Robotic Bees

G. Sandini(1), J. Santos-Victor(2), F. Curotto(1), S. Garibaldi(1)

(1) DIST - University of Genova
Laboratory for Integrated Advanced Robotics (LIRA - Lab)
Via Opera Pia 11A - I16145 Genoa, Italy
e-mail:giulio@vision.dist.unige.it

(2) CAPS/DEEC/ Instituto Superior Técnico
Institute of Systems and Robotics
Complexo 1, Av. Rovisco Pais, 1096 Lisboa codex, Portugal
e-mail: jasv@kappa.ist.utl.pt

ABSTRACT

This paper presents some experiments of a real-time navigation system driven by two cameras pointed laterally to the navigation direction (divergent stereo). The approach is based on the observation that the stereo set-up traditionally used in vision (i.e. with the optical axis pointing forward) may not be the best one for navigation, and particularly for continuous control of a mobile actor moving in unconstrained environment. Similarly to what has been proposed [5, 3], our approach [8, 10] is that for navigation purposes the driving information is not distance (as it is obtainable by a stereo set-up) but motion and, more precisely, by optical flow information computed over different areas of the visual field.

Following this idea, a mobile vehicle has been equipped with a pair of cameras looking laterally (much like honeybees) and a controller based on fast, real-time computation of optical flow, has been implemented. The control of the mobile robot (ROBEE) is based upon the comparison between the apparent image speed of the left and the right eye. The solution adopted is derived from recent studies [13] describing the behavior of freely flying honeybees and the mechanisms they use to perceive range.

This qualitative information (no explicit measure of depth is performed) is used in many experiments to show the robustness of the approach and a detailed description of the control structure is presented to demonstrate the feasibility of the approach in driving the mobile robot within a very cluttered environment.

A discussion about the potentialities of the approach and the implications in terms of sensor's structure is also presented.

1. Introduction

Research in computer vision has been motivated very often by similarities with solutions adopted in natural living systems. S-ince the early work in image processing and computer vision the structure of the human visual system has been used, quite often, as the“natural” model of artificial visual systems. Comparatively less effort has been devoted to the study and implementation of artificial visual systems based on the so-called “lower level” animals. The motivation of this biased research activity stems, mainly, from the idea that by “understanding human vision” one could obtain a “general” solution to the visual process, and eventually being able to develop a “general purpose visual system”.[1] We certainly agree with the general idea that there is a lot to learn from human vision (and conversely there is a lot to be gained by trying to build artificial systems with anthropomorphic features), on the other hand we do not believe that such a “general purpose vision system” actually exist, and that a lot can be gained and understood by looking at much simpler biological systems.

Looking more carefully, in fact, it is evident that the human visual system is not as general purpose as it may look. For example it is useless in underwater environments, it has a limited spectral band, it cannot measure distance or size or velocity in metric terms. Even its recognition system can be fooled very easily for example, if you turn a picture of a face upside down, the recognition of even familiar faces becomes impossible. The apparent generality of the system is due to the fact that we actually perform a limited number of motor and cognitive “actions” and, within this limited domain, our visual system (or more generally the integration of our perceptual systems) performs very efficiently.

Following this ideas, one could say that the goal of a vision system in a “living” agent is not generality but specificity: the physical structure and the purpose drives perception [1].

Within the scope of this paper we would like to argue that the frontal position of the eyes (with a very large binocular field) is mainly motivated by manipulation requirements and, if one restricts the purpose to navigation control a potentially more effective positioning is the one adopted by flies, bees and other flying insects, namely, with the optical axis pointing laterally (divergent stereo). The tradeoff in these two extreme situations is that in the former case the global visual field (i.e. the union of the visual fields of the two eyes) is smaller than in the latter one. For example humans have a visual field which is limited, in the horizontal meridian, to about 180 degrees while some insects have a 360 degrees visual field. On the other hand by having a wider binocular field the regions where a stereo-based depth measure is possible become larger. Looking again at biology one finds out that the position of the eyes in different species becomes more frontal as the manipulative abilities increase (it is not a case that humans and primates have, among all species, the wider binocular field and the more frontal positioning). Reasoning on a different line of thoughts one could ask the question of “what are the visual modalities that may give *depth* (not shape) information”? One of them is stereo, the other one is motion parallax (and some related measures). Motion parallax is useful if and only if the ”actor” or the environment is moving while stereo is mandatory

[1] and, possibly, the notion that “human vision is complex while insect vision is simple.”

if the environment is static. From this observation one could say that stereo is somehow superior for manipulation tasks because "it always works" even if the observer does not move (the objects to manipulate are, very often, static). On the other hand, if the observer is moving (such as during navigation), motion parallax can be used and, moreover, motion-derived features, such as time-to-crash, seem more relevant to dynamically control posture and other navigation parameters: if one has to jump over an obstacle, the change in posture while he is approaching the obstacle (for example when to start to lift the leg), may be driven by "time-to-crash" more than by distance (which would be dependent upon the speed of approaching). Moreover by relaxing the requirements in terms of binocularity of the visual field, a wider field of view may be possible *without increasing the computational power* [2].

