Guidance principle and robustness issues for a biologically-inspired visual homing

G. Bianco, A.Rizzi, R. Cassinis, N. Adami
Dept. of Electronics for Automation
Via Branze 38, I-25123 Brescia - Italy
Phone +39-30-3715.457
E-mail: {bianco, rizzi, cassinis, adami}@ing.unibs.it

Abstract

This paper analyzes the guidance principle and the robustness feature of a biologically-inspired visual homing algorithm for autonomous robots. The homing strategy uses color images and is based on an affine matching model whose parameters are used to estimate real navigation displacement in the environment. The guidance principle of the visual homing is proven to be a visual potential function with an equilibrium point located at the goal position. The presence of a potential function means that classical control theory principles based on the Liapunov functions can be applied to assess the robustness of the navigation strategy.

1 Introduction

Thanks to entomological studies on social insects several mechanisms of visual navigation that can be effectively applied to robotics have been discovered, see e.g. [14, 15, 10, 16]. Among those mechanisms, the so called *homing* process offers interesting ideas for a computer-based implementation.

In order to return to known places, social insects and bees in particular, use two different navigation methods [22]: dead-reckoning and visual homing navigation. Dead-reckoning is exploited by extracting information about the homing vector using polarized light [20] and integration of image flow [16]. Visual homing is used in the final stage of the navigation, for a precise approach to the goal, when dead-reckoning does not have the required degree of precision.

According to experiments on ants and bees [7], insects have the ability to learn a place by storing a sort of snapshot of the panorama to be used later for the homing task. Insects remember not only the apparent dimension and position of objects surrounding a place but also their shape, pattern and color.

In particular, in [3] and [4] different computational models explaining how bees exploit the visual homing navigation have been proposed. In those models, through a comparison between the snapshot taken at the goal position and the actual image, bees compute the movements for a precise approach to the target.

Diverse theories have been introduced to explain the biological navigation models (see e.g. [19] for a review) but the snapshot hypotheses itself offers important starting points to implement navigation algorithms.

To this extent, extensions of the above models have been effectively used on robots in [2, 12], where a visual homing method based on a comparison between color images has been presented. The implemented method is composed of two phases: matching and navigation. In the matching phase the robot compares the stored snapshot of the place surrounding the goal with the snapshot perceived at the moment. In the navigation phase the differences in position and in dimensions between objects in the two images drive the robot to the pre-learnt goal. Images are acquired with a color camera and their comparison is performed by applying an affine model.

The aim of this paper is to analyze the underlying principle that drives the robot to the goal. To this extent, recent tests reveal the presence of a navigation vector field whose basin of attraction is placed at the goal position [5].

Confirmations of this can be found in the results of other experiments conducted on insects [21] or on artificial agents that use biologically-inspired navigation methods [6].

The theory of visual potential function introduced in [1] solves the problem of calculating the potential function that drives a homing method. The numerical calculation of the potential function starts considering the navigation vector field produced by the homing strategy. Considerations on the conservativeness of the field play a crucial role for the process of integration and hence for computing the potential function.

Furthermore, the presence of a potential function around the goal is a sufficient condition for the application of the classical control theory based on Liapunov functions. Therefore, formal considerations on the robustness for the homing method can be introduced. In addition, the idea of applying methods from vector analysis to navigation problems allows us to evaluate the performance of different models and might also represent the key point for the method to be further extended towards topological navigations.

The organization of this paper is as follows: review of the visual homing strategy (section 2); introduction of the visual potential function both theoretically (section 3) and numerically (section 5); formal considerations on the robustness of the method (section 4). Remarks and future work will conclude the paper.

2 Biologically-inspired visual homing

The implemented algorithm starts from the results presented in [4] and [23], with color images and some constraints in the image acquisition.

