Visual Behaviors for Docking *

José Santos-Victor

Instituto Superior Técnico
Instituto de Sistemas e Robótica
Av. Rovisco Pais, 1
1096 Lisboa Codex - Portugal
e-mail: jasv@isr.isr.ist.utl.pt

Giulio Sandini

DIST, Lira-Lab
University of Genova
Via Opera Pia, 13
16145 Genova - Italia
e-mail:giulio@vision.dist.unige.it

Abstract

This paper describes visual-based behaviors for docking operations in mobile robotics. Two different situations are presented : in the *ego-docking*, each robot is equipped with a camera, and the motion is 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 surface normal, and the approaching speed (slowing down during the maneuver).

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 first order time-space image derivatives, which can be estimated robustly and fast.

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.

*The authors wish to thank the anonymous reviewers for their helpful comments and suggestions, that helped to improve the quality and the presentation of the paper. The research described in this paper has been supported by the Special Projects on Robotics of the Italian National Council of Research and by the ESPRIT project VAP-II. A fellowship from a bilateral collaboration between Consiglio Nazionale delle Ricerche (CNR) and Junta Nacional de Investigação Científica e Tecnológica (JNICT) is gratefully acknowledged by José Santos-Victor.

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 Bajczy describing the peculiarity of Active Perception [3] through the papers on Active and Animate vision [4, 2] and, more recently, to the concept of Purposive Vision [1]. 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 behaviors where action is eliciting and simplifying the perceptual processes which, in turn, drive the action itself (the "vision during action" approach [9, 20]). 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 in this paper in spite of the fact that it is based on direct, iconic, visual information, is based on the powerful assumption of being 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 behavior. 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 behaviors addressed in this paper are docking strategies for indoor mobile robots. The robot desired behavior consists in approaching a surface, along the surface normal with controlled forward speed, until it finally stops.

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 maneuver 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 1.

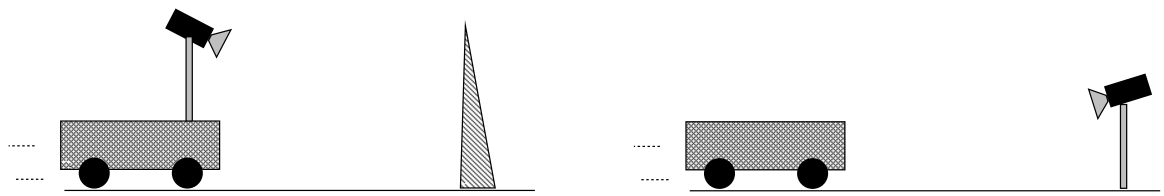

Figure 1: The left diagram shows the *ego-docking* where a robot, equipped with a camera and computing resources, docks to a surface. Instead, in the *eco-docking*, shown on the right, the camera is attached to a single docking station which may serve multiple robots.

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 behaviors 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 neighborhood of the docking station, the control system would take over and perform the maneuver.

To control the robot motion, we could use estimates of the optical flow field in the image. However, it is well known that under the hypothesis of image brightness constancy, it is only possible to determine the component of the optical flow in the direction of the image gradient, the normal flow. This structural limitation, known as the *aperture problem* [10, 11], has motivated the search of alternative methods and constraints on the optical flow allowing to recover both components of the optical flow [10, 14, 15, 16, 17, 19], which are often extremely complex and/or unstable[2].

However, the normal flow alone conveys sufficient information to many perception problems [1, 8] and can be estimated robustly and fast. In our case, the robot motion-controllers are solely driven by information of first order temporal-spatial image derivatives, from which the normal flow can be computed directly[3]. There is no need of imposing any type of smoothness constraints on the flow field.

Section 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. We determine the velocity of world points relative to the camera. The derivation presented leads to a unified description of both the *ego-docking* and *eco-docking*.

[2]Usually, these methods impose smoothness constraints on the optical flow field and perform some sort of regularization technique, or rely on second or higher order time-space image derivatives which tends to be a very ill-posed problem.

[3]Since the normal flow can be computed from the first-order time and space image derivatives, we will often refer either to these derivatives or to the normal flow, as being equivalent.

The core of Section 3 is dedicated to the analysis of planar surfaces in motion, which leads to a global second order parametric description of the flow field [24], and to the use of a simplified affine motion model [13, 12, 27] with improved performance [18] in what regards to the stability and robustness of the estimates. A procedure uniquely based on the first order time and space image derivatives, is proposed to estimate the affine motion parameters. We also address the problem of segmenting the part of the visual field which corresponds to the planar surface.

In Section 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 6 we present a real-time implementation of the behaviors described. We give a detailed description of the experimental tests, showing a robust performance and, finally, in the last section, we present some conclusions and establish further directions of work.

It should again be stressed that the use of visual measurements directly in the motion controller leads to improved robustness as the system is continuously monitoring its performance [7, 22, 23, 26]. There is no need to calibrate the camera and no reconstruction of the environment is performed.

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 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, there is the need 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 2. The robot motion is constrained to a

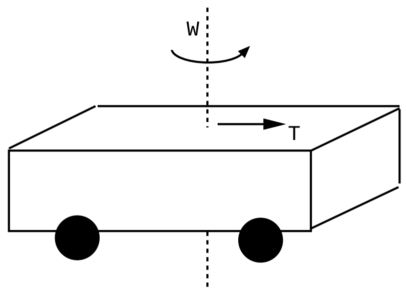

Figure 2: Mobile platform motion degrees of freedom.

forward speed T, and a rotation speed ω, around the vertical axis, achieved by differential control of two motors. Although this is a non-holonomic platform, it does not seem to be a major problem for the kind of trajectories generated by the docking behaviors.

In the following sections we will analyze the sensory-motor coordination for both the *ego-docking* and *eco-docking* problems.

2.1 Ego-docking

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

We assume that the camera coordinate frame, $\{C\}$, is translated with respect to the origin of the motor frame, $\{R\}$ and that, for simplification, this translation is solely along the $\hat{y}$ (forward) and $\hat{z}$ (upward) directions, as shown in Figure 3. Furthermore, $\{C\}$ is rotated relative to $\{R\}$, by a tilt angle, ψ, around the $\hat{x}$ axis, and a pan angle, ϕ, around the $\hat{y}$ axis.

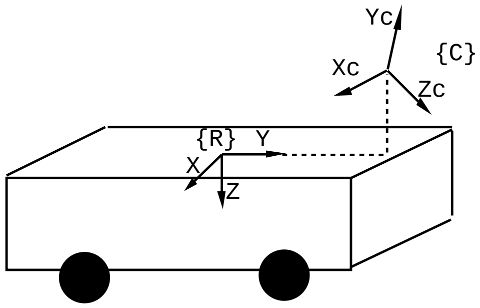

Figure 3: Sensor and motor coordinate coordinate frames, respectively {C} and {R}.

In many situations the camera points in the forward direction, aligned with the direction of the robot translational motion and, therefore, the pan angle $\phi \approx 0$. 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 behaviors. The constraint regarding the translation between the sensor and motor frames can be easily achieved in practice without the need for any specific calibration procedure, and is by no means critical.