A very interesting biological model of this approach comes from the world of insects and the use made by these systems of flow information, to solve apparently complex motor tasks like flying in unconstrained environments and landing on surfaces[12],[5]. From the sensor's point of view the eyes of insects, besides the peculiarities already pointed out, do not have accommodation and, the stereo baseline is so small, that no binocular stereo can be reliably measured. Nonetheless they are able to control their flying direction, avoiding obstacles and landing on surfaces.

Following this line of thought, a mobile robot (ROBEE) has been equipped with laterally looking cameras and a controller has been implemented, based upon motion-derived measures, which do not rely on precise geometric information but take full advantage of the continuous (in time) nature of visual information.

In the experiments reported in this paper, navigation is controlled on the basis of the optical flow field computed over windows of images acquired by a pair of cameras pointing (in analogy with the position of the eyes in the honeybees and other insects) laterally and with non overlapping visual field. We called this camera placement *divergent stereo*.

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

2. The Divergent Stereo Approach

In this section we will describe the *Divergent Stereo* approach proposed for robot navigation. As a first step, we will describe the experimental setup used to perform all the experiments. We will continue by describing our simplified computation of the optical flow, and the motion parallax sensor, and, finally, we will discuss the problem of closing the loop based on this visual information.

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

2.1. Experimental Setup

The experimental setup is based on a computer controlled mobile platform, *TRC Labmate*, with two cameras pointing laterally. The cameras are connected to a VDS 7001 Eidobrain image processing workstation. The left and right images are acquired simultaneously, during the vehicle motion. The experimental setup is illustrated in figure 1.

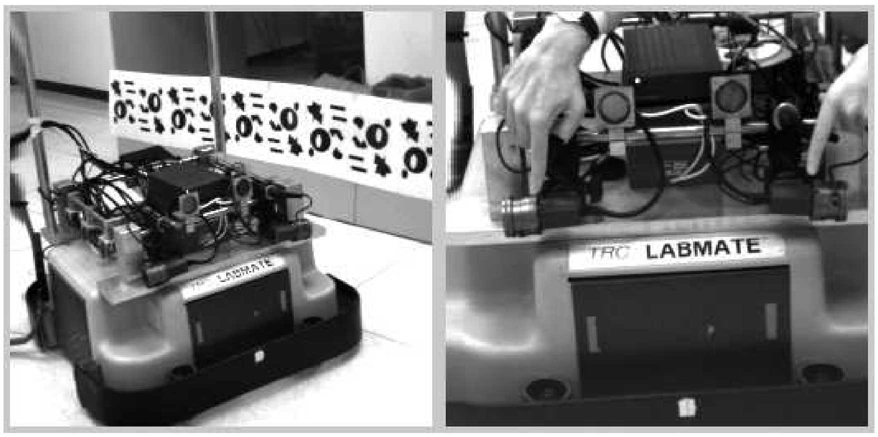

Figure 1: The vehicle with the divergent stereo camera setup

During the experiments, the vehicle forward speed is approximately 80 mm/s. We are using two Cohu cameras and the lenses have auto-iris control and a focal length of 4.8mm.

2.2. Optical Flow Computation

To compare the image velocity as seen by the left and right cameras, we compute an average of the optical flow on each side. The optical flow $V = (u, v)$, is usually obtained by using the fundamental optical flow constraint [7], shown in equation (1), or by imposing the stationarity of the flow field [6, 14, 4], according to equation (2):

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{1}$$

$$\frac{d}{dt}\nabla I = 0 \tag{2}$$

By using expression (2), one ends up with the following system of equations, which can be solved to obtain both components of the optical flow.

$$I_{xx}u + I_{xy}v = -I_{xt} \tag{3}$$

$$I_{xy}u + I_{yy}v = -I_{yt} \tag{4}$$

where I_{xx}, I_{yy} and I_{xy} stand for the second spatial derivatives of the image, while I_{xt} and I_{yt} stand for the time-space derivatives. In order to obtain reliable flow estimates, the usual procedure is to perform some smoothing both in time and space domains.

