Towards Optical Flow-based Robotic Homing

Sotirios Ch. Diamantas, Anastasios Oikonomidis, and Richard M. Crowder

Abstract— **This paper presents a novel biologically-inspired approach for tackling the problem of robot homing. In our method the only information employed is optical flow. Optical flow, which is not a property of landmarks like colour, shape, and size but a property of the camera motion, is used for localising an autonomous robot in *a priori* unknown environment. Our method exploits the optical flow 'fingerprint' of landmarks caused by the motion of the robot in the environment. For this purpose, we have developed a training algorithm that estimates the probability of observing the same landmark from varying distances and velocities. Our method promises to be computationally efficient and inexpensive. The simulation results we present show the validity of our methods.**

I. Introduction

Visual navigation lies at the heart of mobile robotics. Homing (or inbound journey) refers to the navigation process where an autonomous agent performs a return to its home position after having completed foraging (or outbound journey; foraging is mainly attributed to a biological agent). A robot may have to return to its base for a number of reasons like recharging batteries, failure of a subsystem, or completion of a task. The application areas of robots capable of performing homing are plenty and vary. Search and rescue robots are in need in areas that have been hit by earthquakes or in environments that are hazardous for humans [1]. Planetary missions to other regions constitute another application area of robots whose navigation process involves returning back to their base. In this paper we have developed a novel approach to tackle the problem of robot homing using visual modality as the only source of information. No other sensor is provided to the robotic agent apart from two side-ways cameras mounted on a simulated mobile platform.

Optical flow, that is the rate of change of image motion in the retina or a visual sensor, is extracted from the motion of the autonomous agent. The orientation of the cameras on the robotic platform are perpendicular to the direction of motion so as a translational optic flow information is generated. Optic flow, which is not a property of the landmarks, like colour, shape, and size, but a property of the camera motion has been used for building topological maps in *a priori* unknown environment based on the optical flow patterns of the landmarks. The novelty of our method lies in the fact that no information is given such as the position or the velocity of the robot but only the optical flow 'fingerprint' of the landmarks caused by the motion of the robot. For this purpose, a training algorithm has been deployed and a probability is inferred that is computed from the similarity of the optical flow patterns between the outbound and inbound journeys.

Biology is seen as an alternative solution to the problems robots encounter which includes algorithmic complexity, performance, and power consumption among others. Biological inspiration provides simple, yet effective methods for the solutions of such problems. The careful examination of those methods has twofold gain. The study of the principles of biological organisms entails making better autonomous systems that will, in turn, help us perceive and understand better the underlying mechanisms that underpin the biological organisms.

This paper comprises five sections. Following is Section II where related as well as background work is presented. In Section III the methodology of the homing model is described. Section IV presents the results of the statistical model on the homing process of navigation. Finally, Section V epitomizes the conclusions drawn from this work and indicates a number of areas that further research is attainable.

II. Related Work

A large number of insects use optic flow for navigation. Insects like *Drosophila* use the apparent visual motion of objects to supply information about the three-dimensional structure of the environment. The fly *Drosophila* uses optic flow to pick near targets. Collett in [2] shows that in insects the task of evaluating distances between objects is made easier by making side-to-side movements of the head strictly translational and disregarding any rotational components that can influence the distance to the objects. Looming, i.e., image expansion, can also distort the actual distance to the object as the apparent size compared to the physical size of the object differs. Collett in his experiments [2] ascertains that *Drosophila* like many insects limit rotational flow during exploratory locomotion. In fact, *Drosophila* move in straight-line segments and restrict any rotation to saccades at the end of each segment. Schuster et al. [3] have used virtual reality techniques to show that fruit flies use translational motion for picking up the nearest object while disregarding looming.