The main idea is that an estimate of the vector pointing from the current position of the agent to the pre-learnt goal can be computed comparing position and amplitude of matching areas in the considered images [3]. The matching between the goal image and the actual view is performed using an affine model. All possible affine transformations and shifts of the actual image in the allowed range are computed and the one that best fits the goal image is chosen. From the parameters of the affine transformation the algorithm computes an estimate of the robot displacement from the goal position. This has been interpreted as the distance of the robot from the goal.

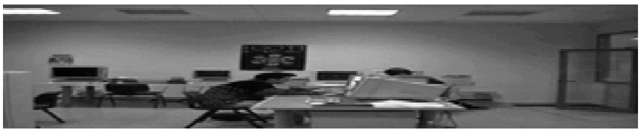

Figure 1: Vertically sub-sampled goal position image

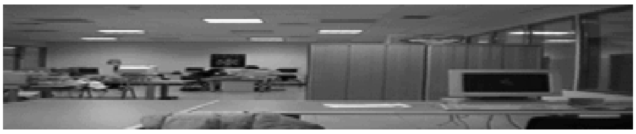

Figure 2: An example of a vertically sub-sampled starting point image

Examples of images used by the algorithm are shown in figure 1 and figure 2. They have been respectively acquired at the goal position and in a generic starting point of the environment. A decimation process has been applied in order to speed up the affine matching.

The above mentioned constraints, namely fixed heading and constant height of the camera, allow for the use of a simplified affine model, given by:

$$\begin{cases} S_X(X,Y) = a_{0X} + a_{1X} \cdot X \\ S_Y(X,Y) = a_{0Y} + a_{2Y} \cdot Y \end{cases} \tag{1}$$

where S_X and S_Y are the displacement components for the matching along x and y axes respectively; (X, Y) are the pixels co-ordinates, a_{0X}, a_{0Y} represent translations in pixels and a_{1X}, a_{2Y} represent expansions (a-dimensional). Furthermore, the camera has been always placed at the same height (about 1.2 m) and, due to the absence of the vertical camera movement, the term a_{0Y} is null. An apparent vertical shift in the image plane can be introduced by the change of object distance and is described by the expansion parameters a_{2Y}.

In the matching phase the algorithm finds the parameter values of the affine model that minimize the Mean Square Error (MSE) on the three chromatic channels of the whole image [11]. In order to speed up the parameter computations and at the same time allow the estimate of large displacement vectors, a multi resolution pyramidal technique has been implemented as in [23]. The ranges of the parameter estimate and the multi resolution levels are chosen according to the maximum displacement considered.

The computation of the real displacement components from the goal become possible with the following relationships:

$$\begin{cases} V_x = K \cdot a_{0X} \\ V_y = H \cdot a_{1X} \end{cases} \tag{2}$$

where V_x and V_y represent an estimation of the distance (along x and y axes of the environment) from the actual position to the goal position. The values of K and H are determined through an initial setting phase, performed by a series of known displacements around the goal position [12].

The system dynamical model is therefore given by:

$$\begin{cases} x(k+1) = x(k) + V_x(x(k), y(k)) \\ y(k+1) = y(k) + V_y(x(k), y(k)) \end{cases} \tag{3}$$

where $x(k)$ and $y(k)$ represent the coordinates of robot at step k; $V_x(x(k), y(k))$ and $V_y(x(k), y(k))$ the displacements computed at step k following equation 2. Those displacements are related to the position at step k given by $(x(k), y(k))$. Lastly, $x(k+1)$ and $y(k+1)$ represent the new positions the robot will move to. Clearly, an important equilibrium point (x^*, y^*) for the system is given by the coordinates of the goal position.

In figure 3 the directions of the estimated displacement vectors $\vec{V} = [V_x \; V_y]$ for the first navigation steps from a group of starting points are shown. The goal (represented by a small circle) seems to be located within a basin of attraction.

The results of a complete navigation process from each given starting point can be seen in figure 4. The navigation phase is completed after two steps, with a mean error of about 5 cm along a path of about 720 cm, less than 0.7 %.

