Divergent Stereo for Robot Navigation : Learning from Bees

J. Santos-Victor

DEEC - Instituto Superior Técnico
ISR, Lisboa - Portugal
e-mail: jasv@kappa.ist.utl.pt

G. Sandini, F. Curotto, S. Garibaldi

DIST - University of Genova
LIRA-Lab, Genoa - Italy
e-mail:giulio@vision.dist.unige.it

Abstract

A qualitative approach to visually-guided navigation based on the computation of optical flow field is presented.

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 computation is based on the computation of the apparent image speed on images acquired during robot's motion.

An example of reflex-type control of motion, driven by differential estimation of the flow field measured by the two eyes, is presented. In particular it is shown how a difficult task like navigating through a funneled corridor with obstacles, is possible without the need for metric depth estimation.

1 Introduction

Most of the work carried out in the field of robot navigation has given very much attention to the analysis of motion but, on the other hand, if one looks more deeply to *how* it has been done, most of the work carried out was aimed more at the reconstruction of the external environment, rather than to the closed-loop control of robot motion. Apart from some exceptions the motion of the robot has been used as a way to change the vantage point and, consequently, to acquire different views of the environment, disregarding the computation of what it is really necessary to keep the robot "on track", to terminate the motion, or to execute more complex motor actions (such as turning to enter a door, slowing down in front of an obstacle, avoiding the obstacle, following a wall etc.).

Starting from the (more or less implicit) assumption that geometry has a predominant role (in some sense, it is the goal) one is led to a somehow limited (and perhaps distorted) view of the role of vision and, consequently, of what to compute and how. It is not rare to see experiments performed with robots moving along linear (or piecewise linear) trajectories, or presenting as a final result the reconstructed map of the environment in terms of *surfaces*. This approach, although important for many reasons (and for some industrial applications), fails, in our opinion, to highlight the role of vision *during* the execution of motor actions and, to some extent, makes the overall process more complex requiring a degree of accuracy much higher than what it is really necessary, and certainly outside current (and possibly even future) technology.

The view presented in this paper is somehow different, because it is aimed at defining some of the visual measures that can be adopted to control navigation, which do not rely on precise geometric information but take full advantage of the continuous (in time) nature of visual information.

Paradoxically, our model is such that a still mobile robot is virtually blind (very much like some "lower" level biological systems). This extreme situation is, by no means, the final goal of our research but it helps in trying to define what can be computed during the motion of the robot which can be used to continuously control the navigation parameters (i.e. direction and velocity). The goal of our system 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 based upon very simple, direct estimation of "range" extracted from 2D (or even 1D) optical flow.

The inspiration of the ideas presented here derives from some recent research results, on the behavior of honeybees [1] and, more generally, from the use some insects make, of flow field information [2] to control their flying trajectory in unconstrained, hostile environments [3]. From the computational view point the problems to be solved are far more complex than what we usually do in controlling a mobile robot moving on a flat surface.

A simple example would suffice to explain the kind of simplifications that can be gained by solving apparently complex navigation problems using qualitative measures of optical flow. In the experiment of Srinivasan [1] honeybees were trained to navigate along corridors in order to reach a source of food. The behavioral observation is the fact that, in spite of the fact that the corridor was wide enought to allow for "irregular" trajectories, bees were actually flying on a quasi-linear trajectory in the middle of the corridor (i.e. maintaining the same distance from the left and the right wall). This finding is even more surprising if we take into consideration that:

- the two eyes of the honeybees are ponting laterally (at about 180 degrees.)
- the distance of the two eyes is so small that, even in the binocular field of view, depth-from-stereo does not provide any useful depth information
- no accommodation mechanism is present providing depth-from-focus information

Apparently, then, no depth information can be derived using traditional methods. What kind of information is used to maintain the bee in the middle of the corridor and how it is computed is, therefore, a non trivial question. The answer presented in [1], is rather simple and it is based on the difference between the velocity information computed from the left and the right eye: if the bee is in the middle of the corridor the two velocities are the same, if the bee is closer to one wall, the velocity of the ipsilateral eye is larger. A simple control mechanism (very similar to a reflex), then, may be based on motor actions that tend to minimize this difference: move to the left if the velocity measured by the right eye is larger than that measured by the left (or vice versa).

