Embedded Visual Behaviors for Navigation

José Santos-Victor

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

The implementation of a set of visually-based behavior for navigation is presented. The approach, which has been inspired by insect's behaviors, is aimed at building a "library" of embedded visually-guided behaviors coping with the most common situations encountered during navigation in an indoor environment. Following this approach, the main goal is no longer how to characterize the environment, but how to embed in each behavior the perceptual processes necessary to understand the aspects of the environment required to generate a purposeful motor output.

The approach relies on the purposive definition of the task to be solved by each of the behaviors and it is based on the possibility of computing visual information **during** the action. All the implemented behaviors share the same input process (partial information of the image flow field) and the same control variables (heading direction and velocity) to demonstrate both the generality of the approach as well as its efficient use of the computational resources. The controlled mobile base is supposed to move on a flat surface but virtually no calibration is required of the intrinsic and extrinsic parameters of the two cameras and no attempt is made at building a 2D or 3D map of the environment: the only output of the perceptual processes is a motor command.

The first behavior, the *centering-reflex* allows a robot to be easily controlled to navigate along corridors or following walls of a given scene structure. The second behavior extends the system capabilities to the detection of *obstacles* lying on the pavement in front of the mobile robot. Finally *docking* behaviors to control the robot to a given position in the environment, with controlled speed and orientation, are presented.

Besides the long-term goal of building a completely autonomous system, these behaviors, can have very short term applications in the area of semi-autonomous systems by taking care of the continuous, tedious control required during routine navigation.

1 Introduction

The mental processes involved in directing human locomotion look, at least in adult humans, suprisingly easy and simple and do not seem to require a conscious activation of different processes or the explicit computation of environmental measures. For example, while walking along a wall it does not look like our brain is constantly monitoring the distance from the wall in spite of the fact that we usually proceed at about constant distance from the wall and if something changes along the wall (either statically or dynamically) the change in trajectory does not seem to be activated consciously. In other situations, when we pass through a narrow door, we do not need to explicitly activate a "find-the-middle-of-the-door" process which will take us safely through it or when we stop in front of an obstacle, either voluntarily or because something unexpected happens, we do not explicitly compute the distance from the obstacle to control our forward speed and approaching angle. All these processes seem to be happening "automatically" and they do in fact occur in parallel with other mental processes. In principle they cannot be considered "reflexes" (at least in the sense physiologists define them) because the same sensory input can cause rather different motor responses according to what the person is doing and the motor response can be voluntarily suppressed. However it is fair to say that the above mentioned processes seems to be running in our brain without a constant conscious supervision. In humans these behaviors are certainly the result of a developmental process during which a mapping between sensing and acting is build giving rise, for example, to a sensory-motor representation of locomotion. Once these representation is learned, usually with trial-and-error procedures and painful mistakes, the emerging behaviors become part of the daily routine.

The plasticity of the human brain accounts for the necessary adaptations to the continuously developing (and aging) structure of the human body. What is learned during the initial stages? Certainly not how to execute faster the sensory-motor processes, so that there is "more time" to do something else. What the developing child is learning is how his/her body is structured and which are the associations between sensing and acting that can be efficiently used to guide the body during the execution of different tasks. As the child matures new skills are acquired and are *embedded* in the everyday behaviors.

What is largely learned in humans often comes as a reflex in so-called lower level animals which cannot afford a long period of learning before becoming "autonomous". The approach presented in this paper has been, in fact, inspired by insect behavior and is aimed at building a library of embedded visually-guided behaviors coping with the most common situations encountered during autonomous navigation.

From the perceptual point of view, the approach has its roots in the paradigm of active perception [3, 4, 2] in the sense that the behaviors are based on active exploration of the environment and take advavantage of both the structure of the environment and the purpose of the task to be solved [1].

Two powerful assumption are made throughout the paper: the first is the possibility of using visual information *during* the action [6, 8], the second is the possibility of designing a set of closed-loop visually guided behaviors whose goal is solely that of controlling direction of heading and forward velocity of the navigating actor (the goal of perception is to act). The solutions presented do not, of course, solve the problem of navigation. Particularly, the building of a system capable of planning a path between two points in the environment has not been even attempted. Our goal, on the contrary has been to implement a set of behaviors which can economically solve a set of low-level (yet complex) navigation problems, freeing the overall navigation system (yet to be implemented) from some of the routine tasks encountered in visually guided navigation. Besides the long-term goal of building a completely autonomous system, these behaviors, can have very short term applications in the area of semi-autonomous, teleoperated systems by taking care of the tedious, tiring task of continuously driving the robot during routine navigation.

From the control point of view the approach presented here can be classified as "visual servoing" in the sense that motor control is directly driven by iconic information (i.e. measures extracted directly from the images).

From the visual processing point of view the only visual measure used is optical flow as, in many senses, it seems to be the most powerful source of information for a navigating agent and, as such, can be exploited in a very wide range of behaviors.