Ladybirds also move in straight-line segments and rely on translational optic flow rather that looming cues. Other animals like locusts and mantids turn their head from one side to the other just before jumping. Kral and Poteser [4] suggest that locusts and mantids use translational motion to infer the three-dimensional structure of the environment and in particular the distance to the object they wish to approach. In some other experiments performed by Tautz et al. [5] trained bees had to travel large distances across various

Sotirios Ch. Diamantas and Richard M. Crowder are with the Intelligence, Agents, Multimedia (IAM) Group, School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom. Email: {sd04r, rmc}@ecs.soton.ac.uk.

Anastasios Oikonomidis is with the Centre for Risk Research, School of Management, University of Southampton, Southampton, SO17 1BJ, United Kingdom. Email: ao1@soton.ac.uk.

scenes that included both land and water. The results showed that the flights over water had a significantly flatter slope than the ones above land. This suggests that the perception of distance covered by bees is not absolute but scene-dependent where the optic flow perceived is evidently larger. This may also suggest why some bees are drowning by 'diving' into lakes or the sea while flying above water. The distance and direction to a food source is communicated in the bees by means of waggle dances that integrate retinal image flow along the flight path [6], [7].

Two well-known homing models are the *snapshot* and the *Average Landmark Vector* (ALV) model. The *snapshot* model is an implementation of the template hypothesis [8], [9]. It requires a panoramic snapshot of the goal position, be it a hive, nest, or a food source. Along with the snapshot the compass direction is stored. The *snapshot* model is an image matching process between a snapshot taken at a goal position and a snapshot containing the current view. The image obtained from the omnidirectional camera is unwrapped and a threshold operation is performed to yield a one-dimensional black and white image. The landmarks are denoted as black marks on the image. Then, this is compared with the snapshot of the current view to produce the homing vector. The *homing vector* is a two-dimensional vector pointing towards the home position and is obtained by summing up all radial and tangential vector components. The ALV model [10] uses, too, a processed panoramic image but, in contrast to the *snapshot* model, it need not be stored. Only a two-dimensional vector for each landmark needs to be stored that points to the direction of the landmark. Matching and unwrapping of the image are not required since the calculations are performed on the basis of vector components. Thus, ALV is more parsimonious than the *snapshot* model. Nevertheless, snapshots in the ALV model have to be captured and processed to produce a one-dimensional picture, as is in the snapshot model. A compass information is required for the ALV model as well.

The *snapshot* and the *Average Landmark Vector* are two models that have been inspired by the way insects perform homing. On one hand, their main advantage is the simplicity of the method that entails a low computational complexity. On the other hand, their disadvantages are that both methods are applied at the end of the homing process, that is, when an agent is close to its home position. Moreover, all landmarks need to be visible both in the current snapshot as well as in the stored snapshot, that is, they must be the same landmarks in both snapshots. In addition to the biologically inspired methods, other methods of homing have been developed that make use of the Extended Kalman Filter (EKF) or homing methods that are based on panoramic vision [11]. The former are trying to tackle the Simultaneous Localisation and Mapping (SLAM) problem and make use of probability to build accurate maps that are based on vehicle position estimates. In turn, these maps provide a more accurate estimate of the vehicle's position.

A work by Newman et al. [12] addresses the problem of homing through the localisation and mapping method using a laser scanner and wheel encoders. In their method they extract line segments from a laser scanner that act as features while the position of the robot is calculated from wheel encoder readings. A major issue of the SLAM methodology lies in the accumulation of errors as the robot navigates. The slippage of the terrain causes erroneous sensor readings and, the harsher a terrain is, the larger the accumulation of errors that occur. The main disadvantage of SLAM, however, is the computational cost that increases quadratically as new observations are made by the sensor be it a laser scanner or a camera [13], [14]. Updating all features and the joint covariance matrix each time a new observation is made adds significantly to the computational cost of the method. Another issue in the SLAM methodology is the *data association* problem where features or landmarks look alike. FastSLAM uses Rao-Blackwellised particle filters and is a method that tries to alleviate the *data association* problem [15], [16]. SLAM methods can be applied to a wide range of environments, such as indoors, outdoors, dynamic, or large-scale environments. It can, thus, be understood why SLAM has risen to one of the most research-intensive problems in the robotics field.