We have developed a simplified method for the computation of the optical flow [10]. On one hand, the time filter is causal, in the sense that we only need the images available up to the instant under analysis. On the other hand, for this particular application, we constrain the flow to be horizontal, as the vehicle is moving on a flat plane, thus simplifying the calculations.

We acquire a sequence of 5 images (at video rate), and start the processing concurrently. The images acquired have a resolution of 256x256 pixels, and the optical flow is computed over a window of 32x64 pixels on each side. At the current implementation, the averaged optical flow vectors are computed at a frequency of approximately 1.5Hz.

2.3. Closing the Loop

As explained before, we aim at using the information of lateral flow, in order to implement a sort of reflex-type control strategy for robot navigation.

The robot control system allows the possibility of performing rotations, at a given speed, while still moving forwards. The rotation velocity is used to control the robot's motion direction. In an extended version of this paper [11] experiments involving also the control of the forward velocity are presented. The resulting behavior is even more interesting in that the forward speed is inversely proportional to the width of the corridor. so that the robot accelerates if the lateral flow is low (meaning that the walls are further away), and slow down whenever the flow becomes larger (meaning that it is navigating in a narrow environment).

2.3.1. Control Synthesis

In this section, we discuss the control system design problem. By using some simplified dynamical models of the several components of the control chain, one can use linear systems theory to provide some useful insight on the closed loop system behavior. We intend to work further on the modelization of the system in order to be able to perform a more accurate control analysis/synthesis.

To control the direction of the robot motion (*controlled variable*), we can manipulate the rotation speed of the vehicle (*control variable*). The rotational motion is superimposed on the translational motion. The simplest model of the system must account for two important terms: The input variable is an angular velocity while the output is an angular position, meaning that there is an integrator in the system. On the other hand, there is the mechanical inertia of the system, which we have modeled by a first order dynamic system.

In continuous time, the transfer function relating the *control* and *controlled* variables, according to our simple model, is given by:

$$G_c(s) = \frac{1}{s}\frac{a}{s+a} \tag{5}$$

where a is the dominant low-frequency mechanical pole.

Since we are using digital control, we must determine how the computer (where the control algorithm is implemented) "sees" the system. Using a zero-order hold sampling procedure, we can use the step invariant method to determine the discretized system [2]. Considering the low-frequency pole at $5Hz$, and a sampling period of $\tau = 0.7s$, we obtain:

$$G_d(z) = \frac{0.468z + 0.0318}{z^2 - (1 + 1.51e^{-7})z + 1.51e^{-7}} \quad (6)$$

Again for simplicity reasons, we may assume that the difference between the left and right flow vectors, is proportional to the error between the robot direction and the direction of the lateral walls. This assumption, of course, is only valid for small course deviations, since the visual perceptual process is, in fact, non linear. Moreover, there is a delay introduced by the flow calculations. This delay is one sampling period. The sensor model is then given by:

$$e(k) = \theta_{walls}(k-1) - \theta_{robot}(k-1) \quad (7)$$

We are using a PID controller to close the navigation loop. The discrete time PID controller performs the following control law:

$$u(k) = K_p[\ \ e(k) + K_i \sum_n e(k-n) + K_d(e(k) - e(k-1))\ \ ] \quad (8)$$

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

$$G_{PID}(z) = K_p \frac{(K_i + K_d + 1)z^2 - (1 + 2K_d)z + K_d}{z(z-1)} \quad (9)$$

By connecting all these models, we obtain a linear feedback loop as shown in figure 2.

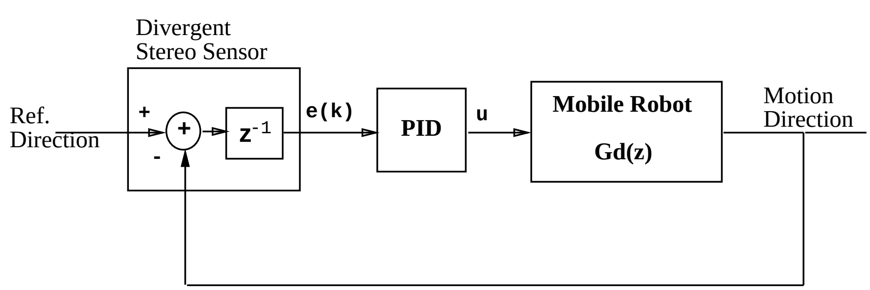

Figure 2: This block diagram illustrates a simple modelization of the robot navigation system

Qualitatively, when $e(t)$ is positive, the flow is higher on the left side, meaning that the left wall is closer and, therefore, an appropriate control action would be to turn right.