2 Approach

A robot moving in a indoor environment safely requires, as basic skills, the ability to detect and avoid obstacles, to navigate along walls and corridors, to enter doors and approach surfaces with a given orientation (e.g. when entering an elevator or docking in a given place to collect or deliver materials). The behaviors described here, even if not integrated yet, do in fact cover almost all of these requirements and they do it by using not only a common visual measure (which is good from the economical point of view) but they are also based on the same control commands: forward direction and velocity.

The control diagram for the various behaviors described here, is shown in Figure 1. The

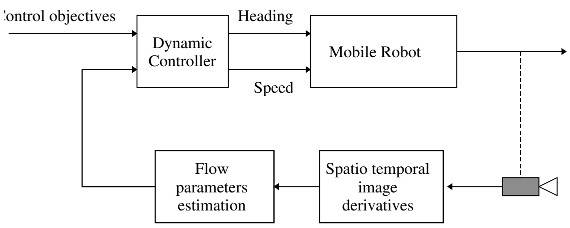


Figure 1: Overall control diagram for the various behaviors

dynamic plant to control is in all the cases a TRC LabMate mobile platform and the controlled variables are the platform linear (forward) and angular velocities. In all the cases the control is based on partial information of the optical flow field and there is no reconstruction of the environment 3D structure.

An important aspect is the way we use the optical flow to pursue our goals. Estimating the full optical flow field is an underconstrained problem due to the *aperture problem*, since with local image measurements we can only determine the flow component along the image gradient direction - the normal flow. In our approach, we exploit the particular nature of each behavior to constrain the required optical flow computations.

In the *centering-reflex*, we assume that due to the lateral position of the cameras, the vertical component of the flow is negligible and therefore, the horizontal flow component can be simply calculated as :

$$u = -\frac{I_t}{I_x} \tag{1}$$

where I_x and I_t stand for the first-order image derivatives in space (along the x direction) and time.

In the *obstacle detection* and *docking* behaviors, the hypothesis made is that the camera observes a planar surface in motion. With this assumption we approximate the flow field by an affine mapping whose parameters are estimated based on the first order spatial and temporal image derivatives. The affine flow parameters are then directly used by the system controller to drive the robot.

Therefore, by considering the specificities of each of the behaviors, we can tailor the visual processing required to achieve the control goals, and therefore overcome some of the limitations related to the *aperture problem.*

In Section 3, we present a visual behavior based on the centering-reflex observed for freely flying honeybees, *Robee*. This behavior can be used to drive a robot along corridor or wall like environments, while controlling the forward speed and avoiding lateral obstacles. Section 4 describes an obstacle avoidance behavior that complements the capabilities of *Robee*. With this functionality, the overall system is also able to detect and avoid obstacles located ahead of the robot. In Section 5 we introduce docking behaviors to control the robot to a certain point in the scene, aligning itself perpendicularly to the docking surface and controlling the velocity. We describe two situations - the ego-docking and eco-docking - which open a large number of potential applications.

3 Divergent Stereo - the Centering Behavior

The basis of the visually guided behavior of *Robee* is the *centering reflex*, described in [14] to explain the behavior 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, as described in detail in [12].

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 toward 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". For example it has been shown the possibility of overcoming unilateral or bilateral absence of flow information, caused by absence of texture, or by localized changes in environmental structure (e.g. an open door along a corridor)[1]. By applying the centering reflex in a crude mode, this lack of flow information would produce a rather unsteady behavior of the robot. This problem has been solved by introducing a *sustaining* mechanism

[1] 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).

to stabilize, in time, the unilateral flow amplitude in case of lack of information. This simple qualitative mechanism does not alter the reflexive behavior 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.

The experimental setup is based on a mobile platform with two cameras pointing laterally. The cameras, mounted 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 2. During the experiments, the vehicle forward speed is approximately 80 mm/s.

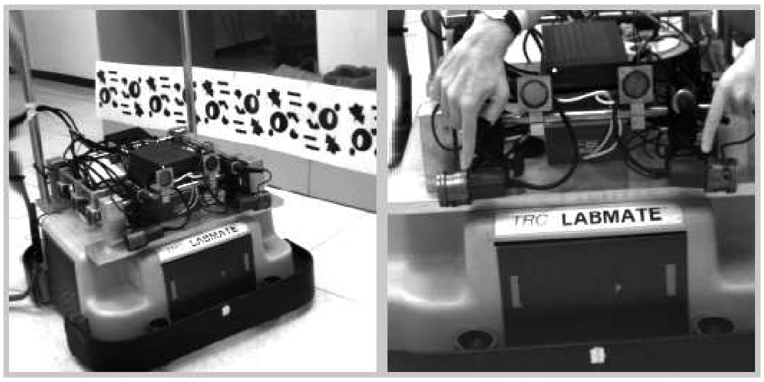


Figure 2: *Robee* with the divergent stereo camera setup

The centering behavior is based on the comparison between partial information of the flow field acquired by two cameras pointing laterally. All the details regarding the flow estimation and a thorough analysis of the influence of the robot rotation in the behavior is presented in [12].

