Proceedings of the Second

Euromicro Workshop on Advanced Mobile Robots (EUROBOT '97)

Brescia, Italy
October 22–24, 1997

COMPUTER SOCIETY

A Bee-Inspired Robot Visual Homing Method

G. Bianco, R. Cassinis, A. Rizzi, N. Adami, P. Mosna

Dept. of Electronics for Automation - University of Brescia
Via Branze 38, I-25123 Brescia - Italy
Phone +39-30-3715.453
E-mail: {bianco, cassinis, rizzi}@bsing.ing.unibs.it

Abstract

This paper presents a proposal for a visual homing algorithm inspired by the behaviours of social insects. The homing method presented is based on an affine motion model which parameters are estimated by a best matching criteria. In the matching phase no attempts are made to recognise objects or to extract 3D models of the scene. Hypotheses and perspectives about the use of single landmarks by bees are introduced. Tests and results are presented.

1.- Introduction

A navigation task is normally composed of two phases: a coarse approach to the goal and a precise positioning to it (homing).

Entomological studies about social insects (bees, ants, etc.) have discovered some mechanisms of visual navigation and landmarks use that can be useful in robotics [Santos-Victor et al. 93, 94]. In fact, several interesting considerations could be made regarding the ability of many insects to return to precise locations for foraging or for finding home [Wehner 92].

According to experiments, the strategy used by bees in order to be able to reach a known point can be summarised in the following two points:

- bees store images (snapshots) and remember the apparent dimension and the position of landmarks [Cartwright, Collett 83] surrounding a place;
- bees remember the shape, the pattern and the colour of a landmark [Gould 86].

Experiments on ants and bees suggested in fact that an insect fixes the location of landmarks surrounding a place by storing a sort of snapshot of the landmarks taken from that place; the snapshot taken from the environment is considered as a constellation of objects and it does not encode explicitly the distance between landmarks or between landmarks and target but, instead, the position of each landmark is labelled by its compass bearing [Cartwright, Collet 87].

This use of landmarks is quite different from the classical one, where a landmark has fixed and known position and it is usually labelled with its distance from an important place.

In order to return to specific places, landmarks are used by social insects in one of two different methods: *dead-reckoning* and *near-by landmark* [Snyder 97].

The former is used, for example, by bees when they search a zone for feeding. In this case, information about distance and direction are provided by the dance of the nestmates [von Frisch 71].

The near-by landmark method is used when the dead-reckoning does not have the required degree of precision and in general bees use it for homing or for approaching precisely the feeding area.

An interesting model introduced to explain how bees exploit the near-by landmark phase navigation was proposed by Cartwright and Collett in 1983, subsequently refined in 1987. The results of this model, simulated on a computer, strongly resemble the actual behaviour of bees [Wittmann 95].

In the model, bees seem to learn some information concerning the landmarks surrounding a place through a two dimensional picture (snapshot) memorised from that place together with their orientation.

The steps involved for homing are then:

- *matching phase:* the bee compares the snapshot stored of the place surrounding the goal with the actual snapshot;
- *near-by landmark navigation phase:* the differences in position and in dimensions between the landmarks of the two images drives the bee for positioning.

The matching and the navigation phases have been implemented and tested and they are presented in the following paragraphs.

Interesting aspects concerning the use of distinct landmarks instead of the whole image (snapshot) are considered in paragraph 4.

2.- Matching and navigation

The visual navigation introduced is based on the following constraints: the robot, with its visual system, navigates without changing its vertical position, the images are grabbed always with the same heading and most of the

0-8186-8174-8/97 $10.00 © 1997 IEEE

objects in the navigation environment are considered fixed.

Thus an estimation 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 [Cartwright, Collet 83].

This vector is computed in the following way. An affine simplified model is introduced to describe the translations, dilations and compressions in the whole image. According to a set of model parameters, a new images with changed pixels positions are computed from the actual image. Between the new computed images the one which best match the goal image is chosen. From this match the motion parameters that will be used to calculate the oriented direction for the robot navigation are extracted, as shown in the following figure.