A. Applications of Optical Flow

Lately a growing number of autonomous vehicles have been built using techniques inspired by insects and, in particular, optical flow. One of the first works that studied the relation of scene geometry and the motion of the observer was by Gibson [17]. A large amount of work, however, has been focussed on obstacle avoidance using optical flow [18], [19], [20]. The technique, generally, works by splitting the image (for single camera systems) into left- and right-hand side. If the summation of vectors of either side exceeds a given threshold then the vehicle is about to collide with an object. Similarly, this method has been used for centring autonomous robots in corridors or even a canyon [21] with the difference that the summation of vectors this time must be equal in both the left-hand side and the right-hand side of the image. Ohnishi and Imiya [22] utilise optical flow for both obstacle avoidance and corridor navigation. The performance of optical flow has also been tested in underwater colour images by Madjidi and Negahdaripour [23]. Vardy [24] deploys various optical flow techniques which are compared using block matching and differential methods to tackle homing.

In a recent work implemented by Kendoul et al. [25] optic flow is used for a fully autonomous flight control of an Unmanned Aerial Vehicle (UAV). The distance travelled in this UAV is calculated by integrating the optical flow over time. A similar work for controlling a small UAV in confined and cluttered environments has also been implemented by Zufferey et al. [26]. Barron et al. [27] discuss the performance of optical flow techniques. Their comparison is focussed on accuracy, reliability and density of the velocity measurements. Other works employ optic flow methods for depth perception [28], motion segmentation [29], or estimation of ego-motion [30].

A similar technique to optical flow developed by Langer and Mann [31] called *optical snow* arises in situations where camera motion occurs in highly cluttered 3D environments. Such cases involve a passive observer watching the fall of the snow, hence, the name of the method. *Optical snow* has been inspired by research in animals that inhabit in highly dense and cluttered environments; such animals include the rabbit, the cat, and the bird. The properties of the *optical snow* are that yields dense motion parallax with many depth discontinuities occurring in almost all image points. This comes in contrast to the classical methods that compute optical flow and presuppose temporal persistence and spatial coherence. In the work of Langer and Mann [31] the properties of *optical snow* in the Fourier domain are presented and investigate its computational problems on motion processing.

B. Mathematical Foundations of Optical Flow

This section describes the mathematics that underlie the optic flow algorithms, and in particular, the Lucas-Kanade (LK) algorithm [32] that has been employed in this research work. In order for the optic flow algorithms to perform well, some suitable images need to be chosen. This suitability refers to images that have high texture and contain a multitude of corners. Such images have strong derivatives and, when two orthogonal derivatives are observed then this feature may be unique, and thus, good for tracking. Tracking a feature refers to the ability of finding a feature of interest from one frame to a subsequent one. Tracking the motion of an object can give the flow of the motion of the objects among different frames. In Lucas-Kanade, algorithm corners are more suitable than edges for tracking as they contain more information. For the implementation of the LK algorithm the *OpenCV* library [33] has been used.

The optic flow algorithm of Lucas-Kanade presupposes three main criteria to produce satisfactorily results. These are:

1) Brightness constancy. The brightness of a pixel does not change from frame to frame, that is $I(x, y, t) = I(x + u, y + v, t + 1)$.
2) Temporal persistence or small movements. The motion of the object that is tracked moves smoothly from frame to frame, that is $I_x u + I_y v + I_t = 0$, where v, u are the x, y components of the velocity $\vec{u}$.
3) Spatial coherence. Neighbouring points of a pixel that belong to the same surface have typically similar motion, and project to nearby points on the image plane [34].