What is worth mentioning here, however, is the fact that, instead of trying to compute the rotation component of the flow field, or to eliminate it by introducing a gaze-stabilization mechanism [5], our approach has been that of designing the system (i.e. the camera's position with respect to the axis of rotation) as well as the controller so that the rotational component of the flow becomes negligible with respect to the component of the optical flow required for the centering reflex.

An important remark is that the system does not critically rely on the accuracy of the optical flow computation, because the measurements are continuously used in closed loop.

3.1 Real-time Control

The overall structure of *Robee* control system involves two main loops. The *Navigation loop* controls the robot rotation speed in order to balance left and right optical flows. The *Velocity loop* controls the robot forward speed by accelerating/decelerating as a function of the amplitude of the lateral flow fields[2].

Additionally, a *sustaining mechanism* is implicitly embodied in the control loops to avoid erratic behaviors of the robot, as a consequence of localized (in space and time) absence of flow information.

3.1.1 Navigation Loop

The goal of the navigation control loop is to maintain the robot at similar distances from structures on the right or left sides. As mentioned before, this goal is accomplished by computing the difference between the left and right flow vectors (along the horizontal direction) and use it to control the robot heading, according to the feedback loop shown in Figure 1.

3.1.2 Velocity Control

The rationale to control the robot forward speed 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.

[2] *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).

The mean flow vector on each side of the robot gives a qualitative[3] 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 behavior 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. This control approach is also thoroughly described in [12].

3.1.3 "Sustained" behavior

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, it would be desirable to use the reactive behavior of *Robee* in environments far more complex than corridors. Additionally, 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 inappropriate behaviors (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 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). Hence, the robot will follow the ipsilateral wall at a fixed distance.

As a final remark, we would like to stress that the approach proposed extends considerably

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

the performance of the "reactive" behavior, and that the use of both the *corridor* and the *wall* following behaviors in task–driven navigation is straightforward.

3.2 Results

In all the tests, using the *Divergent Stereo* approach, the robot trajectory was recorded from odometric data during real-time experiments. Figure 3, shows the trajectories followed by the robot for various types of environment. The trajectories are shown superimposed on the experimental setup.

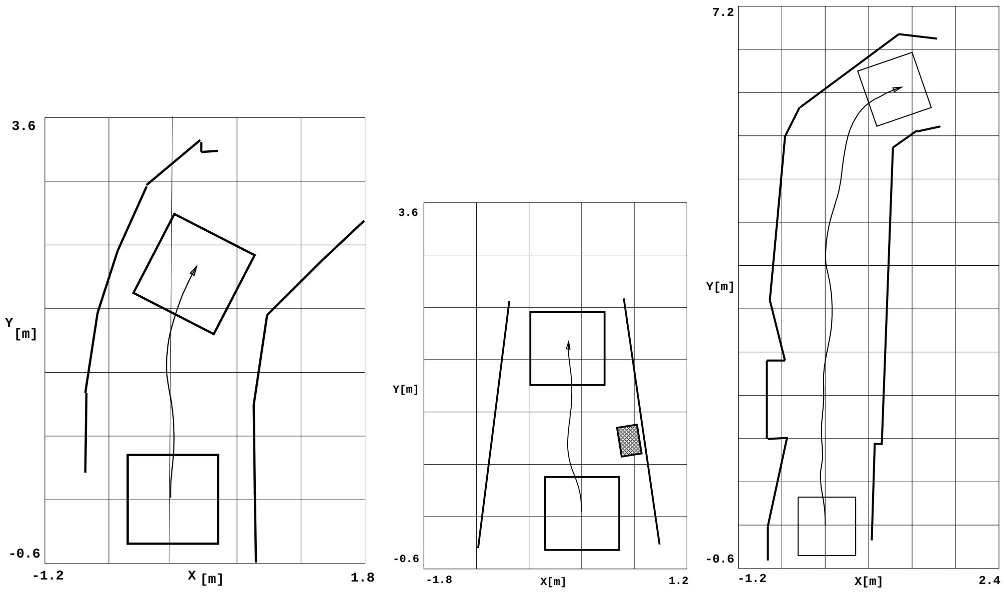
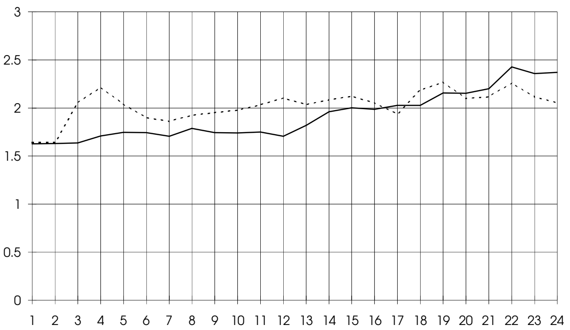


Figure 3: Results obtained in the closed loop operation for different scene layouts.

