps which can be used for planning the robot motion between two points, recognition, object handling or simply to map the 3D structure of a given area of the environment (in marine science, having a 3D reconstruction of the seabed is often the goal). We have shown that through careful modeling of the different components of the system, we can obtain good results.

However, these maps require extensive computation and could hardly be used to safely drive the vehicle along a given path. Instead, the second part of the thesis proposes different perception/action functionalities for mobile robots, which stress the concept of visuo-motor coordination in, at least, two ways :

(i) The visual measurement used (normal flow of target position in the image) is elicited by the motion of the robot or movements of the eyes.

(ii) The perception/action loop is not decoupled in the sense that the performance of the perceptual processes is also a function of the control actions.

The consequences of this approach may be, in our opinion, very general particularly in the area of navigation and manipulation.

A purposive motor action coupled to a specific perceptual process directly elicits a behaviour (a behaviour emerges, in the sense of Brooks [Brooks, 1986a, Brooks, 1986b]), without the need for "understanding" the structure of the scene or continuously monitoring the geometric features of the environment. In doing that, the system behaves in a

parsimonious way by utilizing the minimum amount of information necessary to achieve the current goal (even if it is obvious, it is worth noting that only one goal at a time can be pursued and that, even in case of concurrent processes, the motor commands must be unique).

In Chapter 3, we used an approach (with a divergent stereo setup), for "centering" and "wall following" behaviours. The robot motion is uniquely controlled by the direct link between the normal flow estimation and the motor commands generated by the controller. Only the flow information from the peripheral part of the visual fields was used to maintain the robot in the center of a corridor [Santos-Victor et al., 1994a].

The same approach was again used in the docking experiments described in Chapter 5. Due to the intimate coupling of perception and action, there is no need to interpret the scene to elicit the behaviour : no matter what the robot "sees" it will end up in front of the "docking wall" and perpendicular to it.

In yet another experiment (see Chapter 4) the frontal part of the visual field has been used, extracting again the normal flow [Santos-Victor and Sandini, 1994], to detect obstacles and stop.

For all these visual behaviours it is not necessary to know the calibration and/or the vehicle motion parameters and, moreover, they are all based on the same visual information (optic flow). Two factors characterize the different behaviours :

(i) The part of the visual field analysed (on which part of the visual field is the attention focused).

(ii) The control law adopted (the direct link between visual information and rotation of the wheels).

An important step further arises if we add eye movements to the mobile robot, in order to actively exploit the environment. In the active gaze control described in Chapter 6, visual measurements were again used to control the action of an active observer, namely performing eye movements to keep a stable image of a moving target.

The challenge now is how to combine these different behaviours to accomplish more complex tasks. The simplest solution would be to design a "planner" eliciting the appropriate behaviour according to the current situation. For example, the centering behaviour

if the robot is navigating along a corridor or the wall following, the docking behaviour to stop in front of a door, the obstacle detection to avoid obstacles and search for landmarks or people, using the gaze control.

The problem, then, is no more to understand the environment (each behaviour embeds all the perceptual processes necessary to understand the relevant aspects of the environment) but to understand (or to know) the situation. Of course, this is not necessarily simpler than understanding the environment. However, the fact that it may not be necessary to "tune" a perceptual process, interpret the perceptual information and transform this into motor commands but, on the contrary, "appropriate action" is totally embedded inside the single behaviours, seems to be a very powerful way of breaking a complex problem into simpler ones and, consequently, of designing incremental systems whose capabilities are bounded by the number of behaviours implemented and do not require a general purpose architecture to be developed beforehand.

7.3 Directions for Future Work

Establishing the directions for future work, in the area of computer vision, can certainly be classified as an ill-posed problem mainly due to the enormous number of possible solutions.

A number of improvements can be pointed out in most of the problems addressed in the thesis. Regarding the visual reconstruction, an interesting improvement would consist on taking full advantage of controlling the camera motion using feedback from the estimated model. On the other hand, it would be challenging to determine to what extent we could use less exhaustive models or representations for navigation purposes. For instance, a representation describing features like corridors, walls, rooms, could possibly suffice for some applications in robotics.