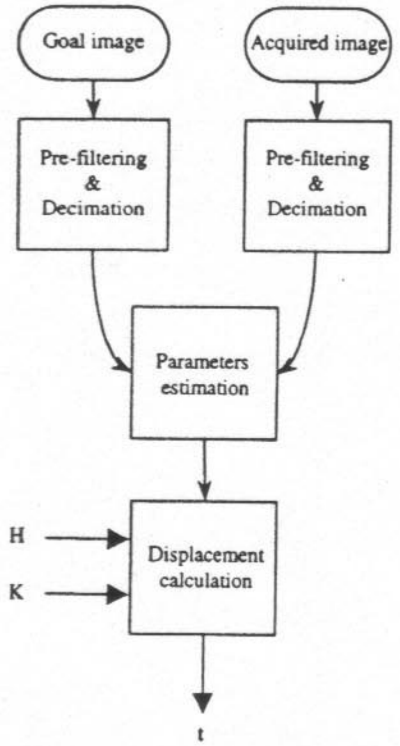

Figure 2.1: Schematic diagram of the algorithm.

2.1- Projection on camera plane

The camera model maps 3D space into image plane using *central projection* [Tse, Baker 91].

Figure 2.2 shows the change of apparent position of a fixed reference objects, after a camera movement described by vector t [Negahdaripour 90].

Relations deriving from Fig. 2.2 are:

$$\begin{cases} X_1 = F\dfrac{x_1}{z_1} \\ Y_1 = F\dfrac{y_1}{z_1} \end{cases} \quad (2.1.1) \qquad \begin{cases} X_1' = F\dfrac{x_1 + t_x}{z_1 + t_z} \\ Y_1' = F\dfrac{y_1}{z_1 + t_z} \end{cases} \quad (2.1.2)$$

with $t = (t_x, 0, t_z)$

where:

- (x_1, y_1, z_1) are the co-ordinates of a point in the 3D space, with respect to the camera reference system
- (X_1, Y_1), are the co-ordinates of the projection of the same point in the image plane
- F is the focal length
- t is the displacement vector.

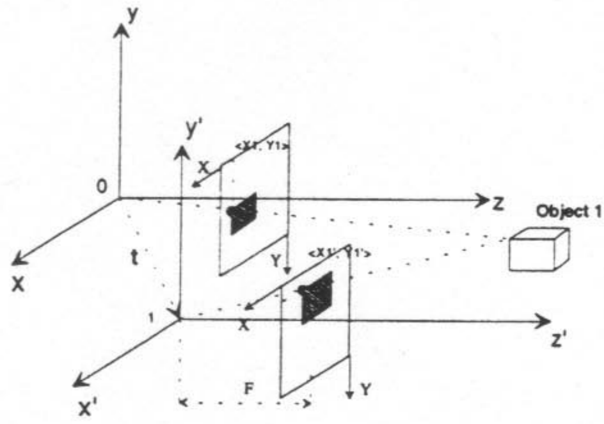

Figure 2.2: Movements on image plane of a fixed point projection after a camera shift

The following is obtained by resolving t_x and t_z in (2.1.1) and (2.1.2):

$$\begin{cases} t_x = \dfrac{1}{F}\left[(X_1' - X_1)z_1 - X_1'\left(\dfrac{Y_1' - Y_1}{Y_1'}\right)z_1\right] \\ t_z = -\dfrac{(Y_1' - Y_1)}{Y_1'} z_1 \end{cases} \quad (2.1.3)$$

2.2- The affine model

A model that takes into account translations, compressions and rotations of an object projection in the image plane, caused by a change in camera-object relative positioning, is described by the following equations [Wu, Kittler 90] :

$$\begin{cases} S_X(X,Y) = a_{0x} + a_{1x}\cdot X + a_{2x}\cdot Y \\ S_Y(X,Y) = a_{0y} + a_{1y}\cdot X + a_{2y}\cdot Y \end{cases} \quad (2.2.1)$$

where:

- S_X and S_Y are the displacement components in image plane
- *(X, Y)* are the pixel co-ordinates, a_{0x} and a_{0y} are the translation parameters
- a_{1x} and a_{2y} are the compressions parameters
- a_{1y} and a_{2x} are the rotations parameters.

Starting from the initial constraints some simplifications can be made in (2.2.1). The fixed objects constraint allows not to take into account object rotation and moreover the vertical height constraint implies that a_{0y} is null. Thus the apparent vertical shift on the image plane is due to objects compression or dilation and is taken into account by a_{2y}.

So, the simplified (2.2.1) is:

$$\begin{cases} S_X(X,Y) = X_1 - X_1' = a_{0x} + a_{1x}X_1' \\ S_Y(X,Y) = Y_1 - Y_1' = ha_{1x}Y_1' \end{cases} \qquad (2.2.2)$$

with a_{0x} in pixels, representing translations, a_{1x} and $a_{2y}=ha_{1x}$, a-dimensional, representing expansions.

