Reprinted from

Robotics and Autonomous Systems

Robotics and Autonomous Systems 25 (1998) 159–164

A bee-inspired visual homing using color images

A. Rizzi *, G. Bianco [1], R. Cassinis [2]

Department of Electronics for Automation, University of Brescia, Via Branze 38, I-25123 Brescia, Italy

ELSEVIER

ELSEVIER

Robotics and Autonomous Systems 25 (1998) 159–164

Robotics and Autonomous Systems

A bee-inspired visual homing using color images

A. Rizzi *, G. Bianco [1], R. Cassinis [2]

Department of Electronics for Automation, University of Brescia, Via Branze 38, I-25123 Brescia, Italy

Abstract

This paper presents a visual homing algorithm for autonomous robots inspired by the behaviour of bees and other social insects. The homing method presented is based on an affine motion model whose parameters are estimated by a best matching criterion. No attempts are made to recognize the objects or to extract 3D models from the scene. Improvements in the algorithm and in the use of colour information are introduced in order to enhance the efficiency of the navigation vector estimate. Tests and results are presented. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Mobile robots; Robot navigation; Visual self-localization; Landmark recognition

1. Introduction

Entomological studies about social insects (in particular bees, ants and wasps) have discovered some mechanisms of visual navigation that can be useful in robotics [4,7–9]. Several interesting considerations can be made regarding the ability of many insects to return to the precise locations for foraging or for finding home [12].

According to experiments on ants and bees [1], an insect fixes the location of objects surrounding a place by storing a sort of snapshot of the panorama taken from that place. The snapshot of the environment does not encode explicitly the distance between objects or between objects and target. Other experiments on bees proved that the insects remember not only the apparent dimension and position of objects surrounding a place [1], but also their shape, pattern and colour [3].

In order to return to specific places, social insects and bees in particular, exploit two different navigation methods: dead-reckoning and visual homing navigation [12]. The dead-reckoning method is exploited by extracting information about the homing vector using polarized light [11] and integration of image flow [9]. The visual homing method 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.

Interesting computational models introduced to explain how bees exploit the visual homing navigation were proposed in [1] and subsequently refined in [2]. The results of these models, simulated on a computer, strongly resemble the actual behaviour of bees. In the above models, bees learn the goal point position taking a black and white picture of the place. Subsequently, through a comparison between the goal snapshot and the current image, bees compute the movements for a precise approach to the target.

The visual homing method presented in this paper, follows the same approach and can be divided in to two phases:

- *Matching phase*. The bee compares the stored snapshot of the place surrounding the goal with the snapshot perceived at the moment.

* Corresponding author. E-mail: rizzi@bsing.ing.unibs.it

[1] E-mail: bianco@bsing.ing.unibs.it

[2] E-mail: cassinis@bsing.ing.unibs.it

0921-8890/98/$ – see front matter

PII: S0921-8890(98)00045-1

- *Navigation phase.* The differences in position and in dimensions between objects in the two images drive the bee towards the goal.

Both the matching and the navigation phases have been implemented and tested. The algorithm is presented in the following section and the tests are presented in section 3.

2. Matching and navigation

As presented, the implemented algorithm starts from the Cartwright and Collett [2] and Wittman [13] results introducing the use of colored images. In order to efficiently use the colour images of the environment with large amounts of information and complexity, some constraints have been introduced:
- the heading of the camera is kept constant by some external means;
- the algorithm can only be used in the proximity of the target (when some parts of its neighbourhood are visible).

As in [13], in fact, the images have been collected with the same orientation. This is done by keeping the camera with the same heading during the navigation task. Moreover, the robot navigates in a flat indoor environment so that the camera height is constant. This allows further simplifications in the affine model, discussed later.

The main idea is that an estimate of the vector pointing from the current position to the goal could be computed comparing positions and amplitude of matching areas in the considered images [1]. This vector derives from a matching between the goal image and the actual one, reconstructed using an affine motion model. All the 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 this transformation the algorithm computes an estimate of the robot displacement from the goal position.