To test the velocity control, we considered the funneled corridor example, where the width changes from 1.65m to 1.25m. Since the corridor is becoming narrower, the average flow increases. By introducing the velocity control mode, the robot velocity decreases in order to keep a constant flow. Figure 4 shows the average between the left and right flows measured over time. The filled line was obtained without velocity control (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. The evolution of the robot velocity along the path is shown also in

Figure 4.

It is seen that, as the corridor is narrowing, the velocity decreases in order to keep the flow to a lower safer value. Using the velocity control, the final turn of the corridor experiment is



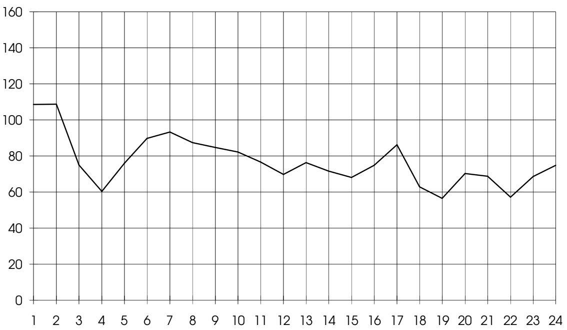


Figure 4: Left: average between the left and right flows obtained in an experiment with (dotted line) and without (filled line) velocity control. Right: robot speed in mm/s during the funneled corridor experiment.

done at a reduced speed enabling the robot to make a softer, safer maneuver.

Various experiments were designed to show the influence of the sustaining mechanism upon the robot behavior. 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 5 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.

Another experiment was made using the corridor setup, where some texture was removed from the corridor walls, and a door, roughly located at the corridor center, is left opened. The results are documented in Figure 5.

With the introduction of the sustained behavior, 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.

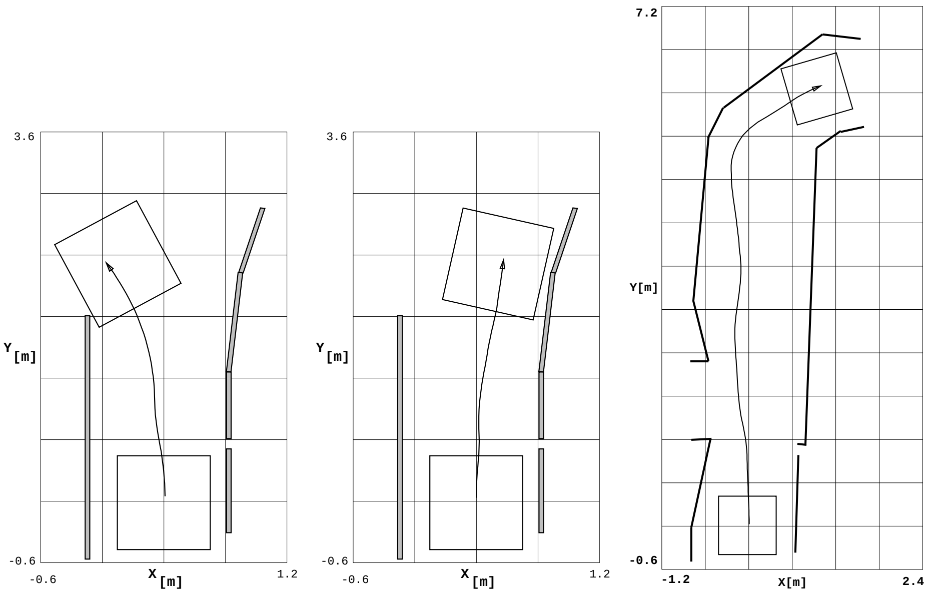


Figure 5: Sustained mechanism experiment. Left: Robot behavior without the sustaining mechanism. Center: due to the sustaining behavior, the robot manages to follow the right wall. Right: experiment along the corridor with an open door on the left and a lack of texture on the right.

4 Obstacle detection

The navigation system of *Robee* is unable to detect obstacles located ahead of the robot. Here, we describe a system for obstacle detection which uses a similar input as *Robee* to avoid obstacles. It exploits the geometric properties of the vehicle-camera-scene arrangement, and does not depend on the knowledge of the camera parameters or vehicle motion, as described in detail in [11]. The basic assumption is that the robot is moving on a ground floor (as it was the case of the centering behavior presented before) and any object not lying on this plane is considered to be an obstacle.

The method exploits the geometric structure of the camera-robot-pavement arrangement to detect obstacles lying outside the ground floor and is based on the inverse projection of the flow vector field onto the ground plane, where the analysis of the flow pattern is much simplified.

As opposed to other systems, an important feature is that the knowledge of the vehicle motion is not required and under certain circumstances, the approach is independent of the camera intrinsic parameters.