An improvement already suggested for the docking behaviours consists on mounting a camera head on top of the robot and controlling the docking point simply by fixation and coupling the docking and gaze-control behaviours. The docking point would simply be specified in the image domain.

Concerning the stereo head, there are some directions of work to be pursued in the

near future. First, we are planning to use the normal flow measurements in order to track a general target. The vergence process can benefit from a different image sampling mechanism, namely space variant sampling strategies. Finally, we intend to add optical degrees of freedom and integrate all the different kinds of eye movements.

Another class of visual behaviours that we would like to address in the future are related to manipulation as this is often the purpose of having a mobile robot navigating through the environment. Also in this domain we intend to keep a similar approach regarding the interaction of perception and action.

Undoubtfully, the integration of all the different behaviours, so that more complex behaviours can emerge, is a big challenge. It also rises the question of using suitable mathematical tools to describe and manage the interaction between these behaviours.

Research on visual perception and the dream of actually building "seeing" systems has attracted, in the past, a number of different scientific communities like psychology, psychophysics, biology, engineering and computer science, which have contributed to the actual state of the art.

Certainly, in the future, this trend will persist and stimulate the development of this exciting endeavour, on achieving truly autonomous mobile agents.

Bibliography

[Abbott and Ahuja, 1988] Abbott, A. and Ahuja, N. (1988). Surface reconstruction by dynamic integration of focus, camera vergence, and stereo. In *Proc. of the 2nd. IEEE International Conference on Computer Vision*.

[Ahuja and Abbott, 1993] Ahuja, N. and Abbott, A. (1993). Active stereo : Integrating disparity, vergence, focus and calibration for surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1007–1029.

[Aloimonos, 1990] Aloimonos, Y. (1990). Purposive and qualitative active vision. In *Proc. of the 10th. IEEE International Conference on Pattern Recognition*, Atlantic City, NJ - USA.

[Aloimonos, 1993] Aloimonos, Y. (1993). Introduction : Active vision revisited. In Aloimonos, Y., editor, *Active Perception*. Lawrence Erlbaum Associates.

[Aloimonos, 1994] Aloimonos, Y. (1994). What I have learned. *CVGIP : Image Understanding*, 60(1):74–85.

[Aloimonos and Duvic, 1994] Aloimonos, Y. and Duvic, Z. (1994). Estimating the heading direction using normal flow. *International Journal of Computer Vision*, 13(1):33–56.

[Aloimonos and Shulman, 1989] Aloimonos, Y. and Shulman, D. (1989). *The integration of visual modules : an extension of the Marr paradigm*. Academic Press.

[Aloimonos et al., 1988] Aloimonos, Y., Weiss, I., and Bandophaday, A. (1988). Active vision. *International Journal of Computer Vision*, 1(4):333–356.

[Anandan, 1989] Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(4):283–310.

[Astrom and Wittenmark, 1986] Astrom, K. and Wittenmark, B. (1986). *Computer Controlled Systems: Theory and design*. Prentice-Hall.

[Ayache and Faverjon, 1987] Ayache, N. and Faverjon, B. (1987). Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2):107–131.

[Bajcsy, 1985] Bajcsy, R. (1985). Active perception vs. passive perception. In *Proceedings of IEEE workshop on Computer Vision*, pages 55–59, Bellair, MI.

[Bajcsy, 1988] Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, 76(8):996–1005.

[Ballard, 1991] Ballard, D. (1991). Animate vision. *Artificial Intelligence*, 48:57–86.

[Ballard and Brown, 1982] Ballard, D. and Brown, C. (1982). *Computer vision*. Prentice-Hall, London.

[Ballard and Brown, 1992] Ballard, D. and Brown, C. (1992). Principles of animate vision. *CVGIP : Image Understanding*, 56(1):3–21.

[Ballard et al., 1989] Ballard, D., Nelson, R., and Yamauchi, B. (1989). Animate vision. *Optics News*, 15(5):17–25.

[Bandopadhay et al., 1986] Bandopadhay, A., Chandra, B., and Ballard, D. (1986). Active navigation : Tracking an environmental point considered beneficial. In *Proc. of the IEEE Workshop on Visual Motion*, South Carolina, USA.