The motion of the camera depends on the motion of the mobile platform, {R}, and can be expressed [28] in {C}, according to :

$$\begin{aligned} \mathbf{T_C} &= {}^{\mathbf{C}}\mathbf{R_R}(\mathbf{T_R} + \boldsymbol{\omega}_\mathbf{R} \times {}^{\mathbf{R}}\mathbf{P_{OC}}) \\ \boldsymbol{\omega}_\mathbf{C} &= {}^{\mathbf{C}}\mathbf{R_R}\, \boldsymbol{\omega}_\mathbf{R}\,, \end{aligned} \tag{1}$$

where $\mathbf{T_C}$ and $\boldsymbol{\omega}_\mathbf{C}$ stand respectively for the camera linear and angular velocities relative to a fixed world coordinate frame, expressed in the camera coordinate frame, respectively. Similarly, $\mathbf{T_R}$ and $\boldsymbol{\omega}_\mathbf{R}$ stand for the robot linear and angular velocities relative to a fixed frame, expressed in the robot coordinate frame. The term ${}^{\mathbf{C}}\mathbf{R_R}$ is the rotation matrix between {C} and {R}, and ${}^{\mathbf{R}}\mathbf{P_{OC}}$ is the position vector of the origin of {C} relative to {R}, expressed in the robot coordinate system, and it is assumed to have zero x component :

$${}^{\mathbf{R}}\mathbf{P_{OC}} = [0 \quad dy \quad dz]^T\,. \tag{2}$$

As explained, the rotation matrix ${}^{\mathbf{C}}\mathbf{R_R}$ describes the orientation of the camera frame relative to the robot and can be decomposed by a pan (ϕ) and tilt (ψ) contributions :

$$\begin{aligned}
{}^{\mathbf{C}}\mathbf{R}_{\mathbf{R}} &= \mathbf{Rot}(\hat{y},\phi)\mathbf{Rot}(\hat{x},\psi) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} .
\end{aligned} \tag{3}$$

The motion degrees of freedom of the mobile platform are constrained, as shown in Figure 2, to a pure rotation around the $\hat{z}$ axis and a pure translation along the $\hat{y}$ direction. Therefore, the robot linear and angular velocities can be simplified to :

$$\mathbf{T}_{\mathbf{R}} = [0 \quad T_{ry} \quad 0]^T \qquad\qquad \boldsymbol{\omega}_{\mathbf{R}} = [0 \quad 0 \quad \omega_{rz}]^T . \tag{4}$$

The motion of the camera depends on the robot motion and on the camera position relative to the robot. It can be written explicitly by combining equations (1) to (4) as :

$$\begin{aligned}
\mathbf{T}_{\mathbf{C}} &= \begin{bmatrix} -T_{ry}\sin\phi\sin\psi - \omega_{rz}dy\cos\phi \\ T_{ry}\cos\psi \\ -T_{ry}\cos\phi\sin\psi + \omega_{rz}d_y\sin\phi \end{bmatrix} , \\
\boldsymbol{\omega}_{\mathbf{C}} &= \begin{bmatrix} \omega_{rz}\sin\phi\cos\psi \\ \omega_{rz}\sin\psi \\ \omega_{rz}\cos\phi\cos\psi \end{bmatrix} .
\end{aligned} \tag{5}$$

In many situations, the camera does not have any mechanical degrees of freedom and, therefore, the rotation matrix, ${}^{\mathbf{C}}\mathbf{R}_{\mathbf{R}}$ is constant over time. Usually, in this case, the camera is pointing in forward direction, i.e. $\phi = 0$, and the camera motion equations can be written as :

$$\begin{aligned}
\mathbf{T}_{\mathbf{C}} &= [-\omega_{rz}dy \quad T_{ry}\cos\psi \quad -T_{ry}\sin\psi]^T , \\
\boldsymbol{\omega}_{\mathbf{C}} &= [0 \quad \omega_{rz}\sin\psi \quad \omega_{rz}\cos\psi]^T .
\end{aligned} \tag{6}$$

We have now determined the camera linear and angular velocities, which depend both on the robot motion and on the camera-robot geometric arrangement. If the camera motion is described by $\mathbf{T}_{\mathbf{C}}$ and $\boldsymbol{\omega}_{\mathbf{C}}$, then any static world point, ${}^{\mathbf{C}}\mathbf{P}$, is perceived with a linear velocity, relative to the camera, given by :

$${}^{\mathbf{C}}\mathbf{T}_{\mathbf{P}} = -\mathbf{T}_{\mathbf{C}} - \boldsymbol{\omega}_{\mathbf{C}} \times {}^{\mathbf{C}}\mathbf{P}, \tag{7}$$

which, using equations (1) leads to :

$$\begin{aligned} {}^{\mathbf{C}}\mathbf{T_P} &= -{}^{\mathbf{C}}\mathbf{R_R}(\mathbf{T_R} + \boldsymbol{\omega}_\mathbf{R} \times {}^{\mathbf{R}}\mathbf{P_{OC}}) - ({}^{\mathbf{C}}\mathbf{R_R}\,\boldsymbol{\omega}_\mathbf{R}) \times {}^{\mathbf{C}}\mathbf{P} \\ &= -{}^{\mathbf{C}}\mathbf{T_R} - {}^{\mathbf{C}}\boldsymbol{\omega}_\mathbf{R} \times ({}^{\mathbf{C}}\mathbf{P} - {}^{\mathbf{C}}\mathbf{P_{OR}}), \end{aligned} \tag{8}$$

where we have used the following equalities :

$$\begin{aligned} {}^{\mathbf{C}}\mathbf{T_R} &= {}^{\mathbf{C}}\mathbf{R_R}\mathbf{T_R} \\ {}^{\mathbf{C}}\boldsymbol{\omega}_\mathbf{R} &= {}^{\mathbf{C}}\mathbf{R_R}\boldsymbol{\omega}_\mathbf{R} \\ {}^{\mathbf{C}}\mathbf{P_{OR}} &= -{}^{\mathbf{C}}\mathbf{R_R}{}^{\mathbf{R}}\mathbf{P_{OC}} \end{aligned} \tag{9}$$

The projection of ${}^{\mathbf{C}}\mathbf{T_P}$ in the image plane is the motion field perceived by the camera while moving in the scene. Notice that if the camera and robot coordinate frames are coincident, then ${}^{\mathbf{R}}\mathbf{R_C}$ is the identity matrix, ${}^{\mathbf{C}}\mathbf{P_{OR}}$ is zero and this equation simply leads to equation (7).

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, serving several robots, which in turn do not need particular visual capabilities nor do require any specific computing power. Typically, a robot is commanded to go to the docking station based on odometric information alone [4] and, once in the vicinity of the docking station, the local behavior takes over and conduces the docking maneuver.

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 robot frame, {R}, and the sensor frame (docking station frame), {C}, depends on the instantaneous position and orientation of the mobile platform with respect to the docking station, as shown in Figure 4.

[4] which is known to be of limited precision

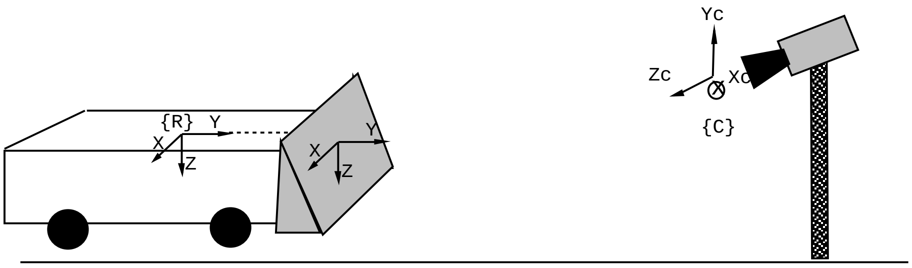

Figure 4: Coordinate systems involved in the eco-docking problem.

Let us now analyze how are these different coordinate systems related. Suppose that the camera is observing a point, $\mathbf{P}$, in the robot front panel, as the robot approaches the docking station. The linear velocity of this point relative to a fixed world frame, can be expressed in {R} as :

$$ {}^{\mathbf{R}}\mathbf{T_P} = \mathbf{T_R} + \boldsymbol{\omega}_\mathbf{R} \times \ {}^{\mathbf{R}}\mathbf{P} \ , \tag{10} $$

where ${}^{\mathbf{R}}\mathbf{P}$ denotes the position vector of the point in the robot front panel relative to the origin of {R}, expressed in {R}. The linear speed of any point in the robot front part results from a term due to the robot linear velocity, plus a term induced by the robot rotation. The coordinate transformation between frames {R} and {C} is given by :

$$ {}^{\mathbf{C}}\mathbf{P} = \ {}^{\mathbf{C}}\mathbf{R_R} \ {}^{\mathbf{R}}\mathbf{P} + \ {}^{\mathbf{C}}\mathbf{P_{OR}} \ . \tag{11} $$

where the rotation angles between {C} and {R} are now different, when compared to the ego-docking (see equation (3)) :

$$ {}^{\mathbf{C}}\mathbf{R_R} = \mathbf{Rot}(\hat{y}, \pi + \phi)\mathbf{Rot}(\hat{x}, \pi - \psi) \tag{12} $$

We are now able to express the linear velocity of $\mathbf{P}$, relative to a fixed world frame, in the camera coordinate system. It yields :

$$ \begin{aligned} {}^{\mathbf{C}}\mathbf{T_P} &= \ {}^{\mathbf{C}}\mathbf{R_R} \ (\ \mathbf{T_R} + \boldsymbol{\omega}_\mathbf{R} \times [\ {}^{\mathbf{R}}\mathbf{R_C} \ (\ {}^{\mathbf{C}}\mathbf{P} - \ {}^{\mathbf{C}}\mathbf{P_{OR}} \ )] \ ) \\ &= \ {}^{\mathbf{C}}\mathbf{T_R} + {}^{\mathbf{C}}\boldsymbol{\omega}_\mathbf{R} \times (\ {}^{\mathbf{C}}\mathbf{P} - \ {}^{\mathbf{C}}\mathbf{P_{OR}} \ ) \ , \end{aligned} \tag{13} $$

where ${}^{\mathbf{C}}\mathbf{P_{OR}}$ is the position vector of the origin of frame {R}, relative to {C} expressed in the camera coordinate system. We have used the same notation as in equation (9) to obtain the simplified form. Again, the image flow perceived in the camera image plane, results from projecting ${}^{\mathbf{C}}\mathbf{T}_{\mathbf{P}}$ in the image plane.