The acquired images can be affected by noise. The major error contributions come from moving objects (people walking, etc.) and from occlusions that cause a punctual error higher than the average.

For this reason in the final matching phase the MSE on the whole image is substituted by a new MSE (MSE Enhanced - MSEE) computed only on the pixels whose values are lower than the MSE previously computed. This method enhances the robustness of the navigation as detailed in section 5.

A simple flow chart of the described method can be seen in figure 5.

3 The visual potential function

A field of forces $\vec{V}(q)$ can be defined as *potential* when it is produced by a differentiable function U (called *potential energy*) with:

$$\vec{V}(q) = \vec{\nabla} U(q) \tag{4}$$

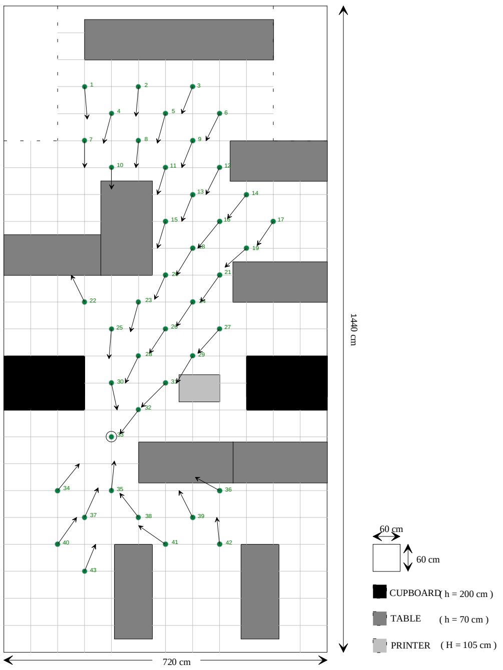

Figure 3: An example of a navigation field computed

where q is the actual *configuration* of the robot [8]. Classically, both U and q are mathematically specified. The actual configuration q of the robot depends on the actual position of the robot: in our case x, y and θ. To simplify the process all tests have been performed considering a fixed θ, as detailed in the previous section. Therefore, from equations 2 and 4 and supposing to deal with continuous-time. smooth and differentiable V_x and V_y, it can be stated that:

$$\vec{V} = [V_x(x,y) \; V_y(x,y)] = \left[\frac{\partial U(x,y)}{\partial x} \; \frac{\partial U(x,y)}{\partial y} \right] \tag{5}$$

A necessary and sufficient condition for the (unique) integration of the vector field is that the following relation holds [17]:

$$\frac{\partial V_x(x,y)}{\partial y} = \frac{\partial V_y(x,y)}{\partial x} \tag{6}$$

called the *Cauchy-Riemann* condition. In other terms:

$$\frac{\partial V_x(x,y)}{\partial y} - \frac{\partial V_y(x,y)}{\partial x} = 0 \tag{7}$$