The equation in the second criterion is an under constrained equation since it involves two unknowns for any given pixel and cannot be used to solve the motion of a pixel in the two dimensions. For this reason the third criterion is used as an assumption to solve the full motion of a pixel in the two dimensions. The third criterion assumes that the neighbouring pixels of any given pixel move coherently as they belong to the same object and project to the same image plane as the given pixel projects. Thus, for solving the problem in case, the brightness values of the neighbouring pixels are taken and solve a system of linear equations [34]. Hence, if we take a window of 5×5 pixels a system of 25 linear equations needs to be solved. However, if a window is too small the *aperture problem* may be encountered where only one dimension of the motion of a pixel can be detected and not the two-dimensional. On the other hand, if a window is too large then the spatial coherence criterion may not be met. Nevertheless, the system that needs to be solved following a window of 5×5 pixels is expressed by (1)

$$\underbrace{\begin{bmatrix} I_x(p1) & I_y(p1) \\ I_x(p2) & I_y(p2) \\ \vdots & \\ I_x(p25) & I_y(p25) \end{bmatrix}}_{A = 25 \times 2} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\vec{u} = 2 \times 1} = - \underbrace{\begin{bmatrix} I_t(p1) \\ I_t(p2) \\ \vdots \\ I_t(p25) \end{bmatrix}}_{b = 2 \times 1} . \quad (1)$$

The goal on the above system of linear equations is to minimise $||A\vec{u} - b||^2$ where $A\vec{u} = b$ is solved by employing least-squares minimisation as in (2),

$$(A^T A)\vec{u} = A^T b \quad (2)$$

where $A^T A, \vec{u}$, and $A^T b$ are equal to (3),

$$\underbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}_{A^T A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\vec{u}} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b} \quad (3)$$

and the solution to the equation is given by (4)

$$\vec{u} = \begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b. \quad (4)$$

If $A^T A$ is invertible, i.e., no zero eigenvalues, it means it has full rank 2 and two large eigenvectors. This occurs in images where there is high texture in at least two directions. If the area that is tracked is an edge, then $A^T A$ becomes singular, that is (5),

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} -I_y \\ I_x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5)$$

where $-I_y, I_x$ is an eigenvector with eigenvalue 0. If the area of interest is homogeneous then $A^T A \approx 0$ implying 0 eigenvalues. The reason that the LK algorithm was chosen is that it is a fast and accurate optical flow algorithm. It relies on local information rather than global as is the nature of Horn and Schunck algorithm [35]. This has the advantage of performing fast optic flow operations. Nevertheless, the disadvantage of local information, that is derived from a small window, is that large motions can move out of the local window. The pyramidal approach of the LK algorithm uses a coarse-to-fine iterative method, that is, various layers in scale-space to overcome the local information problem. Thus, the optical flow problem is first solved at the top layer by tracking over large spatial scales and then as it proceeds downwards to the lower layers the initial velocity criteria are refined until it arrives at the raw image pixels.

III. Methodology

In this section we describe how the optical flow 'signature' of the landmarks, that is caused by the perceived motion of the robot in the environment, can be used to localise the robot during the homing process. Various landmarks have been modelled and simulated from which the robot passes through. The simulated landmarks have geometrical shapes like a sphere or a rectangular and they are textured in order to produce large amounts of optic flow (as is in real environments). As mentioned in Section I the simulated robot consists of two side-ways cameras which are perpendicular to the direction of motion. This creates a translational optic flow as the robot navigates through the environment. Every landmark in the environment 'emits' a number of optic flow vectors that are dependent on the distance between the robot and the landmark, and the velocity of the robot. One of the advantages of our method is that images are only captured and are not used for storage or comparison. Storing and comparing only the properties of vectors between different frames, that is, the mean position of all the vectors and the number of vectors, reduces the computational complexity and the cost of the homing process.