We can now analyze the system behavior for different controller settings. Although it is difficult to perform a very thorough analysis, since we are working with approximated models, we can understand how the control will perform under several classes of controllers.

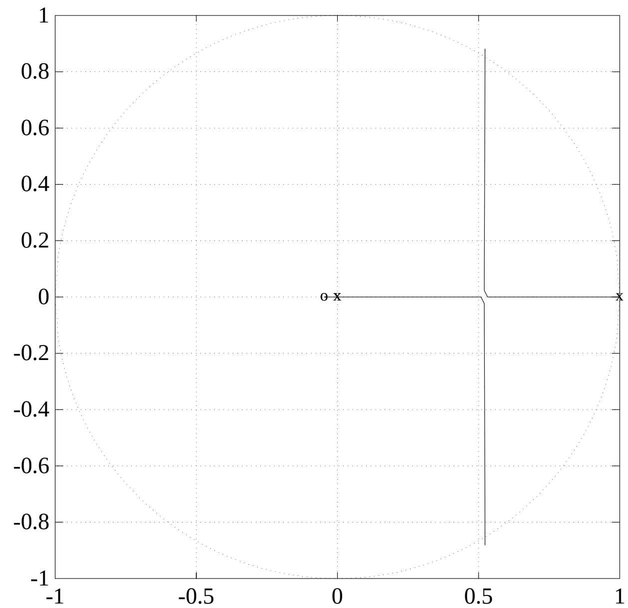

Figure 3: Root-Locus corresponding to the proportional controller. The unit circle is shown in dotted line for easier stability analysis.

As a start, we could just close the loop using the proportional gain (by setting the integral and derivative gains to zero). The root-locus corresponding to this situation is shown in figure 3.

Even though the proportional controller succeeds in stabilizing the system, we would need large gains to achieve fast responses, Also, for large gains the system exhibits a significant oscillatory behavior, as the dominant poles become complex conjugate.

By additionally including a derivative term (which basically works as a predictive term, thus coping better with the delay), one can expect to achieve faster responses, even for smaller gain values. Figure 4 shows the root locus in this situation. The derivative component is fixed to $K_d = 1.5$.

The effect of adding the extra zero is that the low frequency pole is attracted into the higher frequencies, thus improving the response time of the system. Also, for large gains, the oscillatory behavior of the system is reduced.

Finally, we can analyze the effect of inserting the integral component of the controller. Figure 5 shows the root locus for this case. The parameters used are $K_i = 0.1$ and $K_d = 1.5$.

In this last situation, adding a discrete integrating effect (pole in 1), we decrease the stability margin of the system, and decrease the response speed. In practice, the system may even become unstable in this situation considering that the visual sensing process is, in fact, non linear, and that between sampling instants, the system operates in open-loop. Therefore, this extreme situation may leed to unstable behavior.

This simplified analysis, allowed us to understand approximately the behavior of the system under the different controller topologies. Still, further modeling is necessary to perform a quan-

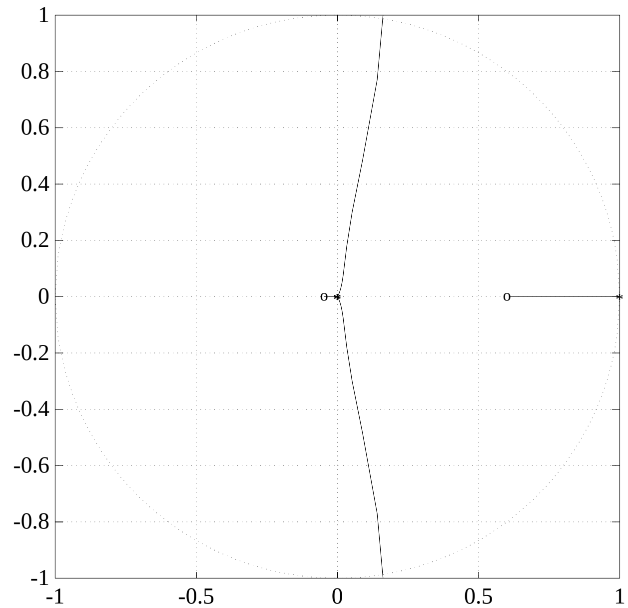

Figure 4: Root-Locus corresponding to the proportional-derivative controller. The derivative gain was fixed at $K_d = 1.5$
.