Forcing $a_{2y}=ha_{1x}$ implies that objects apparent dimension changes are related only with z traslation of the camera system. In fact, in the above model, perspective distortions of objects in the image are not taken into account, but the following matching phase allows the system to choose the best motion parameters approximation from the new point of view.

2.3- The matching algorithm

The implemented algorithm finds the a_{0x}, a_{1x}, parameters values that minimise the following Mean Square Error (MSE) on the whole image [Netravali, Haskell 88]:

$$MSE = \frac{1}{M} \sum_{<x,y>\in S} E_r(X,Y) + E_g(X,Y) + E_b(X,Y) \qquad (2.3.1)$$

with $E_{r,g,b}(X,Y) = [I1_{r,g,b}(X,Y) - I2_{r,g,b}(X+S_X, Y+S_Y)]^2$

where:

- M is the number of couples ($X+S_X, Y+S_Y$) still in the image plane S
- $I1$ is the goal image and $I2$ is the actual transformed image
- r, g and b are the chromatic components
- S_X and S_Y are the estimate displacement vectors for every pixel.

In order to speed up parameters estimation and at the same time to allow the estimation of large displacement vectors, a multi resolution pyramidal technique has been implemented.

According to this technique subsampled images are used: image at level i is obtained by subsampling by a factor two the image at level i-1. The displacement estimation starts from the images at a lower resolution going up the pyramid by maintaining unchanged the dimensions of the estimation intervals.

The parameters estimated at level i are subsequently used at the level i-1 like offset for the relative estimation ranges.

Before each subsampling operation a low pass seven coefficients gaussian filtering is applied in order to exclude possible spatial aliasing [Pratt 91].

The estimation ranges and the multi resolution levels are chosen according to the maximum displacement considered.

Moreover, due to the null value of the vertical shift vertically decimated images of a factor 2^k. have been used (Fig. 3.1).

2.4- Estimation of the navigation vector

From (2.1.3) and (2.2.2) derives:

$$\begin{cases} t_x \cong -\frac{z_1}{F} a_{0x} \\ t_z \cong z_1 a_{1x} \end{cases} \qquad (2.4.1)$$

with $h=1$, which links the simplified affine model parameters, deriving from the projection changes of an object point, with the relative camera translation.

This projection changes is related to the focal length F of the system and to the relative distance z_l of the object point.

Thus, the robot displacement, resulting from the use of the (2.3.2) on the whole image, after the best matching phase, can be calculated as:

$$\begin{aligned} \Delta x &= K \cdot a_{0x} \\ \Delta z &= H \cdot a_{1x} \end{aligned} \qquad (2.4.2)$$

where K and H derives from the average of all the objects depth in the entire image.

The meaning of K and H in the real robot navigation will be discussed in Par. 3.

2.5- Robust matching

Noise in the acquired images is mainly due to the following factors:

- people moving in the room
- changes of displays on computer monitors
- lighting being switched on and off

In order to reduce the effect of such noise, the best matching algorithm is computed minimising a different MSE.

This different MSE is calculated in two steps: the first step calculates the MSE on all the pixels as in (2.3.1), then only on the pixels with a punctual square error (2.5.1) lower than the previously computed global MSE are used to recalculate it. The punctual pixel error is computed using the following equation:

$$E_{r,g,b}(X,Y) = [I1_{r,g,b}(X,Y) - I2_{r,g,b}(X+S_X, Y+S_Y)]^2 \qquad (2.5.1)$$

3.- Tests and discussions

The navigation tests have been conducted by taking the images from the real scene and by computing off line the navigation vectors on a PC.

An example of real image used by the algorithm is shown in Figure 3.1. Image a is acquired in the goal position. Image b is obtained after a low pass filtering and a vertical decimation of image a. Image c is obtained after low pass filtering and a vertical decimation of an image acquired in a generic point of the navigation area.