4.1 Inverse Perspective Mapping

The main idea is the analysis of the particular structure of the optical flow field, when a robot with a camera is moving over a ground plane. In Figure 6, Pc is the image plane of a camera

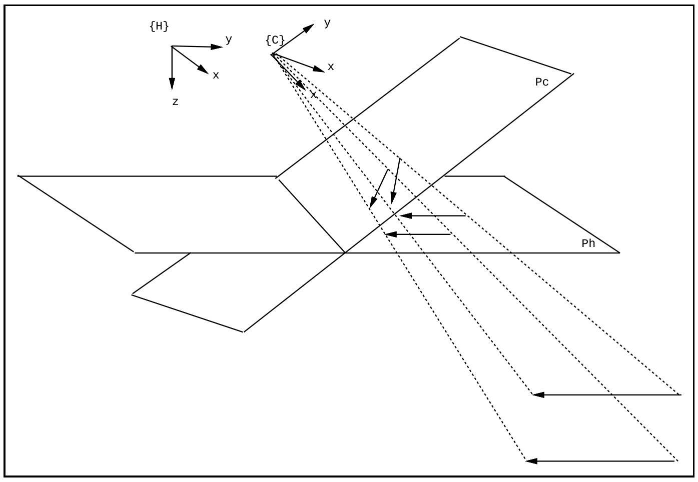


Figure 6: Inverse perspective mapping. The coordinates systems (C) and (H) share the same origin even though in the picture they have been designed separately for simplification. While on (C) the motion of the ground floor is perceived as a complex vectorial pattern, in (H) all the vectors have equal length and orientation under translational motion.

moving forwards with pure translation, while facing a planar pavement. Even with this simple arrangement, the resulting flow pattern is complex. This is due to the perspective effects, and makes the problems of motion analysis or obstacle detection difficult.

Suppose that the camera optical axis is vertical, such that the camera is pointing downwards and the image plane, Ph, is parallel to the ground plane. In this case, the flow pattern becomes much simpler as the distance to any point on the pavement is constant. All the vectors are equal, and points lying above or below the ground plane can be easily detected, as obstacles.

Orienting the optical axis vertically is not useful for practical reasons, but we can inverse project the optical flow in Pc onto the horizontal plane, Ph, as suggested in [7], thus simplifying the original problem.

The method operates in two steps. Initially, the robot moves with pure translation and the projective transformation between the image and ground planes is estimated, without the need to calibrate the camera. During normal operation, the normal flow field is inverse projected onto the horizontal plane, where obstacles are easily detected (see [13] for details).

The planarity assumption is used to approximate the flow field of the pavement to an affine mapping. The affine flow parameters are then used to estimate the projective transformation.

The more salient features are the exclusive use of the normal flow information; and that knowledge about the vehicle velocity, or camera parameters is not required. Apart from the initialization phase, the system can deal both with rotational and translational motion, of the robot, and the method is appropriate for use in real-time.

4.2 Results

The method was tested extensively using synthetic and real image data, and in real time on a mobile robot. The camera was placed in the front part of the robot facing the ground plane with an angle of about 65 degrees, with no calibration. The robot speed was set to 10 cm/s. The images resolution is 128×128 pixels and, to reduce the processing time, the computations are performed on a central window of 80×80 pixels. The running frequency of the vision loop is around 1Hz.

Figure 7 shows an image of the normal flow of the ground plane, measured during the initialization phase where the robot is performing a pure translation. The corresponding time and space image derivatives were used to estimate the affine model parameters and the inverse perspective transformation. The figure also shows the corresponding inverse projected flow, which is approximately constant, as expected.

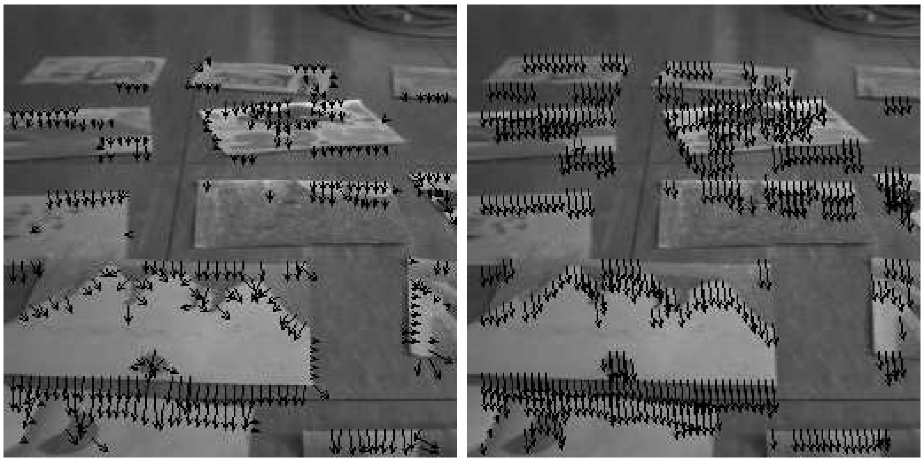


Figure 7: Left: sample of the ground plane normal flow field measured during the robot motion. Right: resulting inverse projected flow.