[Barron et al., 1994] Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–78.

[Bergen et al., 1992] Bergen, J., Anandan, P., Hanna, K., and Hingorani, R. (1992). Hierarchical model-based motion estimation. In *Proc. of the 2nd. European Conference on Computer Vision*, Santa Margherita, Italy.

[Bertero et al., 1988] Bertero, M., Poggio, T., and Torre, V. (1988). Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889.

[Blake and Yuille, 1992] Blake, A. and Yuille, A., editors (1992). *Active Vision*. MIT Press.

[Blake and Zisserman, 1987] Blake, A. and Zisserman, A. (1987). *Visual Reconstruction*. MIT Press.

[Brooks, 1986a] Brooks, R. (1986a). Achieving artificial intelligence through building robots. Technical Report 899, MIT - AI Lab.

[Brooks, 1986b] Brooks, R. (1986b). A robust layered control system for a mobile robot. *IEEE Transactions on Robotics and Automation*, 2:14–23.

[Brown, 1994] Brown, C. (1994). Toward general vision. *CVGIP : Image Understanding*, 60(1):89–91.

[Bruce and Green, 1985] Bruce, V. and Green, P. (1985). *Visual perception: physiology, psychology and ecology*. Lawrence Erlbaum Associates, Publishers.

[Carlsson and Eklundh, 1990] Carlsson, S. and Eklundh, J. (1990). Obstacle detection using model based prediction and motion parallax. In *Proc. of the 1st. European Conference on Computer Vision*, Antibes - France.

[Carpenter, 1988] Carpenter, R. (1988). *Movements of the eyes*. Pion, London.

[Christensen, 1991] Christensen, H. (1991). The AUC robot camera head. Technical Report LIA 91-17, Aalborg University, Laboratory of Image Analysis.

[Christensen et al., 1994] Christensen, H., Bowyer, K., and Bunke, H., editors (1994). *Active Robot Vision : camera heads, model based navigation and reactive control*, volume 6. World Scientific.

[Christensen and Madsen, 1994] Christensen, H. and Madsen, C. (1994). Purposive reconstruction : a reply to "A computational and evolutionary perspective on the role of representation in vision". *CVGIP : Image Understanding*, 60(1):102–108.

[Cipolla and Blake, 1992] Cipolla, R. and Blake, A. (1992). Surface orientation and time to contact from image divergence and deformation. In *Proc. of the 2nd. European Conference on Computer Vision*, Santa Margherita, Italy.

[Cipolla et al., 1993] Cipolla, R., Okamoto, Y., and Kuno, Y. (1993). Robust structure from motion using motion parallax. In *Proc. of the 4th. International Conference on Computer Vision*, Berlin, Germany.

[Coombs, 1991] Coombs, D. (1991). *Real-time gaze holding in binocular robot vision*. PhD thesis, University of Rochester.

[Coombs and Roberts, 1992] Coombs, D. and Roberts, K. (1992). Centering behaviour using peripheral vision. In Casasent, D., editor, *Intelligent Robots and Computer Vision XI: Algorithms, Techniques and Active Vision*, pages 714–21. SPIE, Vol. 1825.

[Crowley et al., 1992] Crowley, J., Bobet, P., and Mesrabi, M. (1992). Gaze control for a binocular camera head. In *Proc. of the 2nd. European Conference on Computer Vision*, Sta Margherita Ligure, Italy.

[Davis et al., 1983] Davis, L., Dunn, S., and Janos, L. (1983). Efficient recovery of shape from texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:485–492.

[Dhond and Aggarwal, 1989] Dhond, U. and Aggarwal, J. (1989). Structure from stereo : A review. *IEEE Transactions on Systems Man and Cybernetics*, 19(6):1489–1509.

[Doorn and Koenderink, 1983] Doorn, A. and Koenderink, J. (1983). The structure of the human motion detection system. *IEEE Transactions on Systems Man and Cybernetics*, 13(5):916–922.

[Edelman, 1994] Edelman, S. (1994). Representation without reconstruction. *CVGIP : Image Understanding*, 60(1):92–94.