2.1. *The projection on the camera plane and the affine model*

The relationship between objects in the environment and their positions in the grabbed images is described by the following camera model derived from the central projection reference system [10], as shown in

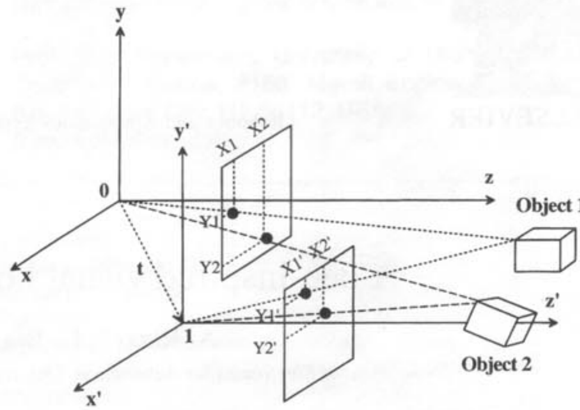

Fig. 1. Object projection shifts due to camera movement.

Fig. 1. Let p be a point of the environment and P its projection on the camera image plane. The relationship between the two points is:

$$X = F \cdot \frac{x}{z}, \qquad Y = F \cdot \frac{y}{z},$$

where X and Y are the co-ordinates of the P point in the image plane, x, y, z the co-ordinates of the p point in the 3D space and F is the focal length.

An affine model that takes into account translations, compressions and deformations of the object projections in the image plane caused by a camera translation is described by the following equations:

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

where S_X and S_Y are the displacement components in x and y directions, respectively, (X, Y) the pixel co-ordinates, a_{0X} and a_{0Y} the translation parameters and a_{1X}, a_{2X}, a_{1Y}, a_{2Y} are the deformation parameters.

2.2. *The implemented model*

The above mentioned constraints, fixed heading and constant height, allow the use of a simplified model. In the considered environment a large amount of objects (tables, printers, shelves, etc.) are fixed, and it is then reasonable not to take into account any object rotation. Therefore, the model becomes

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

where a_{0X}, a_{0Y} represent translations in pixels and a_{1X}, a_{2Y} represent expansions (a-dimensional). Moreover, 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 parameter a_{2Y}.

2.3. The matching algorithm

In the following step, the homing algorithm finds the parameter values that minimize the following mean square error (MSE) on the whole image [5]:

$$\mathrm{MSE} = \frac{1}{M} \sum_{(X,Y)\in \mathcal{S}} E_r(X,Y) + E_g(X,Y) + E_b(X,Y),$$

with

$$E_{r,g,b}(X,Y) = [I_{1(r,g,b)}(X,Y) - I_{2(r,g,b)}(X+S_X, Y+S_Y)]^2,$$

where M is the number of couples $(X+S_X, Y+S_Y)$ still in the image plane $\mathcal{S}$, I_1 and I_2 the images, r, g and b the chromatic components and S_X and S_Y are the estimated displacement vectors for each pixel.

In order to speed up the parameters computation and at the same time allow the estimate of large displacement vectors, a multi-resolution pyramidal technique has been implemented as in [13]. According to this technique the image at level I is obtained by subsampling by a factor 2 the image at level I-1. The displacement estimate starts from the image at a lower resolution going up the pyramid by maintaining the dimensions of the estimate intervals unchanged. The parameters estimated at level I are subsequently used at level I-1 as offsets for the relative estimate ranges. Each subsampling operation is preceded by a low pass filtering, with the aim of excluding possible spatial aliasing. The filter used is a seven coefficients Gaussian filter [6].

The estimate ranges and the multi-resolution levels are chosen according to the maximum displacement considered. It is possible to calculate the displacement components according to the relationships

$$\Delta_x = K \cdot a_{0X}, \qquad \Delta_z = H \cdot a_{1X}.$$

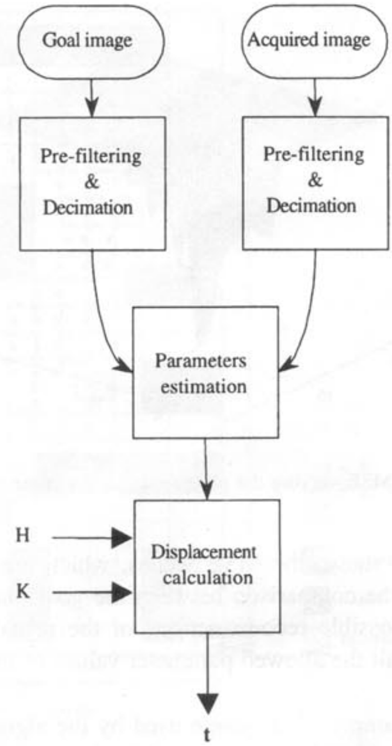

Fig. 2. Flow chart of the algorithm.