Figure 8 shows an example of the normal flow field measured during the robot motion, with an obstacle in the field of view. The rightmost image of Figure 8 shows the image regions where points lying outside the ground plane, have been detected. When the robot is undergoing pure linear motion, we have simply projected the inverse mapped flow in the y direction and subtracted the median flow. As the robot we used has a single forward component of the linear velocity, we have also estimated the rotation component and canceled this term from the overall inverse projected flow and then, the same detection process was applied. In both cases, similar results were obtained (see [11] for details).

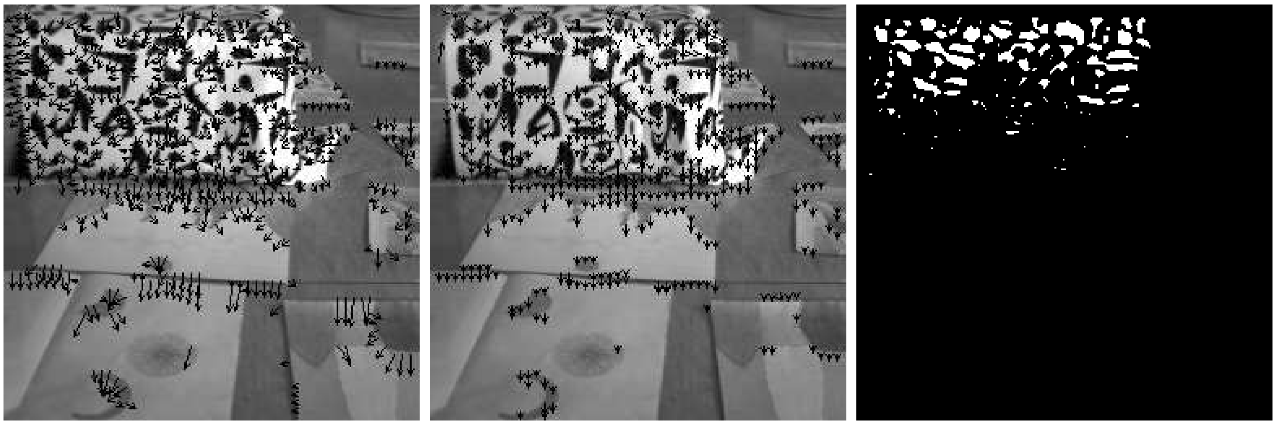


Figure 8: Left: sample of the ground plane normal flow field, during the robot motion. Center: resulting inverse projected flow. Right: detected obstacle.

In all the tests performed, the system performed robustly detecting various obstacles and stopping the robot accordingly. The resolution of the overall system determines the minimum obstacle size that can be detected, and strongly depends on the image resolution. If more computational power is available, the image resolution can be increased, the same happening with the global system resolution.

5 Docking

In this section, we introduce vision-based docking strategies for indoor mobile robots, where the robot should approach a surface, along the surface normal with controlled forward speed, until it finally stops. These behaviors stress, once more, the concept of visuo-motor coordination in, at least, two ways : The visual measure used is elicited by the motion of the robot; and 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.

Two distinct situations for the docking problem are considered. In the *ego-docking*, the camera is mounted on board the vehicle, and the robot egomotion is controlled during a docking maneuver to a particular surface in the environment. In the second scenario, that we call *eco-docking*, the camera and computational resources are installed on a single external docking station with the ability to serve multiple robots. Both scenarios are illustrated in Figure 9.

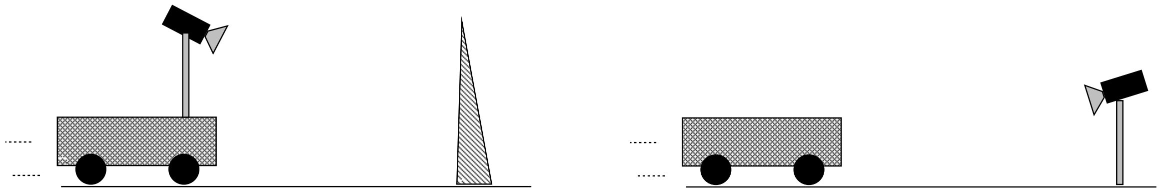


Figure 9: Left: In the *ego-docking*, a robot, equipped with a camera and computing resources, docks to a surface. Right: in the *eco-docking*, the camera is attached to a single docking station which may serve multiple robots.

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, 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. By proper formulation of the problem [9], exactly the same control architecture can be used in both cases.

As before, this behavior does not require information about the full optical flow field. Again, we use the planar surface assumption (see [10]) to approximate the flow by an affine mapping. The affine flow parameters are used directly by the robot motion-controllers. The problem of how to segment the docking surface from the global image is analyzed in detail in [9].

5.1 Visual Based Control

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 -** aligns the camera axis and the docking surface normal, during the docking maneuver. 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.

The control strategy 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.

The affine optical flow parameters, that are estimated from spatio-temporal image derivatives, are directly used to control the robot motion. One of these parameters is inversely proportional to the *time to collision*, and should be kept constant during the maneuver, slowing the robot when it approaches the goal. A second parameter vanishes when the proper orientation is reached, and can be regulated to zero to control the robot heading.