[Enkelmann, 1990] Enkelmann, W. (1990). Obstacle detection by evaluation of optical flow field from image sequences. In *Proc. of the 1st. European Conference on Computer Vision*, pages 134–138, Antibes (France). Springer Verlag.

[Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326.

[Faugeras, 1992] Faugeras, O. (1992). What can we see in three dimensions with an uncalibrated stereo rig ? In *Proc. of the 2nd. European Conference on Computer Vision*, Santa Margherita, Italy.

[Faugeras, 1993] Faugeras, O. (1993). *Three Dimensional Computer Vision*. MIT Press.

[Faugeras et al., 1992] Faugeras, O., Luong, Q.-T., and Maybank, S. (1992). Camera self calibration : theory and experiments. In *Proc. of the 2nd. European Conference on Computer Vision*, Santa Margherita, Italy.

[Fermüller, 1993a] Fermüller, C. (1993a). Global 3D motion estimation. In *Proc. of the 4th. International Conference on Computer Vision*, Berlin, Germany.

[Fermüller, 1993b] Fermüller, C. (1993b). Navigational preliminaries. In Aloimonos, Y., editor, *Active Perception*. Lawrence Erlbaum Associates.

[Fermüller and Aloimonos, 1992] Fermüller, C. and Aloimonos, Y. (1992). Tracking facilitates 3-D motion estimation. *Biological Cybernetics*, 67:259–268.

[Ferrari et al., 1991] Ferrari, F., Fossa, M., Grosso, E., Magrassi, M., and Sandini, G. (June 1991). A practical implementation of a multilevel architecture for vision-based navigation. In *Proceedings of Fifth International Conference on Advanced Robotics*, pages 1092–1098, Pisa, Italy.

[Ferrier, 1991] Ferrier, N. (1991). The Harvard binocular head. Technical Report 9, Harvard Robotics Laboratory.

[Fischler, 1994] Fischler, M. (1994). The modeling and rpresentation of visual information. *CVGIP : Image Understanding*, 60(1):98–99.

[Fossa et al., 1992] Fossa, M., Grosso, E., Ferrari, F., Sandini, G., and Zapendouski, M. (1992). A visually guided mobile robot acting in indoor environments. In *Proc. of IEEE Workshop on applications of Computer Vision*, Palm Springs, U.S.A.

[Franceschini et al., 1991] Franceschini, N., Pichon, J., and Blanes, C. (June 1991). Real time visuomotor control: from flies to robots. In *Proceedings of the 5th Int. Conference on Advanced Robotics*, Pisa, Italy.

[Francisco, 1994] Francisco, A. (1994). *Active structure acquisition by continuous fixation movements.* PhD thesis, CVAP, Royal Institute of Technology, Stockholm, Sweden.

[Franklin et al., 1986] Franklin, G., Powell, J., and Naeini, A. (1986). *Feedback Control of Dynamic Systems.* Addison Wesley.

[Gaspar et al., 1994] Gaspar, J., Santos-Victor, J., and Sentieiro, J. (1994). Ground plane obstacle detection with a stereo vision system. In *Proceedings of the International Workshop on Intelligent Robotic Systems*, Grenoble.

[Gibson, 1950] Gibson, J. (1950). *The perception of the visual world.* Houghton-Mifflin.

[Gibson, 1958] Gibson, J. (1958). Visually controlled locomotion and visual orientation in animals. *British Journal of Psychology*, 49:182–194.

[Gibson, 1961] Gibson, J. (1961). Ecological optics. *Vision Research*, 1:253–262.

[Gibson, 1966] Gibson, J. (1966). *The senses considered perceptual systems.* Houghton-Mifflin.

[Gibson, 1979] Gibson, J. (1979). *The ecological approach to visual perception.* Houghton-Mifflin.

[Girosi et al., 1989] Girosi, F., Verri, A., and Torre, V. (1989). Constraints for the computation of optical flow. In *Proc. of the IEEE Workshop on Visual Motion*, pages 116–124.

[Grimson, 1981] Grimson, E. (1981). *From Images to surfaces : a computational theory of human stereo vision.* MIT Press.