During the outbound trip of the robot the camera calculates and stores the optic flow vectors that are generated by the motion of the vehicle. During this phase the robot builds a topological map from the optical flow 'fingerprint' of the landmarks. After the foraging trip has completed the homing trip is initiated. In the homing phase, the robot compares the optical flow patterns it currently perceives with the ones occurred during the foraging journey. If the similarity score (i.e., probability) between the two patterns is above a given threshold, then the robot assumes the current landmark observed is the same with the landmark observed during the outbound trip. This information is then used to localise the robot within the topological map. The similarity score of the vectors is a probabilistic result of the Euclidean distance of the vectors between the current image and the image taken during the outbound journey.

In order for the robot to localise in an environment using optic flow vectors, a training data set of $n = 1000$ observations has been implemented where a vector is observed at varying distances between the robot and the landmark, and at varying velocities. The distances and velocities chosen to create the training set approximate the real distributions of velocity and distance when a robot navigates in an environment. Thus, a joint probability distribution has been created by two continuous and independent variables, that is velocity, C, and distance, D, and is expressed by (6)

$$f_{C,D}(c,d) = f_C(c) \cdot f_D(d) \qquad \forall c, d. \tag{6}$$

The velocity and the distance variables have been drawn from two Gaussian distributions with $\mu = 4, \sigma = 1$ and $\mu = 11, \sigma = 3$, respectively. The n observations model the position of the vector in the plane in a varying combination of distances and velocities. One assumption that needs to be met in our method is that the majority of the vectors comprising a given landmark should have the same, or almost the same magnitude. In order to solve the similarity problem between vectors, the mean, x_k, y_k, or centre point of every vector is taken. Thus, summing up all the mean points of the training set and dividing by the number of observations we end up having the mean of the means, $\bar{x}, \bar{y}$, as shown in (7)

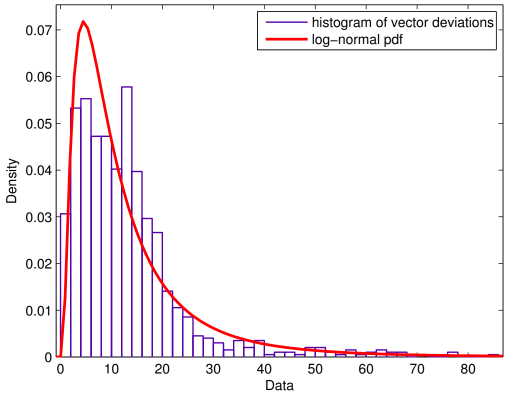


Fig. 1. Histogram of vector deviations of the training algorithm and the log-normal probability density function (pdf) fit. Mean and standard deviation are $\mu = 2.24$ and $\sigma = 0.86$, respectively.

$$\bar{x}, \bar{y} = \frac{1}{n} \cdot \sum_{k=1}^{n} x_k, y_k \qquad n = 1000. \tag{7}$$

The mean of the means in an optic flow pattern can be visualised as the centre of gravity in a physical system. We then compute the Euclidean distances, χ_k, between the mean of the means and the n observations as expressed by (8)

$$\chi_k = \sqrt{(x_k - \bar{x})^2 + (y_k - \bar{y})^2}. \tag{8}$$

The histogram produced by the Euclidean distances, χ_k, forms a log-normal probability density function (pdf) with $\mu = 2.24$ in log location and $\sigma = 0.86$ in log scale. Figure 1 shows the histogram of vector deviations and the probability density function of the log-normal. The log-normal pdf is deployed in order to infer a probability as to how likely it is for the vectors of the current snapshot to have deviated when compared with the vectors of a snapshot stored in memory. Figure 2 depicts the cumulative distribution function (cdf) of vector deviations and the log-normal.