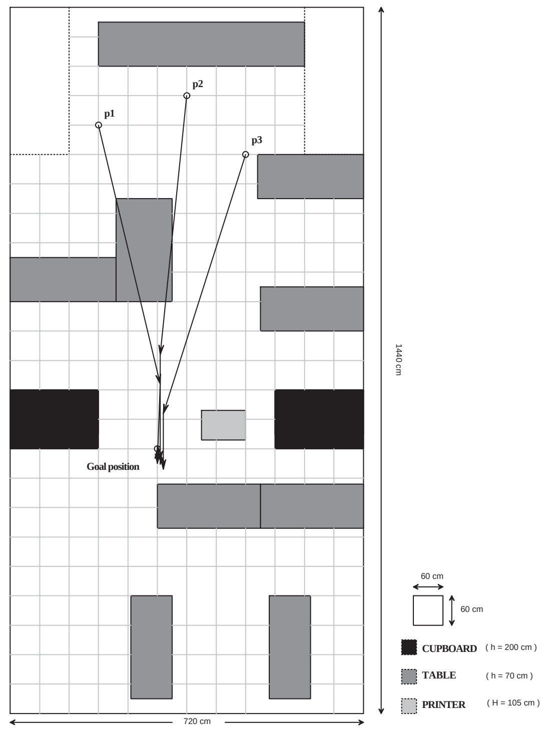

Figure 4: Complete navigation test

can be alternately referred with the term *conservativeness.*

Now, for the integration of equation 4, let c be a curve in the set of definition of the equation (our environment) and suppose that $U(x, y)$ is a continuously differentiable function; in addition, let x and y depend on parameter t, that is $x = x(t)$ and $y = y(t)$ so that the curve c is parameterized on t (i.e. $c = c(t)$); the same holds for the directional vector $\vec{r}$. It can be stated that [13]:

$$\int_c \nabla U(t) \circ d\vec{r}(t) = U(x(b), y(b)) - U(x(a), y(a)) \quad (8)$$

where a and b are the values of t which represent respectively the initial and end points of the curve c. Therefore, the potential function $U(x, y)$ can be calculated by integrating equation 4 following a given curve c.

If the vector field $\vec{V}$ is conservative then the integral is path independent [13]. Therefore, performing the substitution detailed in equation 4 and computing the scalar product, the following is obtained:

$$U(x, y) = \int_c V_x(x, y)dx + V_y(x, y)dy \qquad (9)$$

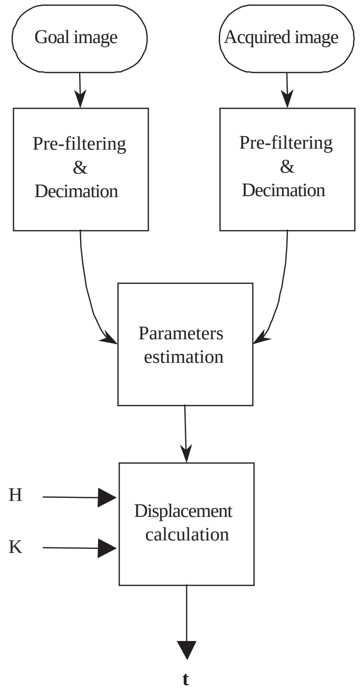

Figure 5: Flow chart of the algorithm

The process can be further simplified by following a *particular* curve c. Therefore, more effectively, it can be written as:

$$U(X, Y) = \int_{p_x}^{X} V_x(x, p_y)dx + \int_{p_y}^{Y} V_y(X, y)dy \quad (10)$$

where $U(X, Y)$ is the potential function and the path of integration is along the horizontal line segment from the initial point (p_x, p_y) to the vertical line through (X, Y) and then along the vertical line segment to (X, Y).

Letting the initial point coincides with the coordinates of the goal position helps the computing of the potential function. Every other point is thus referred to in terms of potential to the goal position.

3.1 Issues on the non-conservativeness of the field

The above introduced theory holds for conservative fields but for the purpose of this paper some considerations on non-conservative fields must be made.

Formally speaking, a non-conservative field is one in which the circuitation is non-null. That is, the calculation of $U(x, y)$ depends on the path followed and equation 8 produces different results according to the curve that was chosen.

This potential function can be considered as *multi-valued*: for every reference position, a general position will have more than one potential value according to the path chosen for integration. This sort of multi-valued potential function can be translated in *non-uniqueness of the vector field*. In other words, repeatedly placing the robot in the same point within a non-conservative area will lead to different paths that will be followed.

It is hard to state what kind of environmental or system feature might be responsible for such non-conservative nature of the field. Theoretically, however, as the field is neither stable nor reliable, the repeatability of the navigation does not hold and this can lead to unpredictable navigation results.

4 Robustness of the method

In order to take advantage of the discussion presented in the previous section and supposing that all the necessary hypotheses held, the dynamic system presented in equation 3 is considered continuous-time with the following (leaving out the vector notations):

$$\dot{x}(t) = V(x(t)) \tag{11}$$

where x represent the generic coordinates and an equilibrium point x^* is located at the goal position.