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 important points are:

- no explicit depth information is necessary;
- the control strategy is rather simple and the only constraint is the presence of texture on the image;
- a reflex-like control is possible (i.e. no geometric and/or a-priori information about the environment is necessary)
- range is measured in terms of image velocity. Even if this solution may seem limited, it is, in our opinion very well suited to navigation. For example the perceived range is implicitly tuned to the speed of the robot: if the robots move faster the objects are perceived as closer. If one consider that the "reaction time" to unexpected events is, in all physical systems, constant, it seems very efficient to scale distance with velocity: what is important is not how far the object is but how much time one has to react to it. In other words, if the velocity of the vehicle is small a close object is "less dangerous" than if the velocity is higher because the system will have a longer time to react (this fact is very evident in driving through a narrow path).

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 control loop based on this visual information.

2.1 Experimental Setup

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

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

2.2 Optical Flow Computation

To compare the image velocity as seen by the left and right cameras, we compute an average of the op-

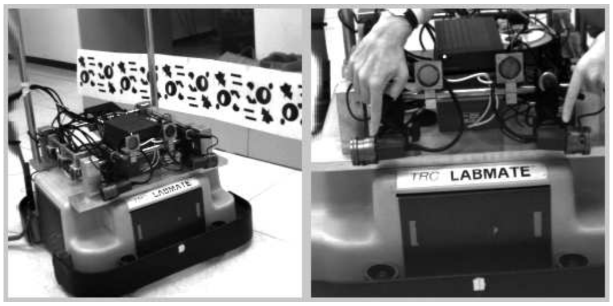

Figure 1: The vehicle with the divergent stereo camera setup

tical flow on each side. The optical flow $V = (u, v)$, is usually obtained by using the fundamental optical flow constraint [4], shown in equation (1), or by imposing second order stationarity constraints in the brightness function [5, 6], according to equation(2):

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

$$\frac{d}{dt}\nabla I = 0 \qquad (2)$$

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.

$$I_{xx}u + I_{xy}v = -I_{xt} \qquad (3)$$

$$I_{xy}u + I_{yy}v = -I_{yt} \qquad (4)$$

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.

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

The time delay, introduced with this procedure, may become important when the visual information is to be used to close a navigation control loop, as it introduces a lag in the feedback control loop.

In our approach, instead, having the control application in mind, we have used a *causal* first order time filter, which recursively updates the time smoothed image:

$$I_s(t) = \lambda I_s(t-1) + (1-\lambda)I(t) \qquad 0 \le \lambda \le 1 \quad (5)$$

$$I_s(0) = I(0) \qquad (6)$$

where $I(t)$ and $I_s(t)$ stand for the image acquired at time t and the corresponding *temporal smoothed* image. The parameter λ controls the degree of smoothing desired. When λ is larger, the images are more smoothed in time. Using the recursive time filtering procedure, only present and past images are required for the time filtering process. Regarding memory storage, it is only necessary to store $I(t)$, $I_s(t)$ and $I_s(t-1)$.

The experiments used to demonstrate the divergent stereo navigation concept, are based upon a mobile platform moving on a flat floor. As the robot motion is constrained to the ground plane, it can be assumed that the flow along the vertical direction will be negligible (unless there is a significant lateral motion, which is seldom the case). Hence, we can use a simpler computation procedure, which will be faster and robust (since it is not necessary to compute second order derivatives), by simply assuming that the vertical flow component, v, is 0:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \qquad (7)$$

$$v = 0 \qquad (8)$$

and u will be given by:

$$u = -\frac{I_t}{I_x} \qquad (9)$$

where I_x and I_t stand for the x spatial derivative and the temporal derivative of the image.

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

In order to be able to operate in a fast mode, we acquire a set of five images at video rate. The temporal smoothing of both the left and right image sequences, starts concurrently to the image acquisition. The last two images (on each side) of the time smoothed sequence, are then used to compute the average left and right optical flow vectors.

The difference between the average flow on the left and right images, is then used for synthesizing the control law. Then, a new sequence of 5 images is acquired, and the whole process is repeated.

In this way, the images are sampled at video rate although, the complete system may operate at a slower rate (determined by the computation time).

The images acquired have a resolution of 256x256 pixels, and the optical flow is computed over a window of 32x64 pixels in 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 it was explained before, in this work we are aiming 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. One could also consider the possibility of controlling the forward velocity of the robot, so that it could accelerate if the lateral flow is low (meaning that the walls are far away) and slow down whenever the flow becomes larger (while navigating in a narrow environment).

The flow difference is used as a misplacement error signal and we implemented a PID controller that actuates on the vehicle rotational speed.

As we are controlling the robot rotation velocity and we have not direct access to the position, it means that a sort of integral action is already present in the loop (in the sense that with a constant control input, the direction of motion would continuously increase along time, as the rotational motion would carry on at fixed speed). Therefore, by adding the integral action of the PID controller, one could not only expect that the vehicle responds slower but also, that it may become unstable.