The cumulative density function of log-normal is expressed by (9), where *erfc* is the complementary error function and Φ is the standard normal cdf. The probability density function of log-normal is given by (10)

$$F_X(\delta; \mu, \sigma) = \frac{1}{2} erfc \left[-\frac{\ln \delta - \mu}{\sigma\sqrt{2}} \right] = \Phi \left(\frac{\ln \delta - \mu}{\sigma} \right) \tag{9}$$

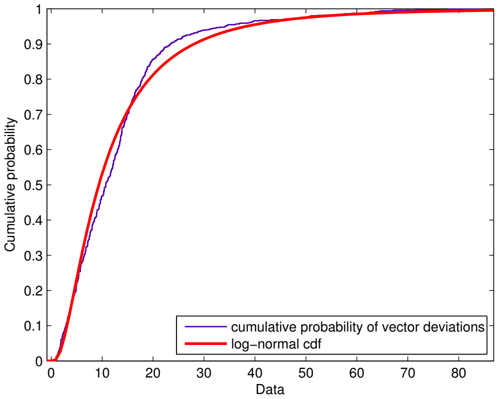


Fig. 2. Cumulative density functions (cdf) of vector deviations and the log-normal distribution.

$$f_X = (\delta; \mu, \sigma) = \frac{1}{\delta\sigma\sqrt{2\pi}} e^{-\frac{(\ln\delta - \mu)^2}{2\sigma^2}} \qquad \delta > 0. \tag{10}$$

Thus far, we have explained the methodology of the training algorithm. We now move on to the process of calculating a probability for the patterns observed by the robot during the foraging and homing process. This probability will aid the robot localise itself in the environment. During the homing navigation process, the robot calculates the Euclidean distance, δ, between the mean position, $\bar{x}, \bar{y}$, of all the vectors in a given landmark with the mean position of the vectors of the landmarks stored in the database. Equations (11), (12), and (13) describe the process for two distinct landmarks. In (11), (12), r and s are the number of vectors for two distinct landmarks i and j, one of which is observed during the outbound trip while the other one is observed during the inbound trip.

$$\bar{x_i}, \bar{y_i} = \frac{1}{r} \cdot \sum_{a=1}^{r} x_a, y_a \tag{11}$$

$$\bar{x_j}, \bar{y_j} = \frac{1}{s} \cdot \sum_{b=1}^{s} x_b, y_b \tag{12}$$

$$\delta = \sqrt{(\bar{x_i} - \bar{x_j})^2 + (\bar{y_i} - \bar{y_j})^2} \tag{13}$$

$$P = 1 - P_\delta \tag{14}$$

The log-normal cdf then gives us the probability P_δ based on the Euclidean distance δ between the two sets of vectors. It is then subtracted from 1 to give the probability P as is in (14). In addition, the probability P of the log-normal is multiplied by the ratio of the number of the vectors as shown in (15) with min_i being the landmark i with the minimum number of vectors and max_j being the landmark j with the maximum number of vectors.

$$P_T = P\left(\frac{min_i}{max_j}\right) \tag{15}$$

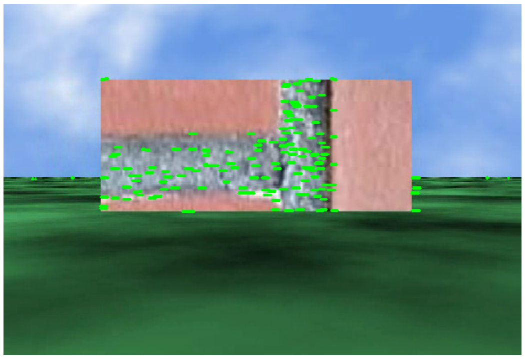


Fig. 3. Snapshot of the reference landmark and its optical flow ‘signature’ taken at a distance of $11m$ and a velocity of $4km/h$.