2.3 Discussion

We have started with the different setups for the eco-docking and ego-docking, and derived the equations that describe the motion of the observed world points relative to the camera. Comparing the eco-docking with the ego-docking, we see that both equations (8) and (13) have exactly the same structure.

The main difference stems mainly from the fact that in the ego-docking we use a camera centered motion frame, while in the eco-docking, the motion frame is object centered.

In the ego-docking, the camera orientation and position relative to the robot are fixed and, therefore, ${}^{\mathbf{C}}\mathbf{R}_{\mathbf{R}}$ and ${}^{\mathbf{C}}\mathbf{P}_{\mathbf{OR}}$ are constant over time. Instead, in the eco-docking, the robot position and orientation, relative to the camera, change continuously in time and ${}^{\mathbf{C}}\mathbf{R}_{\mathbf{R}}$ and ${}^{\mathbf{C}}\mathbf{P}_{\mathbf{OR}}$ are no longer constant. Even though this might be seen as a time varying system, posing hard difficulties to the control problem, we were able to achieve a stable and robust system behavior by using linear feedback control. The reason why, is due to the fact that the system dynamics are *smoothly* changing in time, while the robot performs the maneuver.

Equations (8) and (13) have opposite signs. This is due to the different orientation of the cameras in both docking cases, as seen by comparing equations (3) and (12).

Another aspect worth mentioning is the influence of robot rotation in the perceived image motion. Mathematically, this effect is given by a cross product between the angular velocity, ${}^{\mathbf{C}}\boldsymbol{\omega}_{\mathbf{R}}$, and the vector from the center of rotation to the point observed by the camera :

$$ {}^{\mathbf{C}}\mathbf{P} - {}^{\mathbf{C}}\mathbf{P}_{\mathbf{OR}} = {}^{\mathbf{C}}\mathbf{R}_{\mathbf{R}}{}^{\mathbf{R}}\mathbf{P}, \qquad (14) $$

where ${}^{\mathbf{R}}\mathbf{P}$ is the position vector of the point observed by the camera, relative to $\{\mathrm{R}\}$.

In the ego-docking, the magnitude of ${}^{\mathbf{R}}\mathbf{P}$ is the distance from the robot center to an observed world point, d_{ego}. On the other hand, in the eco-docking, ${}^{\mathbf{R}}\mathbf{P}$ is simply the vector from the robot center to the point in robot front panel observed by the camera, at a distance d_{eco} (comparable to the robot radius). In general d_{eco} is smaller than d_{ego} and, therefore, for the most part of the maneuver, the influence of the rotation in the image flow 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.

3 Planar surfaces in motion

We have seen previously how to determine the motion of an observed point relative to the camera, ${}^{\mathbf{C}}\mathbf{T_P}$, both in the ego-docking and eco-docking cases. The image flow perceived by the camera results from the projection of this velocity field in the image plane.

We will assume that the camera is viewing a planar surface in motion (the docking surface, for the *ego-docking*, or the robot front panel, for the *eco-docking*). With this assumption, we can obtain a globally valid description for the optical flow and derive an estimation procedure based on first order time-space image derivatives alone.

Consider equations (8) and (13) and define the following vectors :

$$
\begin{aligned}
{}^{\mathbf{C}}\mathbf{T_R} &= [T_x \;\; T_y \;\; T_z]^T \\
{}^{\mathbf{C}}\boldsymbol{\omega}_\mathbf{R} &= [\omega_x \;\; \omega_y \;\; \omega_z]^T \\
{}^{\mathbf{C}}\mathbf{P} - {}^{\mathbf{C}}\mathbf{P_{OR}} &= [\; X \quad Y \quad Z \;]^T
\end{aligned}
$$

The camera is modeled using the pinhole model, such that the projection of a 3D point with coordinates (X, Y, Z), in the image plane is given by :

$$
x = f_x \frac{X}{Z} + c_x \qquad\qquad y = f_y \frac{Y}{Z} + c_y \,, \tag{15}
$$

where f_x, f_y denote the camera focal length expressed in pixel dimensions and c_x, c_y denote the image center. Both in the ego-docking or eco-docking, the motion perceived in the image plane, by the camera, is given by the well known equations [5, 6, 24, 25] :

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

$$v(x,y) = f_y \left[\frac{\frac{y}{f_y} T_z - T_y}{Z(x,y)} + \omega_x (1 + \frac{y^2}{f_y^2}) - \omega_y \frac{xy}{f_x f_y} - \omega_z \frac{x}{f_x} \right] , \tag{16}$$

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

With the assumption that the camera is viewing a flat surface, a particular description of the optical flow can be obtained. The plane equation can be given by :

$$Z(X,Y) = Z_0 + \gamma_x X + \gamma_y Y , \tag{17}$$

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 projection in equation (17), and using the optical flow equations (16), one is led to the quadratic equations describing this planar motion [18, 24] :

$$\begin{aligned} u(x,y) &= u_0 + u_x x + u_y y + u_{xy} xy + u_{xx} x^2 \\ v(x,y) &= v_0 + v_x x + v_y y + v_{xy} xy + v_{yy} y^2 , \end{aligned} \tag{18}$$

which is a globally valid model for the optical flow, where the parameters are given by :

$$\begin{aligned} u_0 &= -f_x \left[\tfrac{T_x}{Z_0} + \omega_y \right] & v_0 &= f_y \left[-\tfrac{T_y}{Z_0} + \omega_x \right] \\ u_x &= \tfrac{T_z + \gamma_x T_x}{Z_0} & v_x &= \tfrac{f_y}{f_x} \left[\tfrac{T_y \gamma_x}{Z_0} - \omega_z \right] \\ u_y &= \tfrac{f_x}{f_y} \left[\tfrac{T_x \gamma_y}{Z_0} + \omega_z \right] & v_y &= \tfrac{T_z + \gamma_y T_y}{Z_0} \\ u_{xy} &= v_{yy} & v_{xy} &= \tfrac{-1}{f_x} (\tfrac{\gamma_x T_z}{Z_0} + \omega_y) \\ u_{xx} &= v_{xy} & v_{yy} &= \tfrac{1}{f_y} (-\tfrac{\gamma_y T_z}{Z_0} + \omega_x) \end{aligned} . \tag{19}$$

A straightforward approach, at this point, would consist in directly estimating the 8 parameters of this flow model. However, it has been shown analytically and experimentally in [18], that the second order parameters estimates can have a percentage error up to several orders of magnitude larger than the lower-order coefficients, even for perfect planar motion. If the angle of view is small and the depth of range limited, the second order parameters can be discarded and the motion of the planar surface approximated by an affine transformation. The modeling error (higher at the image periphery) is still 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 model [12, 13, 18, 27] :

$$\begin{aligned} u(x,y) &= u_0 + u_x x + u_y y \\ v(x,y) &= v_0 + v_x x + v_y y \, . \end{aligned} \tag{20}$$

3.1 Estimating the affine parameters

To estimate the affine parameters of the flow, the method we use relies solely on the first order spatio-temporal image derivatives. The first-order constraint for the optical flow computation, assuming the image brightness constancy over time [11], is given by :

$$uI_x + vI_y = -I_t \, , \tag{21}$$

where I_x, I_y and I_t stand for the partial derivatives of the image with respect to x, y, and time t. With this single constraint, we can only recover the normal flow, $u_\perp$, which is the component of the optical flow along the direction of the image gradient, which is the well known *aperture problem* [8, 10].