The relation between the affine parameters and the robot linear and angular velocities is similar both for the *ego-docking* and *eco-docking*. Hence, apart from a sign inversion in the rotation controller, the same controller can be used for both situations, according to Figure 1.

5.2 Results

The system has been tested in real time using a mobile platform equipped with 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 *eco-docking*, we have attached the camera to a fixed position, pointing slightly down. 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 10 shows the setup used for the eco-docking experiments.

Figure 10: Experimental setup used for the *ego-docking* (left) and *eco-docking* (right) behaviors.

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, not only to reduce the computation time but also because the affine approximation to the flow of a planar surface degrades towards the image periphery.

At the current level of implementation, the system is running approximately at 1 Hz (sampling frequency of the visual loop). In the experiments performed we have placed the robot in several initial positions relative to the docking surface, or to the docking station.

Figure 11 describes a typical ego-docking experiment, showing the robot trajectory (recovered from odometry) during the maneuver. Initially there is misalignment of about 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 11 shows also the time evolution of

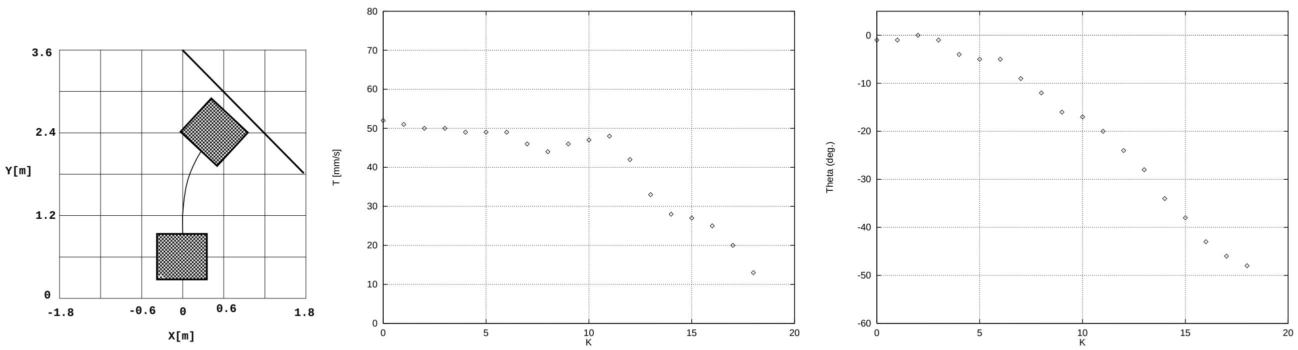


Figure 11: Left: Robot trajectory during a real *ego-docking* maneuver. Time evolution of the robot forward speed in $[mm/s]$ (center) and heading (right) in degrees.

the robot forward speed and heading direction during the maneuver, that vary smoothly as the robot approaches the goal. Figure 12 shows an image sequence of the *ego-docking* maneuver as seen by the camera.

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. We have used the same controller

Figure 12: Image sequence seen by the camera during the *ego-docking* maneuver.

apart from a sign inversion in the rotation control law. Figure 13 shows the robot trajectory during a typical *eco-docking* experiment together with the evolution in time of the robot speed and heading. Again, the robot describes a smooth trajectory until reaching the final orientation and position.

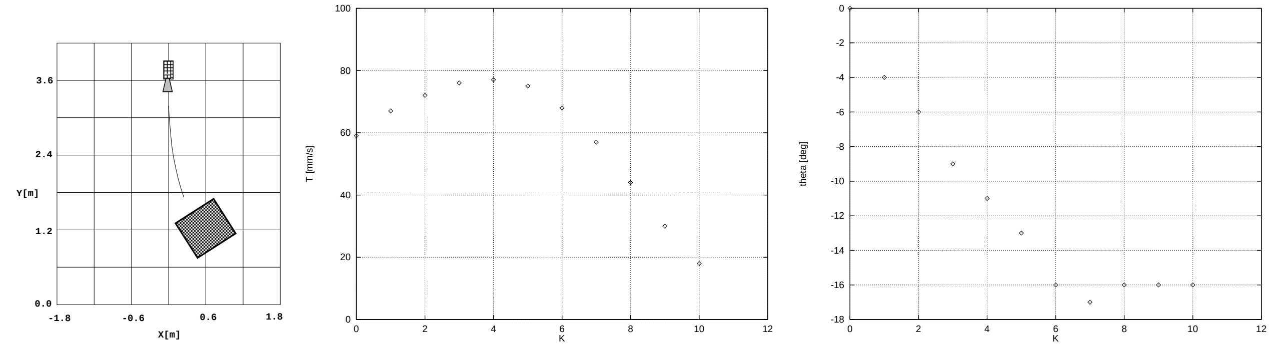


Figure 13: Left: robot trajectory during an *eco-docking* maneuver. Time evolution of the robot forward speed in $[mm/s]$ (center), and heading direction in degrees (right).