In general the controller performs the following control law:

$$u = K_p[e(t) + K_i \int e(\tau)d\tau + K_d \frac{de(t)}{dt}] \qquad (10)$$

where u is the applied rotational speed (in degrees/s) and $e(t)$ stands for the *error signal* observed at time t, (which, for a vehicle moving along a corridor, correspond to the difference between the left and right average flow vectors).

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

3 Results

In this section we present a set of tests that clearly demonstrate several applications of the divergent stereo navigation approach. We will start by discussing a few open-loop examples to illustrate the capabilities of the perception process, and how the different situations are detected. Finally, we will show a tricky closed loop application where the whole control idea is applied.

3.1 Test 1 – Wall Follower

In this first example, we intend to show that the described measures can be used for a mobile robot to accomplish a goal such as following a wall, at a fixed distance. For this purpose, we have used a single camera ("single eye configuration") and the robot is moving at a roughly fixed distance from a wall, which has been covered with some texture panels (see figure 1). The vehicle speed is 80mm/s. We have performed tests with different types of textures and the system seems to operate well independently of the texture pattern, provided that there is some intensity variation.

The diagram of figure 2, shows how the experiment was performed, together with one image with the flow vectors superimposed. Figure 3 shows the evolution in time of the average flow field. Notice that the signal is fairly constant (the major perturbations in the averaged flow are probably due to changes in distance, as the vehicle is not moving exactly parallel to the wall, and also by drifts on the robot motion).

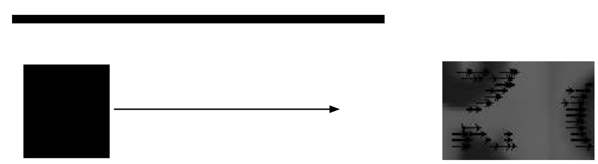

Figure 2: Setup for the wall follower experiment. On the right, a sample of the computed optical flow, containing approximately 400 flow vectors, is shown.

This sort of visual information could therefore be used to control the robot in order to navigate a a fixed distance from the wall, without involving an explicit reconstruction of the 3D environment, as it is done with more traditional approaches.

3.2 Test 2 – Obstacle Detection

Going further, the second test was aimed at studying the obstacle detection capabilities with the divergent stereo setup. Figure 4 shows the diagram describing how the experiment was conducted. We have placed an obstacle in the middle of the corridor in a

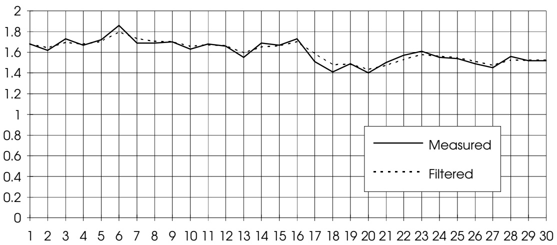

Figure 3: Wall follower. The evolution of the averaged flow (x component) in pixels per frame. The vehicle moves at constant speed along a wall. The full path lasts for 20 seconds, covering a distance of about 2m. The filtered signal, results from applying a first order filter to the measured values.

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.

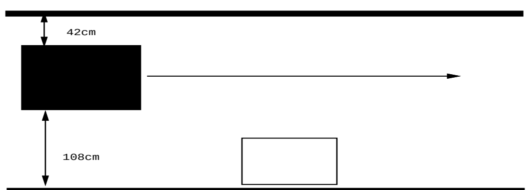

Figure 4: Obstacle detection experiment. The obstacle is approximately at the same distance of the robot as the left wall.

Figure 5 shows the signal corresponding to the difference between the left and right flow fields. As it was expected, the difference starts with a non-zero 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.

3.3 Test 3 – Closed Loop Operation

The final experiment has the goal of showing the closed loop behavior of the visually guided robot. For this purpose, we have considered an experiment which involves obstacle avoidance and navigating in a funneled corridor. Figure 6 shows the experiment diagram.

As described in section 2.3, we have used a PID controller. As it was expected we verified experimentally

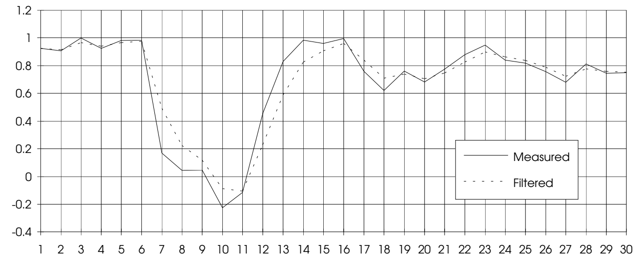

Figure 5: Difference between the left and right average flows, during the obstacle detection experiment. A smoothed version of the error signal is also shown.