The values of K and H are determined for a more efficient navigation through an initial setting phase, performed by a series of known displacements around the goal position. Using the whole image and not a single object in the scene, the values of H and K are mean values concerning all the objects composing the scene. Due to the null value of the vertical shift, vertically decimated images of a factor $2k$ have been used. A simple flow chart of the described method can be seen in Fig. 2.

The noise in the acquired images is mainly due to the following factors: people moving in the room, changes of displays on computer monitors and lights being switched on and off. The major error contribution in the matching of two images acquired in the same point comes from the 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 over the whole image is substituted by a new MSE computed only over the pixels lower than the first MSE previously computed.

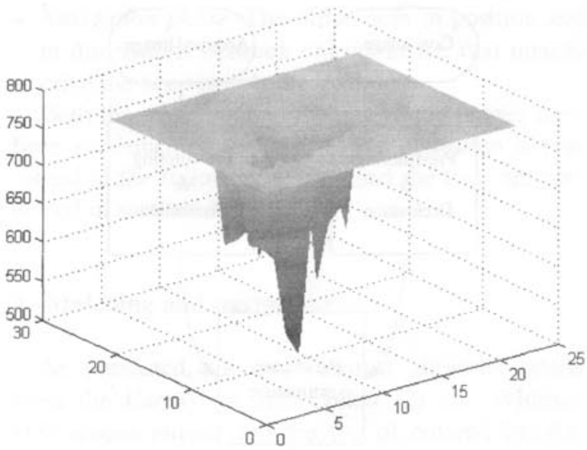

Fig. 3. MSE varying the parameters of the affine model.

Fig. 3 shows the MSE values, which are the results of the comparison between the goal image and all the possible reconstructions of the actual image through all the allowed parameter values of the affine model.

An example of an image used by the algorithm is shown in Fig. 4. It is acquired in the goal position. The image in Fig. 5 is obtained by low pass filtering and decimating image in Fig. 4. Image in Fig. 6 is obtained by low pass filtering and decimating an image acquired in a generic starting point of the navigation area.

3. Tests

The proposed model has been tested in two different ways: complete navigations to the goal position from three randomly chosen starting points and the computation of the first navigation step from a grid of points in the test area.

In all tests, the values of parameters H and K, after the model calibration, are $H = -1.8\,\text{cm/pixel}$ and $K = 680\,\text{cm}$. Anyhow an accurate estimate of the parameters H and K is not a necessary step for the navigation. Their values affect the navigation vector, and this influences the number of steps required to reach the goal position, not the navigation convergence to the target.

The following values have been chosen for the parameters increment in the pyramidal algorithm:

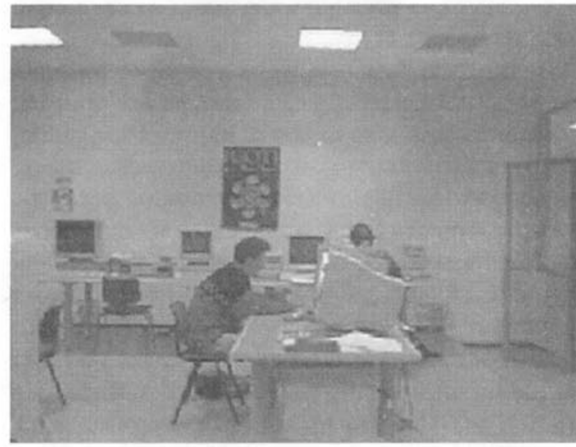

Fig. 4. Goal position image.

Fig. 5. Vertically subsampled goal position image.