The normal flow can be computed directly from the first order space and time image derivatives, by :

$$u_\perp = \frac{I_t}{\sqrt{I_x^2 + I_y^2}} \, . \tag{22}$$

Our estimation procedure is directly based on the first-order space and time image derivatives. Using to equations (20) and (21), we can obtain a linear equation on the affine parameters, involving the first order temporal-spatial image derivatives :

$$\left[\begin{array}{cccccc} I_y & xI_y & yI_y & I_x & xI_x & yI_x \end{array} \right] \boldsymbol{\theta} = -I_t \, , \tag{23}$$

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

$$\boldsymbol{\theta} = [v_0 \;\; v_x \;\; v_y \;\; u_0 \;\; u_x \;\; u_y]^T \; . \tag{24}$$

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

The direct application of the least squares estimation procedure is quite sensitive to the presence of measurements severely affected by noise. In order to cope with this problem we developed a recursive estimation procedure aiming at eliminating the effect of outliers. The algorithm works as follows :

1. Chose randomly a set of data points $\{I_x, I_y, I_t\}$ to get an initial estimate, $\boldsymbol{\theta_0}$. Set $k = 1$.
2. Chose randomly a new set of data points $\{I_x, I_y, I_t\}$ such that the residue in equation (23), evaluated using the available parameter estimates, is small.
3. Estimate $\boldsymbol{\theta_k}$ based on the new data set. Set $k = k + 1$.
4. Proceed to step (2) until $\boldsymbol{\theta}$ remains unchanged or k exceeds a given number of iterations.

The initial estimate is based on a very large set of data points in the image which, in the limit case, could be all the available points with non-zero image gradient. Very often, some of these data points are corrupted by impulsive noise, thus leading to outliers in the measurements. The standard linear least squares estimates can be severely affected by these outliers.

The rationale of the algorithm presented here, is that one can improve this estimate by selecting a new set of data points that are "close" to the initial model, by verifying the residues in equation (23). In this way, the outliers will be discarded when computing a new estimate. This process can be repeated iteratively, as more and more outliers are rejected in the computation of $\boldsymbol{\theta}$. The non-linearity introduced by the rejection mechanism has allowed us to obtain good parameter estimates, even in the presence of a significant amount of outliers.

We have performed a wide variety of tests in order to verify the robustness of the estimation procedure in the presence of noise, using both synthetic and real flow fields.

Figure 5 shows a real image sequence acquired by a camera while moving towards a slanted poster. This is one of the test sequences used to estimate the image derivatives, and evaluate the estimation procedure.

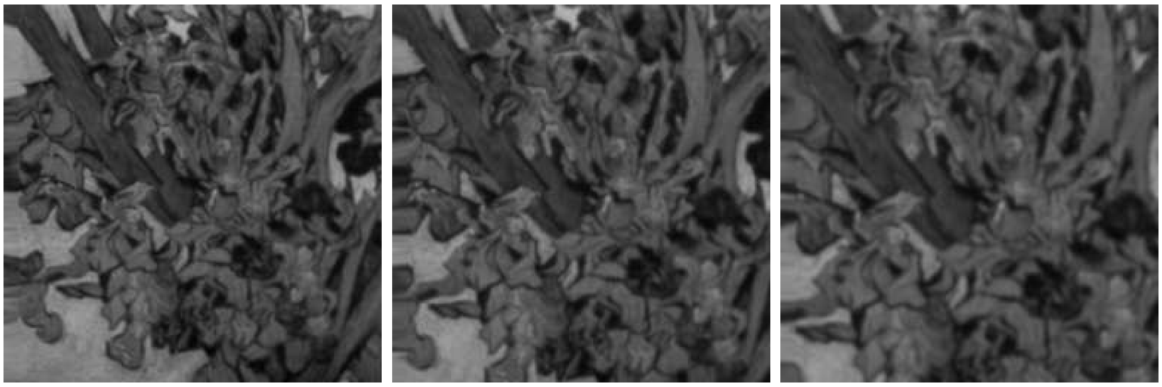

Figure 5: An image sequence used for the affine parameters estimation algorithm. It was acquired by moving a camera towards a slanted poster.

In Figure 6, we show an example of optical flow field computed on the image sequence. For presentation purposes, we deliberately show the 2D flow field, and not just the normal flow field, so that the outlier vectors can be easily detected. One can verify the existence of various significantly large vectors pointing in erroneous directions.

We have then estimated the affine parameters, as described previously. Having determined $\boldsymbol{\theta}$, we can now "reconstruct" the optical flow field, by using the affine model equations (20). The resultant 2D optical flow field is shown again for comparison in Figure 6. It is noticeable that the estimation procedure is able to retrieve important features of the input flow pattern such as the Focus of Expansion, and is not degraded by the presence of the outliers.

This robust performance was exhibited in the multiple tests we performed, and has proved to be quite important, when the total number of image samples is relatively small, as when we are processing low resolution images.

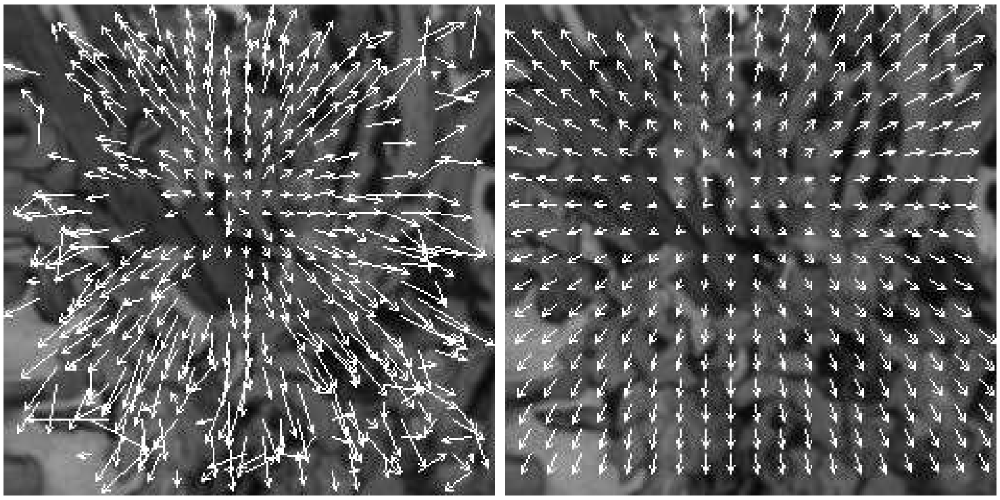

Figure 6: The left image shows the 2D optical flow estimated for the poster sequence. The right image shows the reconstructed flow field based on the estimated affine parameters.

3.2 Planar Surface Segmentation

The affine flow model presented in the previous sections is valid when a camera observes a planar surface in motion. The problem of how to segment this surface has still to be considered.

During the initial stage of an *ego-docking* maneuver, the camera visual field is fully occupied by the ground floor, and the planar approximation holds. As the robot moves ahead, the docking surface will enter the visual field and the planar model is no longer correct. Supposing that the robot is not disturbed by this, it continues to move and the docking surface will end up by covering the full field of view, and the planar approximation is again valid. What happens when the visual field is partially occupied by the docking surface? How do the affine parameters estimater behave, and how will the robot react?

Of course, one solution to the problem would be segmenting the docking surface, eventually with the intervention of an operator. However, segmenting the flow field is a difficult task and we would like to have an automatic procedure.

In our case, since we are using a planar surface approximation to describe the observed normal flow, intuitively we can expect the estimation procedure to *interpolate* between the two planes, when they are both present on the visual field, thus retrieving the parameters of a "mean" plane. The robot perception would gradually evolve from the perception of the ground plane to the perception of the docking plane, and the maneuver would complete successfully.

In order to investigate this hypothesis, and since calibrated experiments are difficult to obtain in practice, due to the non trivial dependencies between visual processing and control[5], we have done some simulations.