[Grimson, 1984] Grimson, W. (1984). Computational experiments with a feature based algorithm. Technical Report 762, MIT - AI Lab.

[Grosso, 1993] Grosso, E. (1993). *Percezione binoculare e movimento: uno studio applicato alla robotica e all'intelligenza artificiale*. PhD thesis, DIST, Università di Genova, Italy.

[Grosso, 1994] Grosso, E. (1994). On perceptual advantages of eye-head active control. In *Proc. of 3rd European Conference on Computer Vision*, Stockholm, Sweden. Springer-Verlag.

[Grosso and Ballard, 1993] Grosso, E. and Ballard, D. (1993). Head-centered orientation strategies in animate vision. In *Proceedings of the 3rd International Conference on Computer Vision*, Berlin - Germany. IEEE Computer Society.

[Guissin and Ullman, 1989] Guissin, R. and Ullman, S. (1989). Direct computation of the focus of expansion from velocity field measurements. In *Proc. of the IEEE Workshop on Visual Motion*, pages 146–155.

[Hartley, 1992] Hartley, R. (1992). Estimation of relative camera positions for uncalibrated cameras. In *Proc. of the 2nd. European Conference on Computer Vision*, Santa Margherita, Italy.

[Heel, 1989] Heel, J. (1989). Dynamic motion vision. In *Proceedings of the DARPA Image Understanding Workshop*. Morgan-Kaufman Publishers.

[Horn, 1986] Horn, B. (1986). *Robot vision*. MIT Press.

[Horn and Brooks, 1989] Horn, B. and Brooks, M., editors (1989). *Shape from Shading*. MIT Press.

[Horn and Shunck, 1981] Horn, B. and Shunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

[Horridge, 1987] Horridge, G. (1987). The evolution of visual processing and the construction of seeing systems. *Proc. Royal Soc. London*, pages 279–292.

[Huang and Aloimonos, 1991] Huang, L. and Aloimonos, Y. (1991). Relative depth from motion using normal flow : an active and purposive solution. In *Proc. of the IEEE Workshop on Visual Motion*, N. Jersey, USA.

[Hummel and Sundareswaran, 1993] Hummel, R. and Sundareswaran, V. (1993). Motion parameter estimation from global flow field data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):459–476.

[Jain, 1994] Jain, R. (1994). Expansive vision. *CVGIP : Image Understanding*, 60(1):86–88.

[Jazwinski, 1970] Jazwinski (1970). *Stochastic processes and filtering theory*. Academic Press.

[Koenderink, 1986] Koenderink, J. (1986). Optic flow. *Vision Research*, 26(1):161–180.

[Koenderink and van Doorn, 1975] Koenderink, J. and van Doorn, J. (1975). Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791.

[Koenderink and van Doorn, 1991] Koenderink, J. and van Doorn, J. (1991). Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385.

[Krotkov, 1989] Krotkov, E. (1989). *Active Computer Vision by Cooperative Focus and Stereo*. Springer Verlag.

[Land, 1975] Land, M. (1975). Similarities in the visual behavior of arthropods and men. In Gazzaniga, M. and Blakemore, C., editors, *Handbook of Psychobiology*. Academic Press.

[Lee, 1976] Lee, D. (1976). A theory of visual control of braking based on the information about the time to collision. *Perception*, 5:437–459.

[Lehrer et al., 1988] Lehrer, M., Srinivasan, M., Zhang, S., and Horridge, G. (1988). Motion cues provide the bee's visual world with a third dimension. *Nature*, 332(6162):356–357.

[Lenz and Tsai, 1988] Lenz, R. and Tsai, R. (1988). Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:713–720.

[Little and Verri, 1989] Little, J. and Verri, A. (1989). Analysis of differential and matching methods for optical flow. In *Proc. of the IEEE Workshop on Visual Motion*, pages 173–180.

[Longuet-Higgins, 1981] Longuet-Higgins, H. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135.

[Mallot et al., 1991] Mallot, H., Bulthoff, H., Little, J., and Bohrer, S. (1991). Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64:177–185.

[Marr, 1982] Marr, D. (1982). *Vision*. W.H. Freeman.