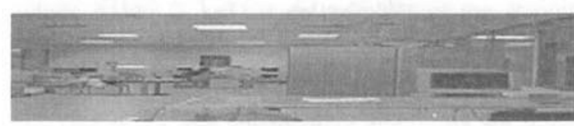

Fig. 6. Vertically subsampled starting point image.

$\Delta a_{0X} = \pm 3[\text{pixel}]$

with increment $da_{0X} = 1$ [pixel],

$\Delta a_{1X} = \pm 0.5[\text{pixel}]$

with increment $da_{1X} = 0.05$ [pixel].

With these values the system resolution is ± 32.38 cm on Z-axis and ± 1.71 cm on X-axis in an area of 720×1440 cm. The image obtained by the camera is 384×288 pixels, the vertical under sampling is $4:1$ and the pyramidal structure used has five levels.

The results of a complete navigation process from each given starting point can be seen in Fig. 7. Navigation errors are reported in Table 1: after two navigation steps they are below the maximum admissible error in centimeters for the selected parameter increment. 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%.

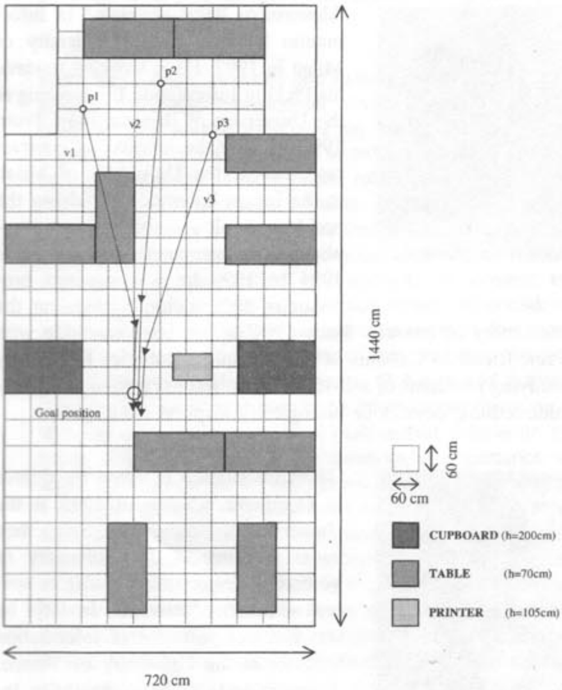

Fig. 7. Complete navigation test.

Table 1
Test errors

Navigation	Steps	Error
P1	2	1.8
P2	2	8.1
P3	2	10.9

Vectors V_1, V_2 and V_3 represent estimated direction and distance of the goal from the starting points p_1, p_2 and p_3. Estimates are not very accurate and lead to points that can be quite far from the actual goal, but, if the process is iterated, it quickly converges to the target position. This algorithm does not take into account obstacles: it only estimates the relative position of the goal with respect to the robot. Actual trajectories should be computed using an obstacle avoidance method.

In Fig. 8 the directions of the estimated displacement vectors for the first navigation steps from some starting points are shown. Almost all points show good navigation behaviours. E.g., at point 32 a cupboard

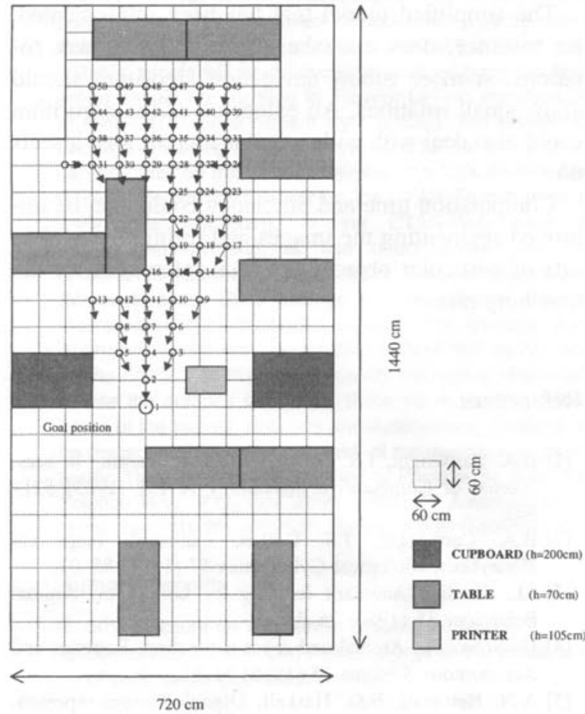

Fig. 8. Directions of every estimate displacement vector (vector lengths are not shown).

occludes a consistent part of the image grabbed from that point so the matching phase fails. A wider image could overcome this situation.