Several important considerations for the stability of the system can be expressed focusing attention on its properties.

The basic idea for verifying the stability and the convergence is to seek an aggregate summarizing function on the states of the system. The underlying principle is to simplify the analysis of a complex high-order dynamic system by considering a single scalar-valued function whose time behavior can be estimated [9].

In particular, when a dynamic system can be represented by $\dot{x} = f(x)$ with a fixed point x^*, and it is possible to find a *Liapunov function*, i.e. a continously differentiable, real-valued function $U(x)$ with the following properties [18]:

1. $U(x) > 0$ for all $x \neq x^*$ and $U(x^*) = 0$
2. $\dot{U} < 0$ for all $x \neq x^*$ (all trajectories flow *downhill* toward x^*)

then x^* is globally stable: for all initial conditions $x(t) \to x^*$ as $t \to \infty$.

There are different types of system configurations: *stable, asymptotically stable, marginally stable* and *unstable*, all of which are explained in terms of trajectories followed by system states. In those cases Liapunov criteria still exist which relax the characteristics of $U(x)$ previously listed.

The system depicted in equation 11 is of type $\dot{x} = f(x)$ but, unfortunately, there is no systematic way to construct Liapunov functions.

For example, the quadratic distance of the robot to the target (as computed by equation 3) might be a good candidate to have the nature of a Liapunov function. But, it can be taken advantage considering the effects of the system, that is the vector field it produces (see figure 3). This represents the only way to perform such analysis as we cannot predict the analytical form of $\vec{V}$ on $x(t)$ as depicted by equation 2.

To this extent, an important Liapunov function can be constructed by integrating the right-hand side of the system equation 11.

This automatically leads us to introduce the visual potential function as a good candidate for analyzing the overall stability of the system.

If the visual potential function had a basin of attraction whose the minimum is at the goal position then the theory of Liapunov functions for the method assures that homing is intrinsically stable, at least starting navigating from part of the environment.

In addition, the intrinsic stability of the system would mean that the necessity of an accurate calibration phase for determining H and K does not hold anymore. Experiments reported in [2, 12] can be considered this way. Moreover, H and K could not probably be strongly related to a particular target, robot and environment.

The practical computation of the visual potential function along with the stability analysis of the system being considered will be detailed in the following section.

5 Tests on conservativeness and calculation of the visual potential function

In section 3 it has been stated how conservative issues are crucial to calculate a unique potential function starting from a vector field. As it was said, for every vector field measured during an experiment it is necessary to compute the cross derivatives and evaluate the Cauchy-Riemann condition.

To this extent, figure 6 plots the components V_x

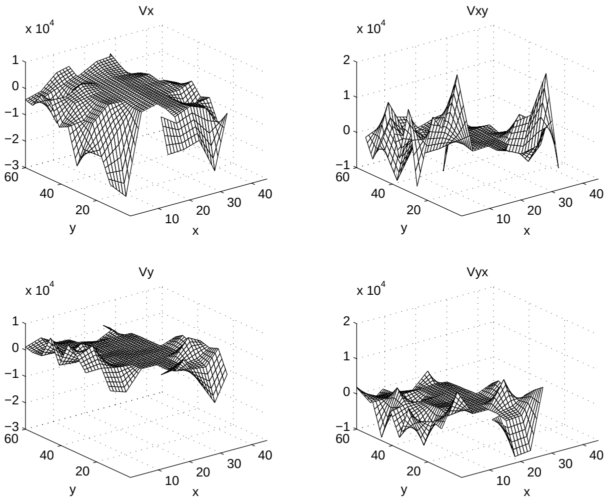

Figure 6: An example of V_x and V_y and their cross derivatives

and V_y and their cross derivatives $\frac{\partial V_x}{\partial y}$ and $\frac{\partial V_y}{\partial x}$ taken from an effective test. The cross derivatives have been computed numerically as expressed in [17].