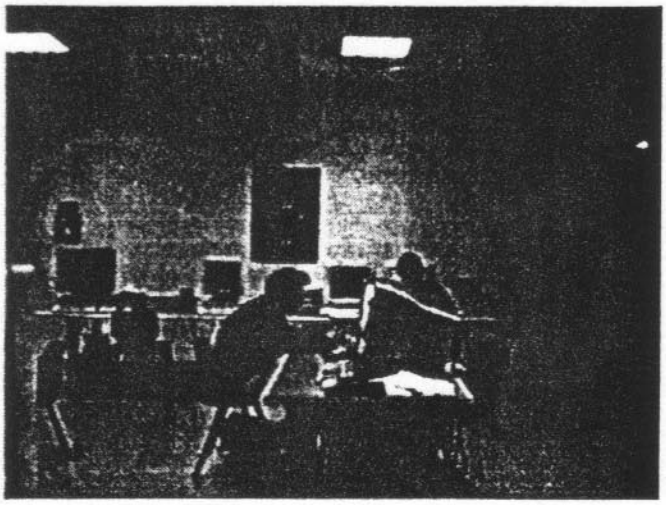

Figure 3.1 a): examples of goal position image

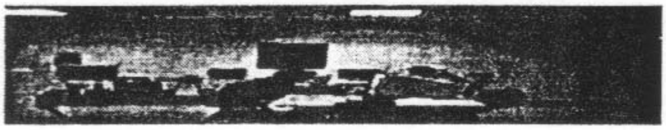

Figure 3.1 b): vertically subsampled goal position

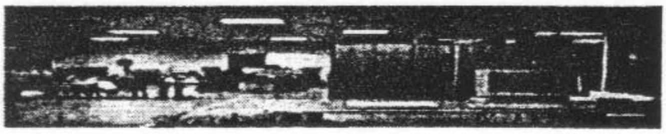

Figure 3.1 c): vertically subsampled starting point image

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

In all tests, the values of parameters H and K, after the model calibration, are: H = -1,8 [cm/pixel] and K = 680 [cm]. Anyhow the estimation of the parameters H and K is not a necessary step for the implemented navigation system. Their values influence the module of the navigation vector, not its direction. 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 parameter increment: $\Delta a_{0x} = \pm 3$ $[pixel]$ with increment $da_{0x} = 1$ $[pixel]$ and $\Delta a_{1x} = \pm 0.5$ with increment $da_{1x} = 0.05$

With these values the system resolution is ± 32.38 centimetres on Z axis and ± 1.71 centimetres on X axis. The image obtained by the camera is 384x288 pixels. The vertical under sampling is 4:1. The pyramidal structure used has five levels. The camera has been always placed at a height of about 1.2 m.

The results of a complete navigation process from each given starting point can be seen in Figure 3.2.

Navigation errors are reported in Table 1: after two navigation steps they are below the maximum admissible error in centimetres for the selected parameter increment.