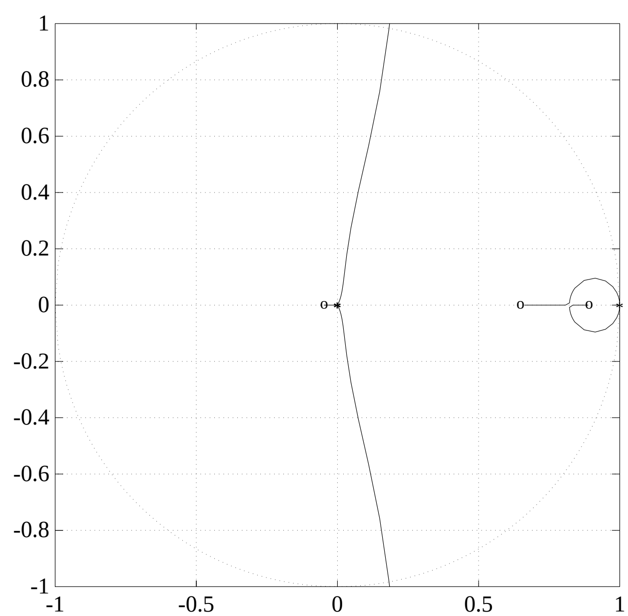

Figure 5: Root-Locus corresponding to the PID controller.

titative analysis or a more systematic control synthesis. Nevertheless, some of the ideas discussed in this section were later on verified in real experiments, as described in the section of results.

3. Results

In this section we present a set of tests to demonstrate several applications of the divergent stereo navigation approach. To start, we will show an example in open loop, that illustrates the capabilities of the perception process, and how the different situations can be detected. Finally, we will show a set of closed loop examples, with different experimental setups, evidencing the importance of the controller design.

3.1. Open loop analysis – Obstacle Detection

In this test, we are interested in studying the obstacle detection capabilities with the divergent stereo setup. For this purpose, we used a setup as shown in Figure 6. We have placed an obstacle in the middle of the corridor in a position such that, when passing by the obstacle, the robot will be at similar distances from the obstacle and from the opposite wall. Therefore, when comparing the flow on both sides, the difference should be close to 0.

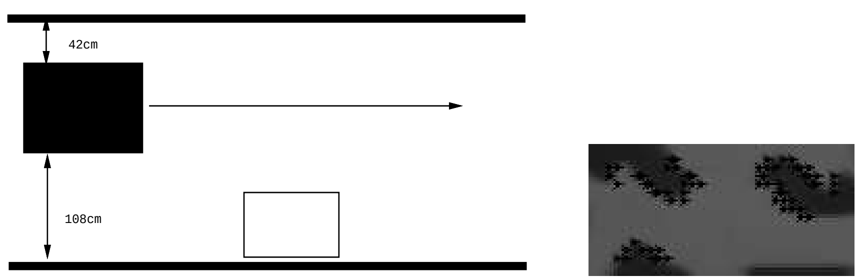

Figure 6: Obstacle detection experiment. The obstacle is approximately at the same distance of the robot as the left wall. On the right it is shown a sample of the computed optical flow. The number of flow is about 400 vectors in each image.

Figure 7 shows the signal corresponding to the difference between the left and right flow fields. The difference starts with a positive value, as the robot is closer to the left wall than to the right wall and, when passing close to the obstacle, the error clearly changes, tending to zero, as the object is at the same distance as the left wall and, finally, after passing the obstacle, the difference tends again to approximately the initial value, as the robot continues to follow a rectilinear path.

3.2. Closed Loop Operation

In this section we will describe a set of closed loop experiments, using the proposed perceptual and control strategies. The goal of the experiments is the study of the closed loop behavior of the visually guided robot. For this purpose, we have considered two experimental situations with increasing degree of difficulty.

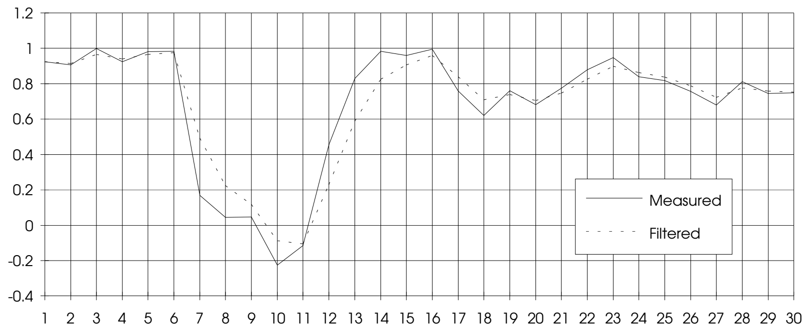

Figure 7: The error between the left and right average flows, during the obstacle detection experiment. The full path corresponds approximately to a distance of 2m, lasting for 20seconds.The presence of the obstacle is clearly detected in the error signal. As before, the filtered version of the error signal is smoother in time.