All the tests have been performed in the same environment reported in figure 3 and the goal position is located in $(20, 30)$. The point with co-ordinates $(0, 0)$ is placed at the top-right corner of figure 3.

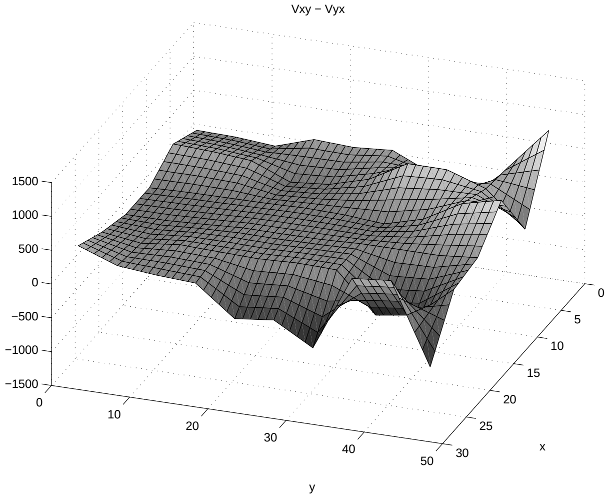

Figure 7: Conservativeness of the navigation field adopting MSE matching

The robot has been manually placed at various points in the environment. From those points, applying the method detailed in equation 2, a displacement vector is computed. The iteration of the method over the whole environment and the collection of every displacement vector produces a vector field, as previously exemplified by figure 3.

In figure 7 the conservativeness of the navigation field computed adopting the MSE matching is shown. The great part of the field can be considered conservative. However, there are some regions of the environment where that condition does not hold, therefore this affects the repeatability of the navigations starting from those areas. In fact, if the vector field is non-conservative then the potential function computed is not unique.

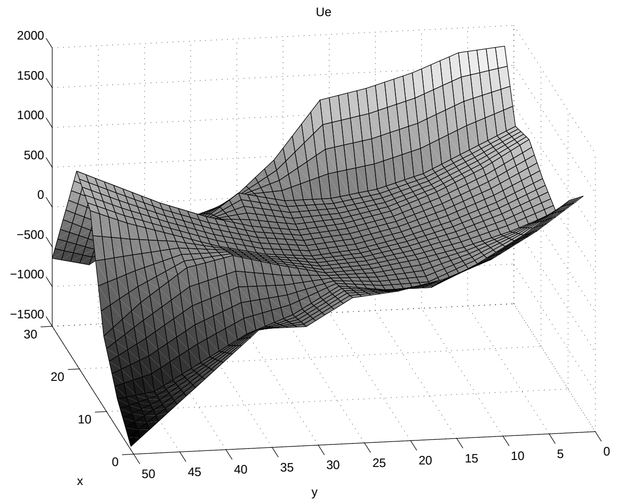

Figure 8: The visual potential function adopting MSE matching. The goal position is located in $(20, 30)$.

The visual potential function that is calculated by integrating the field as detailed in equation 10 is shown in figure 8.

There are large areas of the environment from which the navigation phase can be successfully performed. Nevertheless, starting from other areas the robot gets trapped by *false* goals. Those regions roughly correspond to the bottom-right part of the environment reported in figure 3.

The same calculations can be done by considering the MSEE matching between images. The conservativeness of the field is reported in figure 9 and the visual potential function is shown in figure 10.

The stability is not guaranteed for the whole environment, as can be seen in the figures previously reported. But the potential basin is better shaped than with simple MSE and the goal position is more stable. Furthermore, the use of MSEE instead of MSE affects the conservativeness of the field. In addition, MSEE enlarges the area (see figure 10 compared to figure 8) from which the goal can be reached.