4. Conclusion and perspectives

A visual homing method has been presented. This biologically inspired method, starting from some well-known models, implements a robust algorithm to estimate the relative position between the homing place and the current robot position.

Particular attention has been paid to the use of the method in real environments with small changes in the object disposition, moving people, etc. To deal with these situations a recomputation of the affine model parameter estimate is done. No environment conditioning is required.

The tests have shown good results in driving the robot towards a goal position, but several improvements are still possible.

The simplified model that has been implemented, for instance, does not take into account camera rotations. A more robust navigation algorithm should allow small rotations. An extension of the algorithm could also deal with wide angular images, like insects do.

Computation time and efficiency could also be improved segmenting the images and trying to correlate sets of particular objects as visual references for the matching phase.

References

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

[2] B.A. Cartwright, T.S. Collett, Landmark maps for Honeybees, Biological Cybernetics 57 (1987) 85–93.

[3] J.L. Gould, Landmark learning by honeybees, Animal Behaviour 35 (1986) 26–34.

[4] U. Nehmzow, Animal and robot navigation, Robotics and Autonomous Systems 15 (1995) 71–81.

[5] A.N. Netravali, B.G. Haskell, Digital Pictures-representation and Compression, Plenum Press, New York, 1988.

[6] W.K. Pratt, Digital Image Processing, Wiley/Interscience, New York, 1991.

[7] J. Santos-Victor, G. Sandini, F. Curotto, S. Garibaldi, Divergent stereo for robot navigation: Learning from bees, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, 1993.

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

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

[10] Y.T. Tse, R.L. Baker, Camera zoom/pan estimation and compensation for video compression, SPIE Image Processing Algorithms and Techniques II, vol. 1452, 1991 pp. 468–479.

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

[12] R. Wehner, Arthropods, in: F. Papi (Ed.), Animal Homing, Chapman & Hall, London, 1992, pp. 45–144.

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

Alessandro Rizzi graduated in Information Science at the University of Milan in 1992. He is working towards his Ph.D in Information Engineering at the University of Brescia, Italy. From 1992 to 1994 he worked as contract reseacher at the University of Milan in the Image Laboratory, studying the problem of colour constancy and its application in computer graphics. From 1994 to 1996 he was contract professor of Information Systems at the University of Brescia. Since 1994 he has been woriking with Prof. Riccardo Cassinis at the Advanced Robotics Laboratory studying problems of self-localization and navigation using omnidirectional devices or biologically inspired models.

Giovanni Bianco received his degree in Computer Science in 1989 at the University of Udine, Italy. Since then he is working at the University of Verona as project team leader in several automation activities. In 1994 he was contract professor of Information Systems at the University of Verona. Actually, he is a Ph.D. student in Information Engineering at the University of Brescia. From January to June 1998 he has been at the Australian National University, Canberra, as a research fellow working on autonomous landmark navigation and self-extraction of salient landmarks as part of his Ph.D. He has been working for about 10 years on several topics related to autonomous robots, and is now involved in landmark visual navigation starting from biological models.

Riccardo Cassinis received his degree in Electronic Engineering in 1977 at the Polytechnic University of Milan. In 1987 he was appointed as Associate Professor of Robotics and of Numerical Systems Design at the University of Udine. Since 1991 he is an Associate Professor of Computer Science and of Robotics at the University of Brescia. He has been Director of the Robotics Laboratory of the Department of Electronics in Milan, of the Robotics Laboratory of the University of Udine, and is now Director of the Advanced Robotics Laboratory of the University of Brescia. He has been working for about 15 years on several topics related to industrial robots, and is now involved in navigation and sensing problems for advanced mobile robots.