In the synthetic setup, we have a robot moving forward, with a camera pointing down at an angle of 45^0. The docking surface is vertical and is rotated 30^0, clockwise, with respect to the vertical axis (that is a slant of 30^0). When the robot starts moving ahead, it observes only the ground plane, and gradually, the docking surface will enter the visual field and end up by covering it completely.

We have synthesized the sequence of the corresponding flow fields which are shown in Figure 7. The first plot shows the flow field of the ground plane alone. The following plots, from left to right and top to bottom show the flow field partially corresponding to the horizontal ground floor and partially to the docking surface. Finally, in the last plot, the docking surface covers the full visual field. The line included in the plots shows the separation between both surfaces.

Using this sequence of flow fields, we have estimated the affine parameters of the flow. Since we know the camera intrinsic parameters, we could also compute the estimates of the plane tilt and slant angles describing the planar approximation to the surfaces visualized by the camera. Figure 8 shows the time evolution of these estimates. Figure 8 shows that there is a smooth change from the initial parameter estimates for the ground plane (0^0 of slant and 0^0 of tilt) and the final estimates for the docking surface (30^0 of slant and 90^0 of tilt), according to our expectations.

[5]Moreover, in our experimental situations, at the start of the docking maneuver, the "docking surface" occupied only a portion of the field of view and generated only a part of the flow vectors considered in the estimation procedure. Therefore we are rather confident that the results of the simulations, performed mainly to give a more precise description of the evolution of the planar estimate, are a good approximation of the real situation.

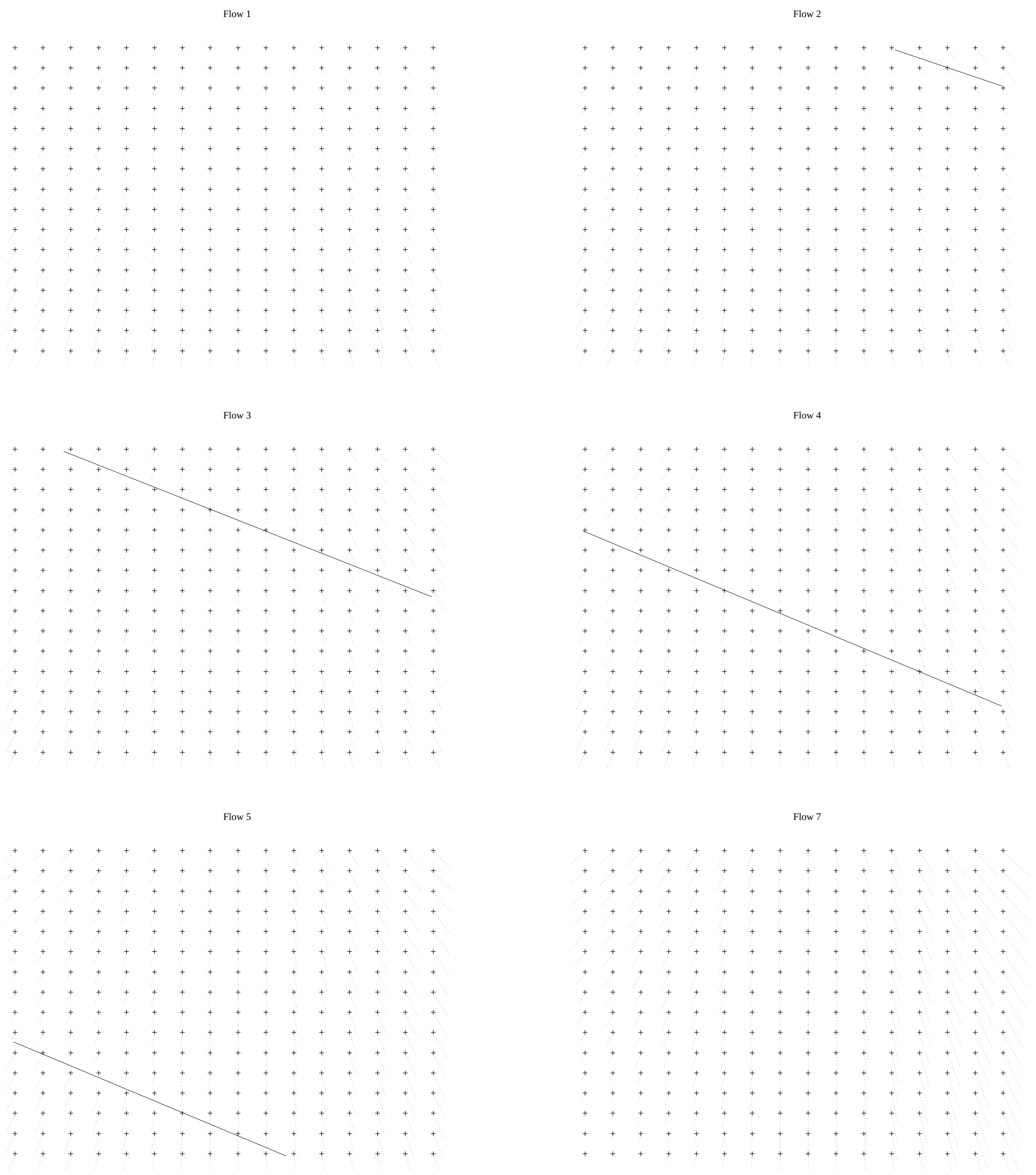

Figure 7: Evolution of the flow fields during the simulated experiment. In the top leftmost case, the robot is simply observing the ground plane, while in the bottom rightmost case it is only observing the vertical docking surface. As the robot moves, the portion of the visual field occupied by the docking surface increases. The line shows the separation between both planes.

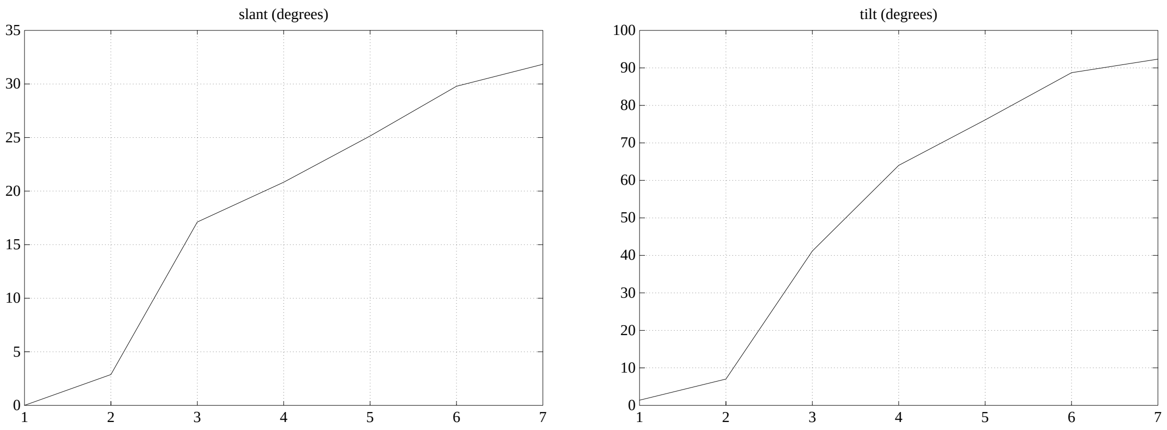

Figure 8: Evolution of the estimated tilt and slant angles as the robot approaches the vertical wall.

The small difference between the values shown in Figure 8 and the ground-truth values are due to the affine approximation to the flow field, instead of the full second-order model. However, the evolution of the orientation estimates during the full path is smooth and therefore, the robot should be able to orient itself with respect to the docking surface, without the need of explicitly segmenting the image.

Regarding the *eco-docking*, the problem is somewhat simpler, as we can assume that the robot is close to the docking station and, therefore, it is perceived as the only moving object in the scene. The desired segmentation is simply obtained by considering only the image points with non-zero flow.

4 Visual Based Control

Within this section, we will show how to use the parameters of the affine flow model to control the docking maneuver. The emphasis will be on the direct use of visual measurements to control motion and accomplish a given task. The visual input consists on the first order time-space image derivatives.

The coordination of perception and action results in an improved performance as the visual information is continuously being used to monitor the robot behavior. 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 maneuver. 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.