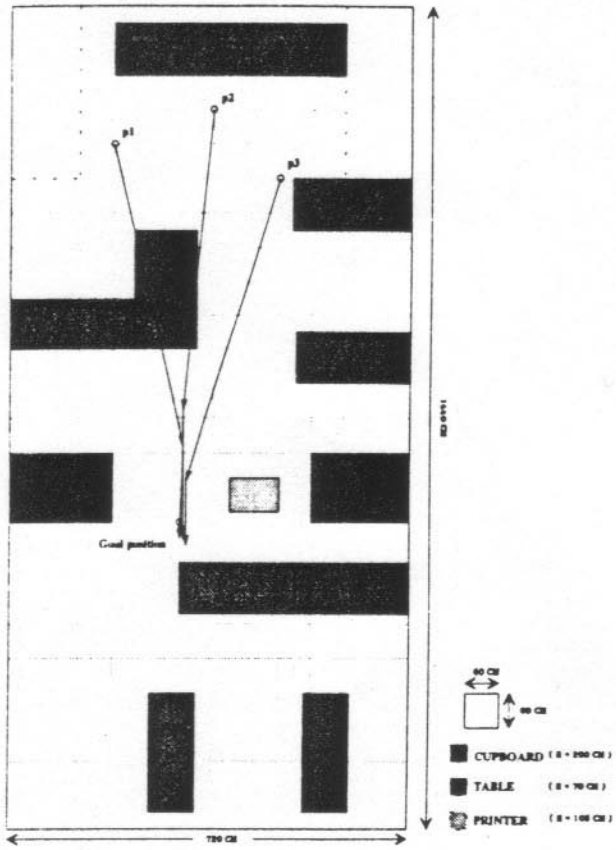

Figure 3.2: Test navigation results

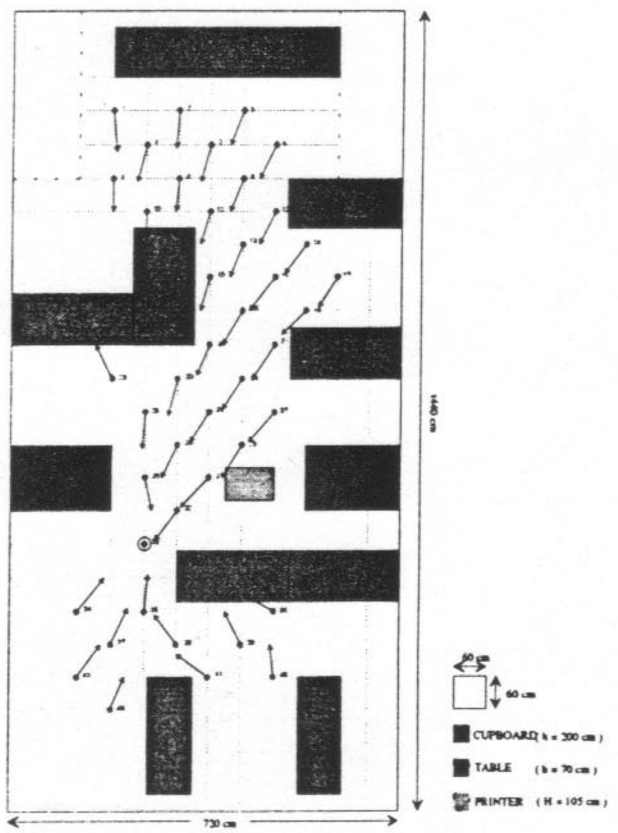

Figure 3.3: Directions of every estimate displacement vector (vector lengths are not showed)

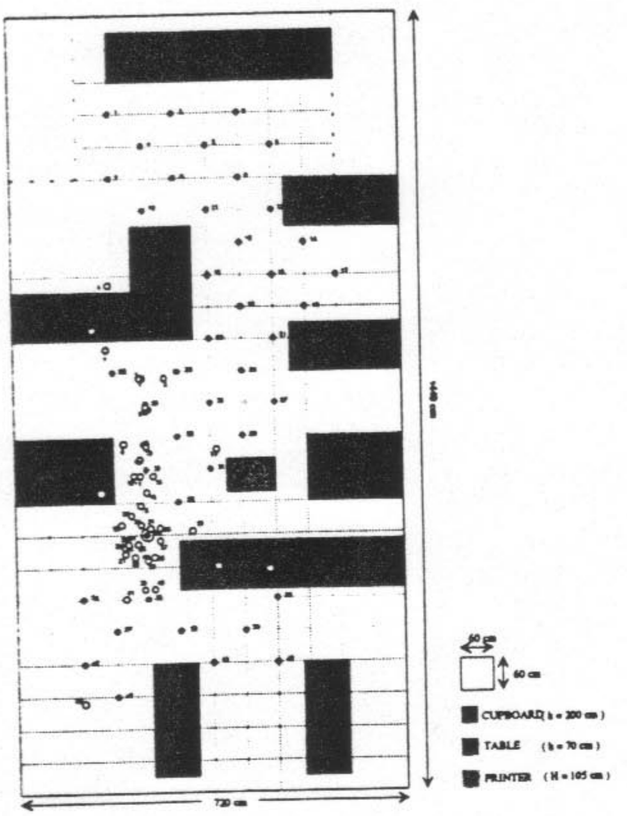

Figure 3.4: Target points of every estimate displacement vector

walk	steps	error (cm)
P1	2	1.8
P2	2	8.1
P3	2	10.9

Table 1: Test error table

In Figure 3.3 the direction of the estimated displacement vectors for the first navigation steps from some starting points are shown.

In Figure 3.4 the estimated target point after the first navigation steps starting from the same points of Fig. 3.3 are shown.

In the first group of tests (Fig. 3.2), the navigation phase is completed after 2 steps, with a mean error of about 5 cm along a path of about 720 cm: less than 0.7 %.

In the second group of tests the initial directions for each starting point are considered (Fig. 3.3 and 3.4). Almost all points show good navigation behaviours except point 22: a cupboard occludes a consistent part of the image grabbed from that point so the matching phase fails. A wider image could overcome this situation.

4 .- Conclusions and perspectives

The homing method described in this paper has shown good results in conducting the robot toward a goal position, but several improvements are still possible. The simplified model that was 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.