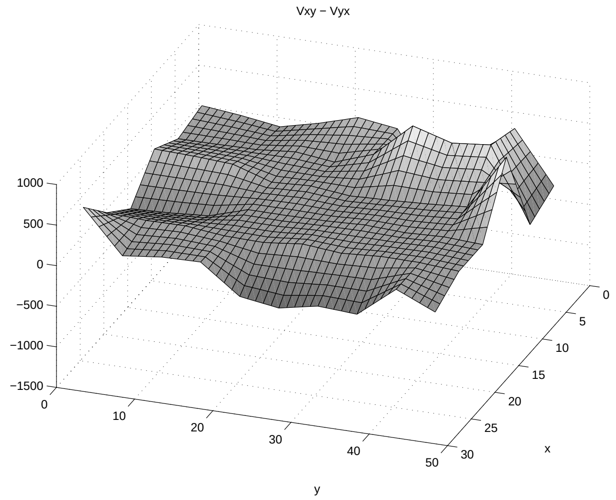

Figure 9: Conservativeness of the navigation field adopting MSEE matching

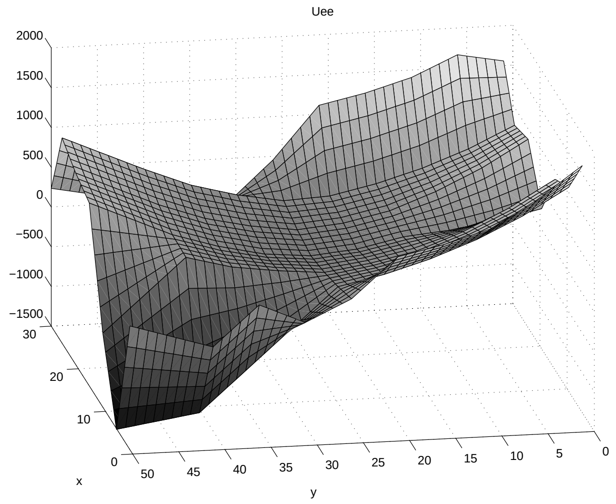

Figure 10: The visual potential function adopting MSEE matching. The goal position is located in $(20, 30)$.

But, either MSE or MSEE cannot deal with the intrinsic limitations of the homing strategy when the robot is placed in some regions of the environment (in figures 8 and 10, when $x \leq 10$ and $y \geq 35$). The use of a camera with a limited field of view can lead to bad navigation results: the robot is attracted to *false* goals because the actual view might not contain enough information for a good comparison with the stored snapshot taken at the goal position. This can be overcome by acting either on the matching method or, more likely, by widening the camera field of view.

5.1 Issues on the visual potential as a Liapunov function

From the potential function previously plotted it can be easily understood why the system gets sometimes trapped into false goals or what can be the region of convergence for the main goal position.

This implicitly states that the system has not overall stability on the whole environment. Therefore, the visual potential function itself cannot be considered Liapunov compliant unless reducing its domain of application to a region around the goal position, starting from which the system converges.

This restricted visual potential function can be formally adopted as a Liapunov function, though in a restricted area of the environment.

In addition, the parameters H and K can be suitably computed to allow for the system to be better convergent but this needs to be further investigated.

6 Conclusions and perspectives

This paper has shown how the presented biologically-inspired method for navigating can be explained in terms of a visual potential field. The conservativeness of the navigation vector field can be affected by different matching or vision enhancement algorithms.

The presence of a potential function around the goal allows us to apply classical control theories to assess the robustness of the system such as the Liapunov functions.

The use of MSE or MSEE alone is not sufficient to produce a potential function whose profile is globally directed to the goal. To this extent, a panoramic field of view is thought to be necessary to produce well-shaped potential basins where the goal is the only global minimum. This aspect has to be further investigated as well as the relation of the introduced model with others that consider different aspects of the environment such as landmarks.

The presence of a potential function is considered to be important for further extending the method: subgoals can be automatically placed at the boundaries of the basin of attraction.