A point worth mentioning is that the control strategy we propose for the docking behavior 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 enters the visual field, will the control generate changes in the heading. One can say that the behavior is elicited by the visual information without any need for "higher order" computations, to launch the behavior.

4.1 Ego-docking behavior

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 image measurements (affine motion parameters) into the control variables (robot forward speed and angular speed), by relating the sensor (camera) and control (robot) coordinate frames. Considering $\phi = 0$, and using equation (6) together with equations (19), we have :

$$
\begin{array}{rclcrcl}
u_0 & = & f_x \left[\frac{dy}{Z_0} - \sin\psi\right] \omega_{rz} & \quad & v_0 & = & -f_y \frac{T_{ry}\cos\psi}{Z_0} \\
u_x & = & -\frac{T_{ry}\sin\psi + \gamma_x \omega_{rz} dy}{Z_0} & & v_x & = & \frac{f_y}{f_x}\left[\frac{T_{ry}\gamma_x\cos\psi}{Z_0} - \omega_{rz}\cos\psi\right] \\
u_y & = & \frac{f_x}{f_y}\left[\cos\psi - \frac{\gamma_y dy}{Z_0}\right]\omega_{rz} & & v_y & = & \frac{\gamma_y\cos\psi - \sin\psi}{Z_0} T_{ry}
\end{array} . \quad (25)
$$

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 moving at the same speed. In fact, v_0 consists in a ratio between the robot forward speed, T_{ry}, and the distance measured along the optical axis, Z_0. Hence, to adjust the speed of the docking maneuver, we can control T_{ry} in order to keep v_0 constant. The controller error signal is the difference between the observed v_0 and a nominal reference v_o^{ref}, corresponding to a desired cruise speed :

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

The robot speed is incrementally adjusted (thus introducing an integrator in the control chain) by means of a PID controller :

$$\Delta T_{ry}[n] = K_p^T \ \{ \ e[n] + K_i^T \sum_k e[n-k] + K_d^T (e[n] - e[n-1]) \ \} \ . \tag{26}$$

We will use v_x to control the robot heading direction. To align the camera axis perpendicularly to the visualized surface, the controller must regulate γ_x to zero (see equation (17)). Let us then consider a controller with the following structure:

$$\omega_{rz} = -K\gamma_x \ , \tag{27}$$

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

$$\begin{aligned} \omega_{rz} &= -\frac{f_x}{f_y} \frac{Z_0 K}{(T_{ry} + Z_0 K)\cos\psi} v_x \\ &= -\frac{G_\omega}{\cos\psi} v_x \ . \end{aligned} \tag{28}$$

Therefore, the regulation of v_x to zero accomplishes the goal of regulating γ_x to zero, thus orienting the robot perpendicularly to the surface. The angular velocity controller consists in a PID having v_x, as the error signal :

$$\omega_{rz}[n] = K_p^{\omega} \; \{ \; v_x[n] + K_i^{\omega} \sum_k v_x[n-k] + K_d^{\omega}(v_x[n] - v_x[n-1]) \; \} \; . \tag{29}$$

Figure 9 illustrates the overall structure of the controller.

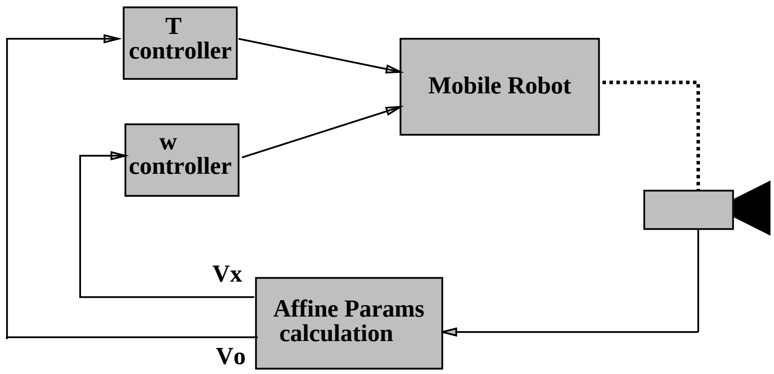

Figure 9: Structure of the overall control system.

4.2 Eco-docking

We have already shown that the sensory-motor equations for both the *ego-docking* and *eco-docking* are very similar. This unified approach allows us to use the same controller for both situations. In the absence of rotation, the visual parameters used in the control system are given by :

$$\begin{aligned} v_0 &= -f_y \frac{\cos\psi \; T_{ry}}{Z_0} \\ v_x &= -\frac{f_y}{f_x} \frac{\gamma_x \; \cos\psi \; T_{ry}}{Z_0} \, , \end{aligned} \tag{30}$$

which is **exactly** the same situation as in equations (25) except that there is an inversion on the rotation direction. Note that simply by 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.

The differences regarding the time changes in $^{\mathbf{C}}\mathbf{R}_{\mathbf{R}}$ and $^{\mathbf{C}}\mathbf{P}_{\mathbf{OR}}$ are accommodated by the control system during operation, while disturbances induced by the robot rotation [6] are less noticeable than in the ego-docking case.

5 Results

The system has been tested in real time using a TRC Labmate mobile platform and we have used a camera with a 8mm lens.

For the *ego-docking* problem, the camera was placed in the front part of the robot pointing at the ground plane with an angle ranging from 30 to 60 degrees, roughly aligned with the robot forward axis, as shown in Figure 10. In the various experiments done, we have had no particular care in the camera placement. The systems behaves equally well for a number different camera orientations and is very tolerant to misalignments. We have only to guarantee that the robot sees the docking surface at some point.

Figure 10: Experimental setup used for the *ego-docking* behavior.

In the *eco-docking*, we have attached the camera to a fixed position, pointing slightly down. Again, the system behaved equally well for a number of different camera orientations and the sole constraint is that the robot must enter the camera visual field, at some point. Figure 11 shows the setup used for the eco-docking experiments, where we can see the robot during a maneuver and the docking-station.

[6] Even when commanded to have pure translational motion, a real mobile platform will always exhibit some level of rotational disturbances.

Figure 11: *Eco-docking* behavior : the robot during the maneuver and the docking station.

The images are grabbed with a resolution of 128×128 pixels. The the first order time-space image derivatives are computed over a central window of 80×80 pixels. The reason for doing this is twofold. On one hand, there is the obvious computation gain by considering only a part of the visual field. On the other hand, the affine approximation to the flow of a planar surface, is worse towards the image periphery, when compared to the central image regions. The first-order derivatives are determined using a set of five images acquired at video rate, and are used to estimate the affine parameters, and synthesize the control laws.

At the current level of implementation, the system is running approximately at 1 Hz (sampling frequency of the visual loop) on a VDS Eidobrain image processing workstation. Figure 12 shows an image of the normal flow measured for the ground plane.

In the experiments performed we have tried various parameters for the linear and angular velocity PID controllers, and placed the robot in several initial positions relative to the docking surface, or to the docking station.

The system behavior is not too sensitive to the control parameters chosen. The final set is shown in Table 1. The output of the controllers are the increment command in the robot forward speed, in mm/s, and the rotation speed command, in deg/s. The order of magnitude of the rotation proportional gains, is mainly due to the small amplitude of v_x. As the linear velocity control loop has already a dynamic integrator, due to the incremental control strategy, we are not using the integral term of the PID controller,

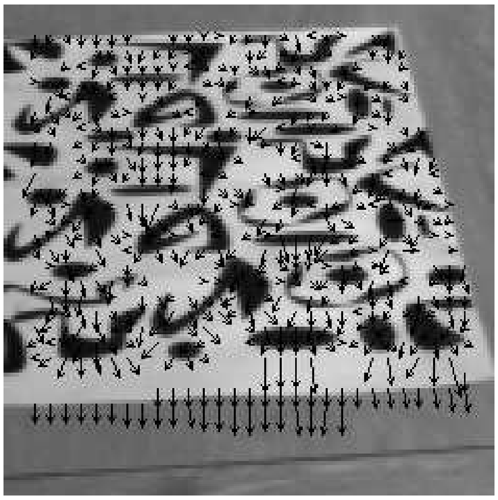

Figure 12: Sample of the normal flow field computed using the first order time and space image derivatives, used to estimate the affine motion parameters.

which would lead to oscillation and instability.