Anyhow the main criticism is the computational load that is still heavy.

Different visual navigation algorithms should be investigated in order to accomplish a real time navigation.

Some interesting answers could be found in the *Turn Back and Look* (TBL) phase fulfilled by bees [Lehrer, Collett 94]. In particular, it is asserted that a distance estimation of the landmarks is learned during the first departures from a place through this phase and the color and the apparent dimension of the landmark are learned subsequently.

In this phase bees learn the position, the shape and the colour of useful navigation landmarks by exploiting specific "learning flights". In this way, an initial learning phase could reduce the portion of scene needed to compute the navigation vectors and consequently reduce the computational load.

The performance improvement achieved by the use of an higher level navigation module that activates alternatively each different visual navigation (snapshot and landmark) could also be explored.

5.- Bibliography

[Cartwright, Collett 83]] B. A. CARTWRIGHT, T. S. COLLETT, *"Landmark Learning in Bees"*. J. of Comp. Physiol. A 151, 521-543, 1983.

[Cartwright, Collett 87] B. A. CARTWRIGHT, T. S. COLLETT, *"Landmark Maps for Honeybees"*. Biol. Cybern. 57, 85-93, 1987.

[Cheng et al. 87] K. CHENG, T. S. COLLETT, A. PICKHARD, R. WEHNER, *"The use of visual landmarks by honeybees: Bees weight landmarks according to their distance from the goal* J. of Comp. Physiol. A 161, 469-475, 1987.

[Gould 86] J. L. GOULD, *"Landmark learning by honey bees"*. Anim. Behav. 35, 26-34, 1986.

[Lehrer, Collett 94] M. LEHRER, T. S. COLLETT, *"Approaching and departing bees learn different cues to the distance of a landmark"*. J. of Comp. Physiol. A 175, 171-177, 1994.

[Negahdaripour 90] S. NEGAHDARIPOUR, *"Multiple interpretation of the shape and Motion of Objects from Two Perspective images"*. IEEE Trans. on P.A.M.I., vol. 12, no. 11, pp. 1025-1039, 1990.

[Netravali, Haskell 88] A.N. NETRAVALI, B.G. HASKELL, *"Digital Pictures-Representation and Compression"*, Plenum Press, New York, 1988.

[Pratt 91] W.K. PRATT, *"Digital Image Processing"*. A Wiley interscience publication, Second Edition, pp.248, New York, 1991.

[Santos-Victor et al. 93] J. SANTOS-VICTOR, G. SANDINI, F. CUROTTO, S. GARIBALDI, *"Divergent stereo for robot navigation: learning from bees"*. IEEE Computer Society Conference on Computer Vision e Pattern Recognition, New York City, June 15-18, 1993.

[Santos-Victor et al. 94] J. SANTOS-VICTOR, G. SANDINI, F. CUROTTO, S. GARIBALDI, *"Divergent stereo in autonomous navigation: from bees to robots"*. Int. Jour. of Computer Vision, 14, 159-177, Kluwer Academic Publishers, Boston, 1995.

[Srinivasan et al. 89] M. V. SRINIVASAN, M. LEHRER, S. W. ZHANG, AND G. A. HORRIDGE, *"How honeybees measure their distance from objects of unknown size"*. J. of Comp. Physiol. A 165, 605-613, 1989.

[Snyder 97] M. SNYDER, *Personal Communication*, University of Alberta, Canada, January 1997.

[Tse, Baker 91] Y.T. TSE, R.L. BAKER, *"Camera zoom/pan estimation and compensation for video compression"*. SPIE Image Processing Algorithms and Techniques II, vol. 1452, pp. 468-479, 1991.

[von Frisch 71] K. VON FRISCH, *"Bees. Their Vision, Chemical Senses and Language"*. Cornell University Press, Ithaca (NY), 1971.

[Wehner 92] R. WEHNER, *"Arthropods"*. In Animal Homing, ed. by F. Papi, London, Chapman and Hall, 45-144, 1992.

[Wittman 95] T. WITTMAN, *"Insect navigation: Models and Simulations"*. Tech. rept. 02/95. Zentrum fur Kognitionswissenschaften, University of Bremen, 1995.

[Wu, Kittler 90] S.F.WU, J. KITTER, *"A Differential Method for Simultaneous Estimation of Rotation, Change of Scale and Translation"*. Sigal Processing: Image Comunication, 2 (1990), pp.69-80.