3.2.1. Turn Experiment

On the first set of experiments, we have used the robot to navigate in a turn. The experimental setup is shown in figure 8.

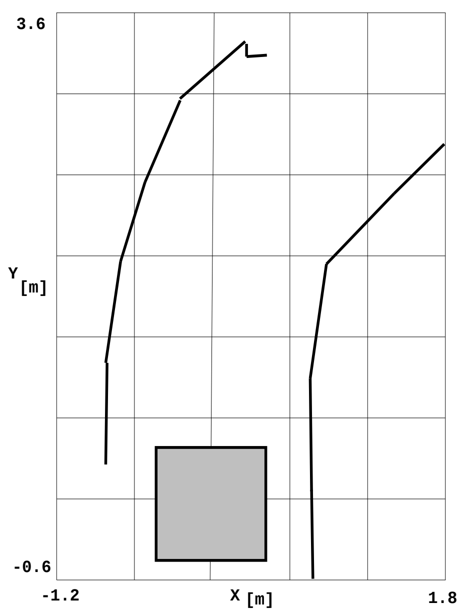

Figure 8: Setup used for the *turn* experiment. The vehicle is supposed to perform the turn, using the divergent stereo navigation strategy.

We have tested the whole navigation system with different settings of the controller parameters. For these experiments, the parameter K_i, the integral gain, was kept fixed to zero, as suggested by the discussion on the controller design.

In the first three experiments, we studied the influence of the derivative gain on the navigation performance. Figure 9, shows the trajectories followed by the robot with a fixed proportional gain, $K_p = 1.5$, and using for K_d the values of 1.0, 1.2 and 1.4. The trajectories were recorded using the odometry information and are shown superimposed on the experimental setup.

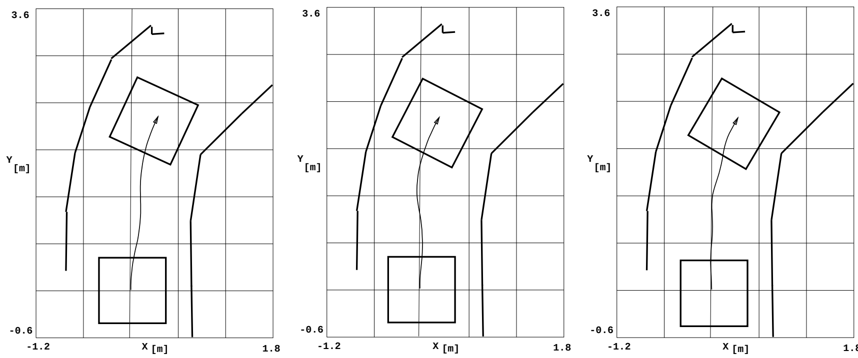

Figure 9: Results obtained in the closed loop operation in the turn experiment. The controller settings were $K_p = 1.5$, $K_i = 0$ and using for K_d the values (from left to right) of 1.0, 1.2 and 1.4. The trajectory was recovered using odometry.

To test the influence of the proportional gain on the controller, we performed another set of experiments. In this situation, we kept the integral and derivative gains constant ($K_i = 0$ and $K_d = 1.2$) while using for K_p the values of 1.25, 1.5 and 1.75. The results are shown in figure 10.

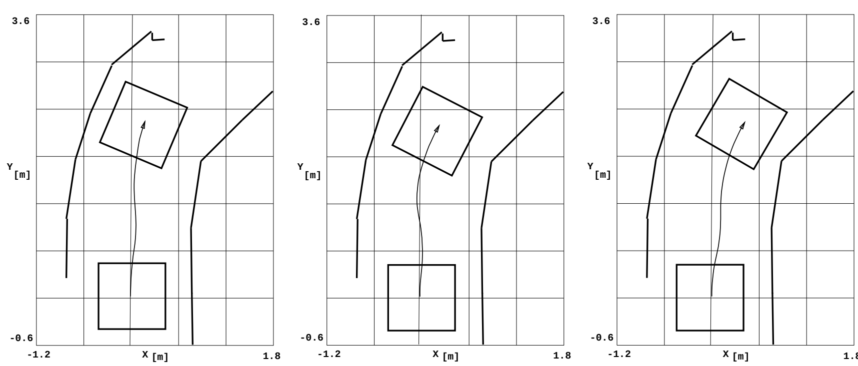

Figure 10: Results obtained in the closed loop operation in the turn experiment. The controller settings were $K_d = 1.2$, $K_i = 0$ and using for K_p the values (from left to right) of 1.25, 1.5 and 1.75. The trajectory was recovered using odometry.