[Marr and Poggio, 1979] Marr, D. and Poggio, T. (1979). A computational theory of human stereo vision. *Proc. Royal Society of London*, B(204):301–328.

[Martinuzzi and Questa, 1993] Martinuzzi, E. and Questa, P. (1993). Calcolo del flusso ottico e del tempo all'impatto con un sensore visivo spazio-variante. Tesi di Laurea in Ingegneria Elettronica, Universitá degli Studi di Genova.

[Mathies et al., 1989] Mathies, L., Kanade, T., and Szelisky, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(4):209–238.

[Maver and Bajcsy, 1993] Maver, J. and Bajcsy, R. (1993). Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5).

[Micheli et al., 1988] Micheli, E. D., Sandini, G., Tistarelli, M., and Torre, V. (1988). Estimation of visual motion and 3d motion parameters from singular points. In *Proc. of IEEE Int. Workshop on Intelligent RObots and Systems*, Tokyo, Japan.

[Moravec, 1977] Moravec, H. (1977). Towards automatic visual obstacle avoidance. In *Proc. of the 5th IJCAI*, page 584.

[Mundy and Zisserman, 1992] Mundy, J. and Zisserman, A., editors (1992). *Geometric Invariance in Computer Vision*. MIT Press.

[Murray, 1992] Murray, D. (1992). Stereo for gazing and converging cameras. Technical Report OUEL 1915/92, Robotics Research Group, Dept of Engineering Science, Univ of Oxford.

[Nagel, 1983] Nagel, H. (1983). Displacement vectors derived from second-order intensity variations in image sequence. *CVGIP*, 21:85–117.

[Nagel, 1987] Nagel, H. (1987). On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–323.

[Nagel and Enkelmann, 1986] Nagel, H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593.

[Negahdaripour and Lee, 1992] Negahdaripour, S. and Lee, S. (1992). Motion recovery from images sequences using only first order optical flow information. *International Journal of computer Vision*, 9(3):163–184.

[Nelson and Aloimonos, 1988] Nelson, R. and Aloimonos, J. (1988). Finding motion parameters from spherical motion fields (or the advantage of having eyes in the back of your head. *Biological Cybernetics*, 58:261–273.

[Nielsen and Sandini, 1994] Nielsen, J. and Sandini, G. (1994). Learning mobile robot navigation : a behavior based approach. In *Proc. of the IEEE International conference on Systems, Man and Cybernetics*, S. Antonio, Texas.

[Okutomi and Kanade, 1991] Okutomi, M. and Kanade, T. (1991). A multiple-baseline stereo. In *Proc. of the IEEE conference on Computer Vision and Pattern Recognition*, Hawaii.

[Omnit. Robotics, 1989] Omnit. Robotics (1989). *The MC-3000 Motion Controller : User Manual and programming guide*. Omnitech Robotics, USA.

[Otte and Nagel, 1994] Otte, M. and Nagel, H. (1994). Optical flow estimation: Advances and comparisons. In *Proc. of the 3rd. European Conference on Computer Vision*, Stockholm, Sweeden.

[Pahlavan, 1993] Pahlavan, K. (1993). *Active Robot Vision and Primary Ocular Processes*. PhD thesis, CVAP, Royal Institute of Technology, Stockholm, Sweden.

[Pahlavan and Eklundh, 1992] Pahlavan, K. and Eklundh, J. (1992). A head-eye system : Analysis and design. *CVGIP: Image Understanding*, 56(1):41–56.

[Pahlavan et al., 1993] Pahlavan, K., Uhlin, T., and Eklundh, J. (1993). Active vision as a methodology. In Aloimonos, Y., editor, *Active Perception*. Lawrence Erlbaum Associates.

[Pobuda and Erkelens, 1993] Pobuda, M. and Erkelens, C. (1993). The relation between absolute disparity and ocular vergence. *Biological Cybernetics*, 68:221–228.

[Poggio et al., 1985] Poggio, T., Torre, V., and Kock, C. (1985). Computational vision and regularization theory. *Nature*, 317:314–319.

[Pollard et al., 1981] Pollard, S., Mayhew, J., and Frisby, J. (1981). PMF : a stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470.