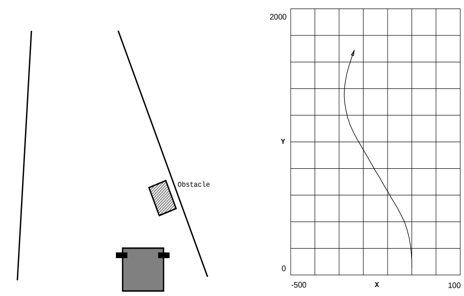

Figure 6: Closed loop operation. The left diagram shows the setup for the closed loop test. On the right we show the trajectory followed by the robot (the two diagrams are not drawn exactly on the same scale).

that the integral action of the controller often leads to slow response times and even to unstable behavior (overreacting in the turns). Instead, the differential component is used for increasing the response speed of the system. After some experiments, we found an acceptable behavior with the following controller settings: $K_p = 1.5, \quad K_i = 0.0, \quad K_d = 1.5$

The trajectory followed by the robot under the reflex-type closed loop control is shown in figure 6. Notice how the trajectory allows obstacle avoidance, and tends to a proper alignment later on.

In figure 7, we show both the error signal under the closed loop operation, and the control action applied to the robot.

Although the error and control signals present some sharp variations, it should be noted that the error is in the order of magnitude of 1 pixel per frame, while the changes in the rotation speed (that actually determine the direction of motion), are seldom larger than 3 deg/s. Finally, as it is shown in figure 6, the robot trajectory is quite smooth.

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 to the walls but also to the

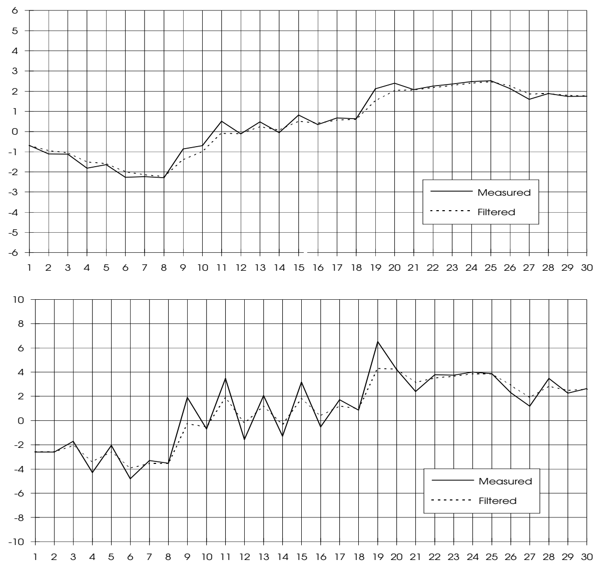

Figure 7: On top: the error evolution in time (difference between right and left average flow vectors). In the bottom, we show the control applied to the robot together with smoothed versions.

vehicle rotational velocity. This, of course, interferes with the simple control strategy adopted. If however, as in our case, the rotational velocity is kept small, this added error (which causes the system to perceive the wall contra-lateral to the direction of rotation as closer) does not degrade the performance significantly. This point, however, deserves a more carefull study and it is our intention to explain further details in a following paper how to implement a more general solution.

4 Conclusions

A qualitative approach to visually-guided navigation from qualitative optical flow measures has been presented, based on a "divergent stereo" camera placement.

A set of experiments illustrating the possibility of performing tasks such as wall-following, obstacle avoidance, corridor following and door's negotiation using a one-eye or two-eyes configuration is presented in the experimemental part.

An example of reflex-type control of motion driven by differential estimation of the flow field measured by the two eyes is presented. In particular it is shown how a difficult task like navigating through a narrow funneled corridor while avoiding an obstacle is possible without the need for accurate depth or motion estimation.

In a more extended paper [7] this kind of reactive control will be presented in more details and its extension to forward-velocity control and discontinuities in optical flow measures will be described. In this extended paper a comparison with a similar approach (which was published after our submission to CVPR) is also discussed [8]. [1]

References

[1] 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.

[2] G.A. Horridge. A theory of insect vision: velocity parallax. *Proceedings of the Royal Society of London, B*, 229:13–27, 1986.

[3] 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.

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

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

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

[7] 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.

[8] 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.

[1] **Acknowledgements:** The research described in this paper has been supported by the Special Projects on Robotics of the Italian National Council of Research and bu the Esprit Project VAP-2