	K_p	K_i	K_d
ΔT	20	0	0.1
ω	2500	0.15	0.1

Table 1: PID controllers parameters used for the docking experiments. The sign of the rotation proportional gain must be inverted for the eco-docking case.

Several tests were made using both the *ego-docking* and *eco-docking* docking, starting the robot from different initial positions. Figure 13 describes a typical ego-docking experiment. It shows the real trajectory of the robot (recovered from odometry) during an ego-docking maneuver. Initially there is an angular difference of approximately 45^0 between the robot heading and the direction perpendicular to the docking surface. During the maneuver the robot describes a smooth trajectory and aligns the camera axis with the direction normal to the surface, while controlling the forward speed. Figure 14 shows an image sequence of the *ego-docking* maneuver as seen by the camera.

In Figure 15, we show the evolution of the forward speed and angular position (heading direction) of the robot during operation. Note how the velocity and orientation vary

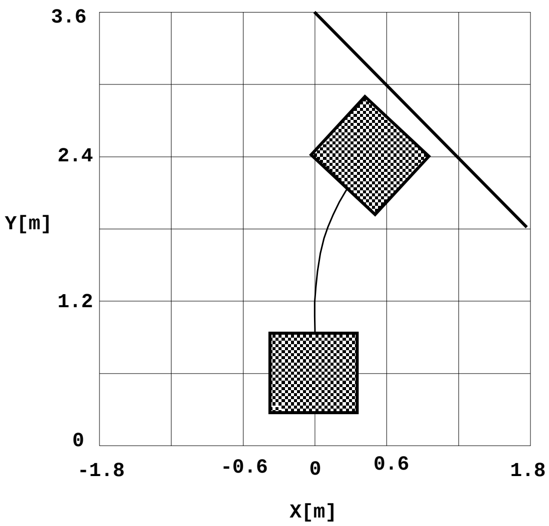

Figure 13: Robot trajectory during a real *ego-docking* maneuver. The trajectory is recovered using odometry.

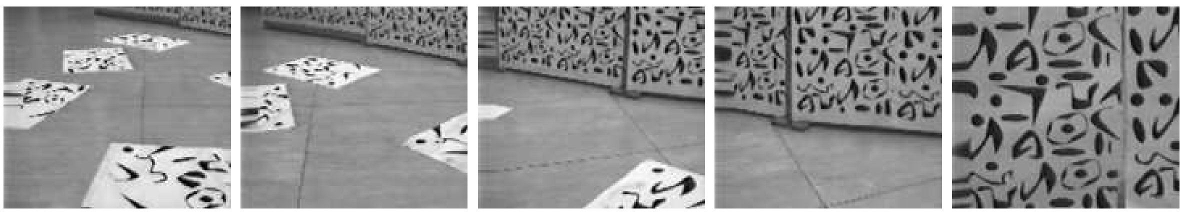

Figure 14: Image sequence seen by the camera during the *ego-docking* maneuver.

smoothly as the robot approaches the goal. It also shows that the robot orientation tends to the appropriate value. The robot stops within a few centimeters from the docking surface.

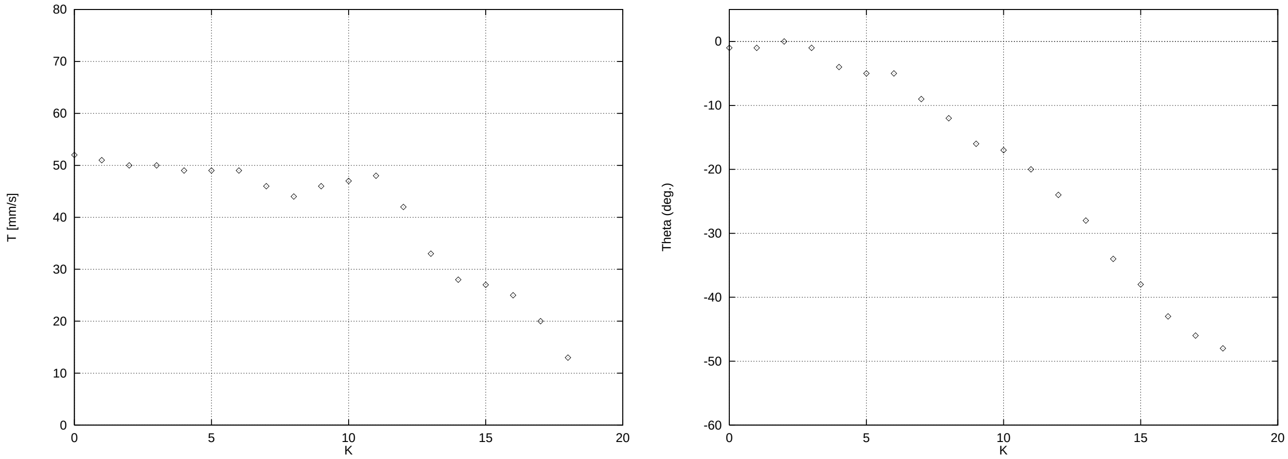

Figure 15: The left plot shows the evolution of the robot forward speed in $[mm/s]$, while the right plot shows the evolution of the heading direction in degrees, during the *ego-docking* maneuver, where k denotes the sampling instants.

Also in the eco-docking, the system has revealed a robust behavior and we have made several tests using different initial positions for the robot. The only constraint is that the robot must be visible from the docking station at some time during the maneuver. We have used the same controller apart from a sign inversion in the rotation control law. In Figure 16 we show the plot of the real robot trajectory (recovered using odometry) during a typical *eco-docking* experiment. Again, the robot described a smooth trajectory until reaching the final orientation and position.

Figure 17 shows an image sequence acquired during the *eco-docking* maneuver, as seen by the camera at the fixed docking station. Figure 18 shows the evolution of the robot forward speed and heading direction during the maneuver, smoothly changing over the full path. Again, the robot reached the desired final position with the desired orientation.

We can also see that while the robot is far from the docking station, the speed control loop originates an increase of the robot velocity until a cruise speed. Then, when the

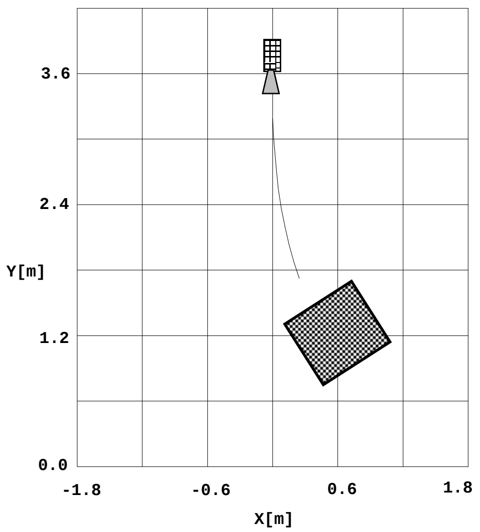

Figure 16: Robot trajectory during an *eco-docking* maneuver to a static docking station. The trajectory is recovered using odometry.

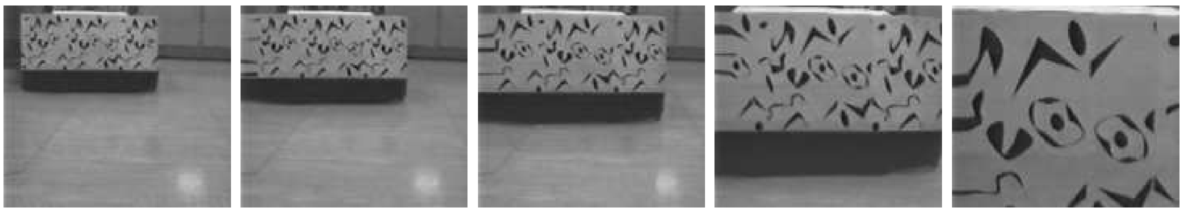

Figure 17: Image sequence seen by the camera during an *eco-docking* maneuver.