The analysis of the trajectories show that, by increasing the value of K_d, the response becomes faster, even though the trajectories may become less smooth. On the other hand, by increasing K_p, also the response is faster but the vehicle does not succeed to have such fast reactions as by increasing the derivative component.

3.2.2. Funneled corridor

In another set of experiments, we have used a funneled corridor which has an obstacle to be avoided by the robot. The setup is shown in figure 11. Therefore, the robot will have to avoid an obstacle while trying to align to the center of the funneled corridor.

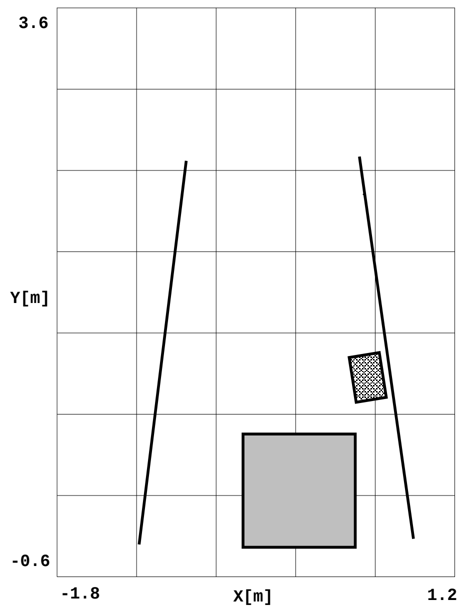

Figure 11: Setup used for the funneled corridor experiment. In this scenario, the robot will have to avoid an obstacle while managing to keep the track in the funneled corridor.

Again, we have used different settings of the PID controller, in order to study the obtained trajectories. In figure 12, we show the trajectories corresponding to three of those experiments. The robot's trajectory, recovered from odometry, is superimposed on the experimental setup layout.

The first trial represents an experiment with the controller tuned with $K_p = 1.5$, $K_i = 0.0$, and $K_d = 1.5$, which led to a nice trajectory. On the second trial, we have introduced the integral component, using the following controller settings: $K_p = 1.5 \quad K_i = 0.1 \quad K_d = 1.5$ As suggested by the discussion made in section 2.3., an unstable behavior was observed, with the vehicle moving towards the lateral wall. Finally, on the third experiment, we also obtained a nice performance by slightly increasing the proportional gain, while diminishing the derivative gain: $K_p = 1.8$, $K_i = 0.0$ and $K_d = 1.2$.

Again, as expected we obtained a somewhat smoother trajectory. The tests performed, enhance the importance of suitable control system design, and clearly show the use of our discussion of the modeling and control system design. Further modeling is nevertheless necessary.

In general, though, it seems reasonable to claim that the proposed approach, both on the control and visual perception facets, led to good results and proved the feasibility of a navigation system based on these principles.

It is worth noting here that, during the rotational motions of the vehicle (e.g. while avoiding an obstacle), the values of the flow fields are not only proportional to the distance of the walls but also to the rotational velocity of the vehicle. This interferes

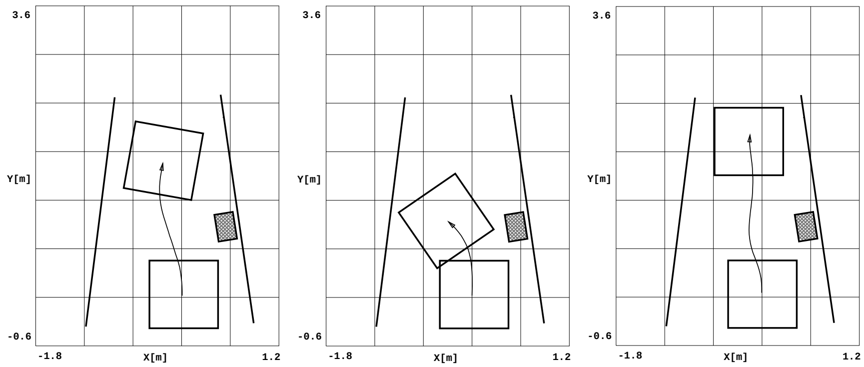

Figure 12: Funneled corridor experiment. The leftmost and rightmost plots show the results of increasing the proportional gain while decreasing the derivative gain. In the center, it is seen the unstable behavior due to the insertion of the integral action.

the simple control introduced here. The analysis of the influence of the rotational speed in the the proposed scheme is studied by the authors in [11], where some solutions are pointed out. The main conclusion there is that it is possible to design the set-up in such a way that this rotational component does not degrade the performance of the system. An alternative solution has been proposed recently by Coombs and Roberts [3] based on the use of a gaze control system counter-rotating the cameras with respect to the vehicle.