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 robot gets closer to the docking station, the speed decreases until a final stop. Figure 14 shows an image sequence acquired during the *eco-docking* maneuver, as seen by the camera at the fixed docking station.

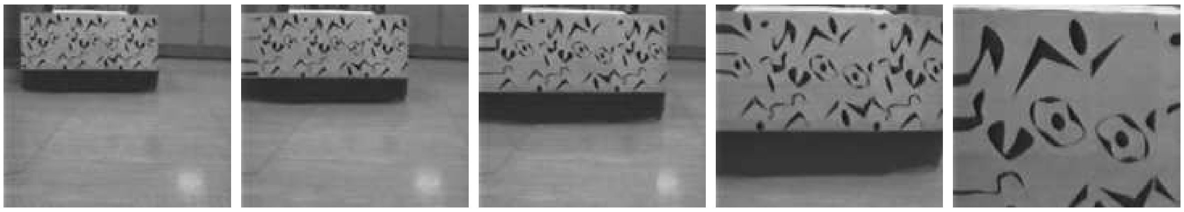


Figure 14: Image sequence seen by the camera during an *eco-docking* maneuver.

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 to the direct integration of the visual measurements in the visual loop, without any need to reconstruct the environment or calibrate the cameras.

6 Conclusions

Various visual behaviors for mobile robots have been presented solving a set of important navigation problems, like controlling a robot when crossing a corridor-like environment, following walls, avoiding obstacles or approaching a given point in the scene with controlled orientation and speed.

Apart from some specific differences, the approach adopted is based on a number of common principles. One of the main aspects is **purposive** definition of the sensory apparatus both at the geometric level (camera location) and processing level (how to adjust the visual processing for each of the perceptual tasks). Even though the approach cannot be considered general, it solves with limited complexity relevant problem in navigation: the control of heading direction in a cluttered environment, obstacle avoidance and docking to a scene point.

Another issue worth mentioning is that **qualitative and direct** visual measures are used, rather than aiming at reconstructing the workspace of the robot, allowing, in our opinion, to achieve a reasonable autonomy with limited computational power.

Another important facet of the approach is the **continuous use of visual measures**, providing a continuous stream of environmental information. This framework leads to an increased robustness, since the system as a whole, relies on a continuous stream of visual cues and does not depend on single (eventually mistaken) measurements.

Hence, the behaviors herein described illustrate 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 pre-programmed, known trajectories to reduce uncertainty). The use of vision **during** action [8], 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.

The behaviors described in the paper are examples of an holistic view of "design 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 implementation tricks but, once acting together, produce intelligent behaviors.

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 (partial information of the optic flow). Two factors characterize the different behaviors: the part of the visual field where the attention focused on; and the fact that the control law adopted reflects the direct link between visual information and vehicle motion.

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, the wall following or the docking behavior to stop in front of a door, or the obstacle detection to avoid obstacles.

However, the fact that the "appropriate action" is totally embedded inside each single behavior, seems to be a very powerful way of breaking a complex problem into simpler, tractable ones. The complexity of the perceptual processes tuned to each specific behavior and the need for "general purpose" perceptual processes becomes less crucial. We strongly believe that in the long run, the economical advantage, of this approach will become evident for artificial systems as it is already evident for natural living systems.

References

[1] J. Aloimonos. Purposive and qualitative active vision. In *Proc. ECCV90 - European Conference of Computer Vision*, Antibes, France, April 1990.

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

[3] R. Bajcsy and C. Tsikos. Perception via manipulation. In *Proc. of the Int. Symp. & Exposition on Robots*, pages 237–244, Sydney, Australia, November 6-10 1988.

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

[5] D. Coombs and K. Roberts. Centering behaviour using peripheral vision. In D.P. Casasent, editor, *Intelligent Robots and Computer Vision XI: Algorithms, Techniques and Active Vision*, pages 714–21. SPIE, Vol. 1825, Nov. 1992.

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

[7] H. Mallot, H. Bulthoff, J. Little, and S. Bohrer. Inverse perspective mapping simplifies opticla flow computation and obstacle detection. *Biological Cybernetics*, 64:177–185, 1991.

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

[9] J. Santos-Victor and G. Sandini. Visual behaviors for docking. *Submitted to Computer Vision and Image Understanding.*

[10] J. Santos-Victor and G. Sandini. Docking behaviours via active perception. In *Proc. of the 3rd Int. Symposium on Intelligent Autonomous Systems*, Pisa, Italy, 1995.

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

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

[13] José Santos-Victor. *Visual Perception for Mobile Robots : from Percepts to Behaviors.* PhD thesis, Instituto Superior Técnico, Lisboa, Portugal, 1995.

[14] M.V. Srinivasan, M. Lehrer, W.H. Kirchner, and S.W. Zhang. Range perception through apparent image speed in freely flying honeybees. *Visual Neuroscience*, 6:519–535, 1991.