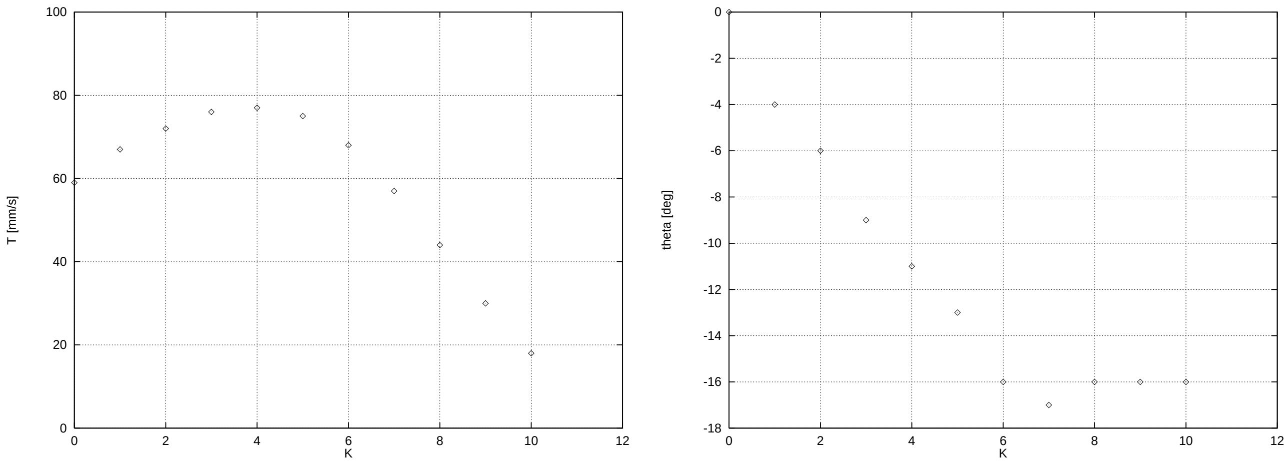

Figure 18: Eco-docking example. The left plot shows the evolution of the robot forward speed in $[mm/s]$, while the right plot shows the evolution of the heading direction in degrees, during the *eco-docking* maneuver, where k denotes the sampling instants.

robot gets closer to the docking station, the speed decreases until a final stop.

The angular and position errors in the maneuvers are in the range of a few degrees (typically up to 5^0) in orientation and a few centimeters (typically up to $5cm$) in the distance to the docking surface. These errors are mainly due to the low resolution of the images we use, to the relatively low sampling frequency, and to mechanical problems in the platform when the commanded speeds are very low.

The robust performance of the system in both docking situations is mainly due, in our opinion, to two kinds of reasons. On one hand, we have carefully analyzed all the different processing stages in the system, assuring a robust performance of each one of them. On the other hand, it results from the direct integration of the visual measurements in the visual loop, without any need to reconstruct the environment or calibrate the cameras.

6 Conclusions

The docking behaviors described in this paper, besides the peculiarities summarized below, stress the concept of visuo-motor coordination in, at least, two ways :

1. The visual measure used is elicited by the motion of the robot .
2. The perception/action loop is not decoupled in the sense that the performance of the perceptual processes is also a function of the control ones.

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 behavior (a behavior emerges as Brooks puts it), 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 the experiment presented here, for example, the behavior of the robot is solely controlled by the direct link between the affine flow parameters estimation and the motor commands generated by the controller: no matter what the robot "sees" it will end up in front of the "docking wall" and perpendicular to it. In another experiment [23], the same approach was used (with a divergent stereo set-up) for "centering" and "wall following" behaviors. In that case, only the flow information from the peripheral part of the visual fields was used to maintain the robot in the center of a corridor. In yet another experiment, the frontal part of the visual field has been used (extracting normal flow) to detect obstacles and stop [21]. For all these visual behaviors there is no need 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 behaviors:

1. The part of the visual field analyzed (in which part of the visual field is the attention focused on).

2. The control law adopted (the direct link between visual information and rotation of the wheels).

The challenge now is how to combine these different behaviors to accomplish more complex tasks. The simplest solution would be to design a "planner" eliciting the appropriate behavior according to the current situation. For example, the centering behavior if the robot is navigating along a corridor or the wall following or the docking behavior to stop in front of a door or the obstacle detection to avoid obstacles. The problem, then, is no more to understand the environment (each behavior embeds all the perceptual processes necessary to understand the aspects of the environment strictly necessary) 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 behaviors, 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 behaviors implemented and do not require a general purpose architecture to be developed beforehand.

More specifically we have presented an active vision approach for the problem of docking 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 is controlled during docking maneuver 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 maneuver). 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. As the visual input, we use directly the first order time and space image derivatives, which determine the normal flow. The approach does not rely on the estimation of the 2D optical flow.

An affine model is fitted to the measured motion field, and a fast estimation procedure robust to outliers was presented. The affine parameters of the flow field 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 to for the *ego-docking* or *eco-docking* cases, in spite of some differences which are discussed in the paper.

A real time implementation was realized and a robust docking behavior achieved, with examples given in the paper both in the *ego-docking* and *eco-docking* problems. A major issue is the fact that there is no need to calibrate the camera intrinsic or extrinsic parameters nor is it necessary to know the vehicle motion.

References

[1] Y. Aloimonos. Purposive and qualitative active vision. In *Proc. of the 10th. IEEE International Conference on Pattern Recognition*, Atlantic City, NJ - USA, June 1990.

[2] Y. Aloimonos, I. Weiss, and A. Banddophaday. Active vision. *International Journal of Computer Vision*, 1(4):333–356, January 1988.

[3] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, August 1988.

[4] D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

[5] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Proc. of the 2nd. ECCV*, Santa Margherita, Italy, May 1992.

[6] R. Cipolla, Y. Okamoto, and Y. Kuno. Robust structure from motion using motion parallax. In *Proc. of the 4th. ICCV*, Berlin, Germany, May 1993.

[7] B. Espiau, F. Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[8] C. Fermüller. Global 3d motion estimation. In *Proc. of the IEEE CVPR*, New York, USA, 1993.

[9] C. Fermüller. Navigational preliminaries. In Y. Aloimonos, editor, *Active Perception.* Lawrence Erlbaum Associates, 1993.

[10] B. Horn. *Robot Vision.* MIT Press, 1986.

[11] B.K.P. Horn and B. Shunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[12] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc. of the IEEE CVPR*, Seattle, USA, 1994.

[13] J. Koenderink and J. van Doorn. Affine structure from motion. *Jounal of the Optical Society of America*, 8(2):377–385, 1991.

[14] J.J. Little and A. Verri. Analysis of differential and matching methods for optical flow. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 173–180, 1989.

[15] H. Nagel. Displacement vectors derived from second-order intensity variations in image sequence. *CVGIP*, 21:85–117, 1983.

[16] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on PAMI*, 8:565–593, 1986.

[17] H. H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–323, 1987.

[18] S. Negahdaripour and S. Lee. Motion recovery from images sequences using only first order optical flow information. *International Journal of Computer Vision*, 9(3):163–184, 1992.

[19] M. Otte and H. H. Nagel. Optical flow estimation: Advances and comparisons. In *Proc. of the 3rd. ECCV*, Stockholm, Sweeden, May 1994.

[20] G. Sandini, F. Gandolfo, E. Grosso, and M. Tistarelli. Vision during action. In Y. Aloimonos, editor, *Active Perception.* Lawrence Erlbaum Associates, 1993.

[21] J. Santos-Victor and G. Sandini. Visual based obstacle detection : a purposive approach using the normal flow. In *Proc. of the International Conference on Intelligent Autonomous Systems*, Karlsruhe, Germany, 1995.

[22] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: Learning from bees. In *IEEE CVPR.*, 1993.

[23] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation : From bees to robots. *International Journal of Computer Vision, Special Issue on Qualitative Vision, Y. Aloimonos (Ed.)*, 14(2):159–178, 1995.

[24] M. Subbarao and A. Waxman. Closed form solutions to image flow equations for planar surfaces in motion. *CVGIP*, 36:208–228, 1986.

[25] V. Sundareswaran. Egomotion from global flow field data. In *Proc. of the IEEE Workshop on Visual Motion*, Princeton, New Jersey, October 1991.

[26] V. Sundareswaran, P. Bouthemy, and F. Chaumette. Active camera self-orientation using dynamic image parameters. In *Proc. of the 3rd. ECCV*, Stockholm, Sweeden, May 1994.

[27] J. Wang and E. Adelson. Layered representation for motion analysis. In *Proc. of the IEEE CVPR*, New York, USA, 1993.

[28] T. Yioshikawa. *Foundations of Robotics : Analysis and control.* MIT Press, 1990.