[Robinson, 1968] Robinson, D. (1968). The oculomotor control system: A review. *Proceedings of the IEEE*, 56(6).

[Rosenfeld, 1984] Rosenfeld, A. (1984). *Multiresolution Image Processing and Analysis*. Springer-Verlag, New York.

[Sandini et al., 1993a] Sandini, G., Gandolfo, F., Grosso, E., and Tistarelli, M. (1993a). Vision during action. In Aloimonos, Y., editor, *Active Perception*. Lawrence Erlbaum Associates.

[Sandini and Grosso, 1994] Sandini, G. and Grosso, E. (1994). Why purposive vision. *CVGIP: Image Understanding*, 60(1):109–112.

[Sandini et al., 1993b] Sandini, G., Santos-Victor, J., Curotto, F., and Garibaldi, S. (1993b). Robotics bees. In *Proceedings of IROS 1993*.

[Sandini and Tistarelli, 1990] Sandini, G. and Tistarelli, M. (1990). Robust obstacle detection using optical flow. In *Proc. of the IEEE Intl. Workshop on Robust Computer Vision*, pages 396–411, Seattle, (WA).

[Santos-Victor and Sandini, 1994] Santos-Victor, J. and Sandini, G. (1994). Uncalibrated obstacle detection using normal flow. *submitted to Machine Vision and Applications*.

[Santos-Victor et al., 1993] Santos-Victor, J., Sandini, G., Curotto, F., and Garibaldi, S. (1993). Divergent stereo for robot navigation: Learning from bees. In *IEEE International Conference on Computer Vision and Pattern Recognition - CVPR93*, New York.

[Santos-Victor et al., 1994a] Santos-Victor, J., Sandini, G., Curotto, F., and Garibaldi, S. (1994a). Divergent stereo in autonomous navigation : From bees to robots. *International Journal of Computer Vision*.

[Santos-Victor and Sentieiro, 1992a] Santos-Victor, J. and Sentieiro, J. (1992a). A 3D vision system for underwater vehicles: an extended Kalman-Bucy filtering approach. In *NATO Advanced Studies Institute - Acoustic Signal Processing for Ocean Exploration*, Madeira, Portugal. Kluwer Academic Press.

[Santos-Victor and Sentieiro, 1992b] Santos-Victor, J. and Sentieiro, J. (1992b). Generating 3D dense depth maps by dynamic vision : an underwater application. In *Proc. of the British Machine Vision Conference*, Leeds, UK.

[Santos-Victor and Sentieiro, 1993] Santos-Victor, J. and Sentieiro, J. (1993). Image matching for underwater 3D vision. In *International Conference on Image Processing: Theory and Applications*, San Remo, Italy.

[Santos-Victor et al., 1994b] Santos-Victor, J., van Trigt, F., and Sentieiro, J. (1994b). Medusa : A stereo head for active vision. In *Proceedings of the International Workshop on Intelligent Robotic Systems*, Grenoble.

[Schwartz, 1977] Schwartz, E. (1977). Spatial mapping in the primate sensory projection : Analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194.

[Sinclair et al., 1994] Sinclair, D., Blake, A., and Murray, D. (1994). Robust estimation of egomotion from normal flow. *International Journal of Computer Vision*, 13(1):57–70.

[Srinivasan, 1992] Srinivasan, M. (1992). Distance perception in insects. *Current Directions in Psychological Science*, 1:22–26.

[Srinivasan et al., 1991] Srinivasan, M., Lehrer, M., Kirchner, W., and Zhang, S. (1991). Range perception through apparent image speed in freely flying honeybees. *Visual Neuroscience*, 6:519–535.

[Subbarao and Waxman, 1986] Subbarao, M. and Waxman, A. (1986). Closed form solutions to image flow equations for planar surfaces in motion. *Computer Vision Graphics and Image Processing*, 36:208–228.

[Sundareswaran, 1991] Sundareswaran, V. (1991). Egomotion from global flow field data. In *Proc. of the IEEE Workshop on Visual Motion*, Princeton, New Jersey.