The idea of applying methods from vector analysis to navigation problems allows us to evaluate the performance of different models and can represent an important step for topological navigations.

References

[1] G. Bianco. *Biologically-inspired visual landmark learning and navigation for mobile robots.* PhD thesis, Department of Engineering for Automation, University of Brescia (Italy), December 1998.

[2] G. Bianco, R. Cassinis, A. Rizzi, N. Adami, and P. Mosna. A bee-inspired robot visual homing method. In *Proceedings of the Second Euromicro Workshop on Advanced Mobile Robots (EUROBOT '97)*, pages 141–146, Brescia (Italy), October 22-24 1997.

[3] B. Cartwright and T. Collett. Landmark learning in bees. *Journal of Comparative Physiology*, A(151):521–543, 1983.

[4] B. Cartwright and T. Collett. Landmark maps for honeybees. *Biological Cybernetics*, (57):85–93, 1987.

[5] R. Cassinis, A. Rizzi, G. Bianco, N. Adami, and P. Mosna. A biologically-inspired visual homing method for robots. In *Proceedings of seventh workshop of AI*IA on Cybernetic and Machine Learning*, Ferrara (Italy), April 1998.

[6] P. Gaussier, C. Joulain, and J. Banquet. Motivated animat navigation: a visually guided approach. In *Simulation of Adaptive Behavior*, Zurich (Switzerland), August 1998.

[7] J. Gould. Landmark learning by honey bees. *Animal Behaviour*, (35):26–34, 1987.

[8] J. Latombe. *Robot motion planning.* Kluwer Academic Publisher, Boston/Dodrecht/London, 1991.

[9] D. Luenberger. *Introduction to dynamic systems - theory, models, and applications.* John Wiley and Sons, New York Chichester Brisbane Toronto, 1979.

[10] U. Nehmzow. Animal and robot navigation. *Robotics and Autonomous Systems*, 15:71–81, 1995.

[11] A. Netravali and B. Haskell. *Digital Pictures-Representation and Compression.* Plenum Press, New York, 1988.

[12] A. Rizzi, G. Bianco, and R. Cassinis. A bee-inspired robot navigation using color images. *Robotics and Autonomous Systems*, (25):159–164, 1998.

[13] C. Ross. *Differential Equations An Introduction with Mathematica.* Springer-Verlag, New York Berlin Heidelberg London Paris Tokyo Hong Kong Barcelona Budapest, 1995.

[14] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: learning from bees. In *IEEE Computer Society Conference on Computer Vision e Pattern Recognition*, New York, 1993.

[15] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation: from bees to robots. *International Journal of Computer Vision*, (14):159–177, 1995.

[16] M. Srinivasan, S. Zhang, M. Lehrer, and T. Collett. Honeybee navigation en route to the goal: visual flight control and odometry. *Journal of Experimental Biology*, (199):155–162, 1996.

[17] G. Strang. *Introduction to Applied Mathematics.* Wensley-Cambridge Press, 1986.

[18] S. Strogatz. *Nonlinear dynamics and chaos.* Addison-Wesley Publishing Company, New York, 1994.

[19] O. Trullier, S. Wiener, A. Berthoz, and J. Meyer. Biologically based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544, 1997.

[20] K. von Frisch. *Bees. Their Vision, Chemical Senses and Language.* Cornell University Press, Ithaca (NY), 1971.

[21] W. Voss. *Information durch Eigenbewegugn: Rekonstruktion und Analyse des Bildflusses am Auge Fliegender Insekten.* PhD thesis, Eberhard-Karls-Universitat Tubingen, 1995.

[22] R. Wehner. Arthropods. In F. Papi, editor, *Animal Homing*, pages 45–144. Chapman and Hall, London, 1992.

[23] T. Wittman. Insect navigation: Models and simulations. Technical report, Zentrum fur Kognitionswissenschaften, University of Bremen, 1995.