4. Conclusions

In this paper, we have presented a qualitative approach to visually-guided navigation based on optical flow. The approach followed was originally motivated by the study and experiments performed on freely flying honeybees to understand the range perception mechanisms.

The approach is based on the use of two cameras mounted on a mobile robot and with the optical axis directed in opposite directions such that the two visual fields do not overlap (divergent stereo). Range is perceived by computing the apparent image speed on images acquired during robot's motion. No attempt is made to compute range in metric terms (i.e. distance) but the direct measure of optical flow is used to control in real-time the motion of the vehicle.

We proposed a control strategy to visually close the navigation loop. Based on a simple modelization of the different system components, we succeeded in performing an analysis of the system performance, using a PID controller.

A real-time computation of optical flow is presented based upon the constraints imposed by the geometry of the cameras and by the navigation strategy.

Several examples of reflex-type control of motion driven by differential estimation of the flow field measured by the two eyes are presented. We have tested the system under different difficult tasks, like performing a tight turn or navigating through a funneled corridor, and investigated the influence of the control parameters on the trajectory. The experimental analysis confirmed, to some extent, the discussion on the control system design.

All the experiments were performed without the need for accurate depth or motion estimation or required a calibration procedure (besides the manual positioning of the two cameras).

Before concluding it is worth mentioning the fact that, at its current level of implementation, ROBEE is by no means satisfactory because it is blind in the direction of motion (it will bump against an obstacle just in front of it). On the other hand we are currently using only a very limited portion of the visual field and, the use of a more frontal part of the visual field, is currently investigated to extract other motion-derived measures (e.g. time-to-crash) which could help ROBEE not only to avoid obstacles [9], but also to control docking speed.

Before concluding it is important to refer to a recent article [3] presenting a system (*beebot*) which has very strong similarities with our *ROBEE* approach. The main difference between the two systems lies on the fact that, while in the system proposed by Coombs and Roberts gaze control is part of the navigation strategy, in the present implementation a simpler set-up has been analyzed where gaze control becomes unnecessary by appropriately positioning the two cameras and by controlling some behavioral variables such as the "turning speed". A deeper descussion about the differences of the two systems will be presented in an article submitted for publication [11]

Acknowledgements

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 VAPII

References

[1] J. Aloimonos. Purposive and qualitative active vision. In *Proc. of Int. Workshop on Active Control in Visual Perception*, Antibes (France), 1990.

[2] K. Astrom and B. Wittenmark. *Computer Controlled Systems: Theory and design.* Prentice-Hall, 1986.

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

[4] E. DeMicheli, G. Sandini, M. Tistarelli, and V. Torre. Estimation of visual motion and 3d motion parameters from singular points. In *Proc. of IEEE Int. Workshop on Intelligent RObots and Systems*, Tokyo, Japan, 1988.

[5] N. Franceschini, J. Pichon, and C. Blanes. Real time visuomotor control: from flies to robots. In *Fifth Int. Conference on Advanced Robotics*, Pisa, Italy, June 1991.

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

[7] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 No.1-3:185–204, 1981.

[8] G. Sandini, J. Santos-Victor, F. Curotto, and S. Garibaldi. Robotic bees. Technical report, LIRA-Lab - University of Genova, October 1992.

[9] G. Sandini and M. Tistarelli. *Robust Obstacle Detection Using Optical Flow.* Proc. of IEEE Intl. Workshop on Robust Computer Vision, Seattle, (WA), Oct. 1-3, 1990.

[10] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: Learning from bees. In *To appear on Proc. CVPR-93*, New Yors, U.S.A., 1993.

[11] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: A step forward to a robotic bee. *Submitted for publication*, 1993.

[12] M.V. Srinivasan. Distance perception in insects. *Current Directions in Psychological Science*, 1:22–26, 1992.

[13] 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.

[14] S. Uras, F. Girosi, A. Verri, and V. Torre. Computational approach to motion perception. *Biological Cybernetics*, 60:69–87, 1988.