[Sundareswaran, 1992] Sundareswaran, V. (1992). A fast method to estimate sensor translation. In *Proc. of the 2nd. European Conference on Computer Vision*, Sta Margherita Ligure, Italy.

[Sundareswaran et al., 1994] Sundareswaran, V., Bouthemy, P., and Chaumette, F. (1994). Active camera self-orientation using dynamic image parameters. In *Proc. of the 3rd. European Conference on Computer Vision*, Stockholm, Sweeden.

[Szeliski, 1987] Szeliski, R. (1987). Regularization uses fractal priors. In *Proc. AAAI-87*, pages 271–301, Seattle, WA.

[Szeliski, 1990] Szeliski, R. (1990). Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 3(5):271–301.

[Tarr and Black, 1994a] Tarr, M. and Black, M. (1994a). A computational and evolutionary perspective on the role of representation in vision. *CVGIP : Image Understanding*, 60(1):65–73.

[Tarr and Black, 1994b] Tarr, M. and Black, M. (1994b). Response to replies. Reconstruction and purpose. *CVGIP : Image Understanding*, 60(1):113–118.

[Terzopoulos, 1986a] Terzopoulos, D. (1986a). Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):129–139.

[Terzopoulos, 1986b] Terzopoulos, D. (1986b). Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424.

[Tikhonov and Arsenin, 1977] Tikhonov, A. and Arsenin, V. (1977). *Solution of ill-posed problems*. Washington DC: Winston.

[Tistarelli and Sandini, 1993] Tistarelli, M. and Sandini, G. (1993). On the advantages of polar and log-polar mapping for direct estimation of the time-to-impact from optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):401–411.

[Trigt et al., 1993] Trigt, F., Santos-Victor, J., and Sentieiro, J. (1993). Medoesa : Design and construction of a stereo head for active vision. Technical Report rpt/07/93, VISLAB/ISR, Instituto Superior Técnico.

[Tsai, 1986] Tsai, R. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In *IEEE International Conference on Computer Vision and Pattern Recognition - CVPR86*, Miami Beach, Florida.

[Tsotsos, 1994] Tsotsos, J. (1994). There is no one way to look at vision. *CVGIP : Image Understanding*, 60(1):95–97.

[Uras et al., 1988] Uras, S., Girosi, F., Verri, A., and Torre, V. (1988). Computational approach to motion perception. *Biological Cybernetics*, 60:69–87.

[Verri et al., 1989] Verri, A., Girosi, F., and Torre, V. (1989). Mathematical properties of the 2d motion field : from singular points to motion properties. In *Proc. of the IEEE Wrorkshop on Visual Motion*, pages 190–200.

[Wallace et al., 1994] Wallace, R., Ong, P., Bederson, B., and Schwartz, E. (1994). Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90.

[Wann et al., 1993] Wann, J., Edgar, P., and Blair, D. (1993). Time-to-contact judgement in the locomotion of adults and preschool children. *Journal of Experimental Psychology : Human perception and performance*, 19(5):1053–1065.

[Warren and Hannon, 1990] Warren, W. and Hannon, D. (1990). Eye movements and optical flow. *Journal of Optical Society of America*, 7(1):160–169.

[Witkin, 1980] Witkin, A. (1980). *Shape from contour*. PhD thesis, MIT - Departement of Psychology.

[Yarbus, 1967] Yarbus, A. (1967). *Eye movements and vision*. Plenum, New York.

[Yeshurun and Schwartz, 1989] Yeshurun, Y. and Schwartz, E. (1989). Cepstral filtering on a colunar image architecture : a fast algorithm for binocular stereo segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):759–767.

[Yioshikawa, 1990] Yioshikawa, T. (1990). *Foundations of Robotics : Analysis and control*. MIT Press.

[Yuille and Geiger, 1990] Yuille, A. and Geiger, D. (1990). Stereo and controlled eye movement. *International Journal of Computer Vision*, 4:141–152.

[Zielke et al., 1990] Zielke, T., Storjohann, K., Mallot, H., and Seelen, W. (1990). Adapting computer vision systems to visual environment: Topographic mapping. In *Proc. of the 1st. European Conference on Computer Vision*, Antibes, France.