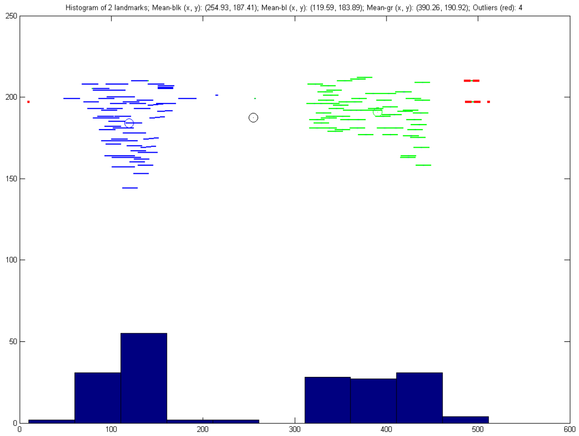


Fig. 4. Clustering of two landmarks by employing the histogram of their vectors.

This results to the total probability (or similarity score), P_T. Thus, even if the Euclidean distance, δ, between two sets of vectors is small, the total probability, P_T, can be low if the ratio of the vectors is small. Hence, two patterns which are totally different may have a small Euclidean distance that yields a high probability. Multiplying the probability value, P, by the ratio of the number of vectors can drop significantly the total probability value, P_T, assuming that the number of vectors of the two sets are not of the same multitude. The landmark of Fig. 3 acts as a reference for the following snapshots in order to demonstrate the similarity score at varying distances and velocities, and between different landmarks. The optic flow images are created by calculating the motion of a landmark between two contiguous frames. It should also be noted that the flow vectors appear upside down since the images are read from top to bottom.

In this work, a clustering algorithm has also been implemented for counting the number of landmarks. The counting of landmarks is a method that is used by insects as revealed in a recent work by Dacke and Srinivasan [36]. Although the clustering algorithm is a rather simple one, distinguishing between different landmarks is an important task. The number of landmarks each one of the two cameras can distinguish is two. Therefore four is the maximum number of landmarks that can be seen at any time. However, in the case where one or more landmarks have gone missing then the robot can still recognise its location from the optical flow 'fingerprint' of the remaining ones. The clustering algorithm creates a histogram of the vectors and finds the minimum point between the two peaks of the histogram. Figure 4 shows an application of the algorithm with two landmarks. In the same figure some outliers have been located, that is, vectors with small length that have been disregarded.

IV. Results

The homing model described in this paper has been implemented in C++ programming language and the MATLAB [37] software has been used for the analysis of the data. The *breve* simulator [38] was used for the creation of landmarks in 3D. The algorithm was run offline on a Pentium 4 machine at 3.00 GHz with 1.00 GB of RAM. The following graphs, Fig. 5, demonstrate the effectiveness of our approach by comparing the vectors of the reference image, Fig. 3, taken at a distance of $11m$ and a velocity of $4km/h$ with the vectors of the same landmark taken at different distances and velocities. Figures 5(a), 5(b), and 5(c) depict the similarity score at a distance of $11m$ and a velocity of $5km/h$ while Figs. 5(d), 5(e), and 5(f) depict the similarity score at a distance of $8m$ and a velocity of $4km/h$.

The circle in the graphs represents the mean position of all the vectors that comprise a landmark. The *green*, (Gr), optic flow vectors refer to the reference image while the *blue*, (Bl), ones refer to the current snapshot. Deviation is the Euclidean distance, δ, between the mean position of the vectors of the current snapshot, $\bar{x}, \bar{y}$, with the mean position of the vectors of the reference image. The number of elements, i.e., vectors, in the current snapshot differs from frame to frame as the angle of perception changes. Time, t, denotes the time steps the images were captured. It is clear that the similarity score is quite high in all three images, Figs. 5(a)-5(c). This shows that velocity does not influence significantly the patterns of the images. However, as it is expected, the similarity score drops as time step, t, changes, Fig. 5(b) and Fig. 5(c). In the remaining graphs, Figs. 5(d)-5(f), of Fig. 5, the distance at which images were taken is $8m$ while the velocity has been kept the same as is in the reference image, that is, $4km/h$. The similarity score in these figures appears to be lower revealing that distance influences more than velocity the optic flow patterns. Nevertheless, the similarity score can be considered quite satisfactory, in general.

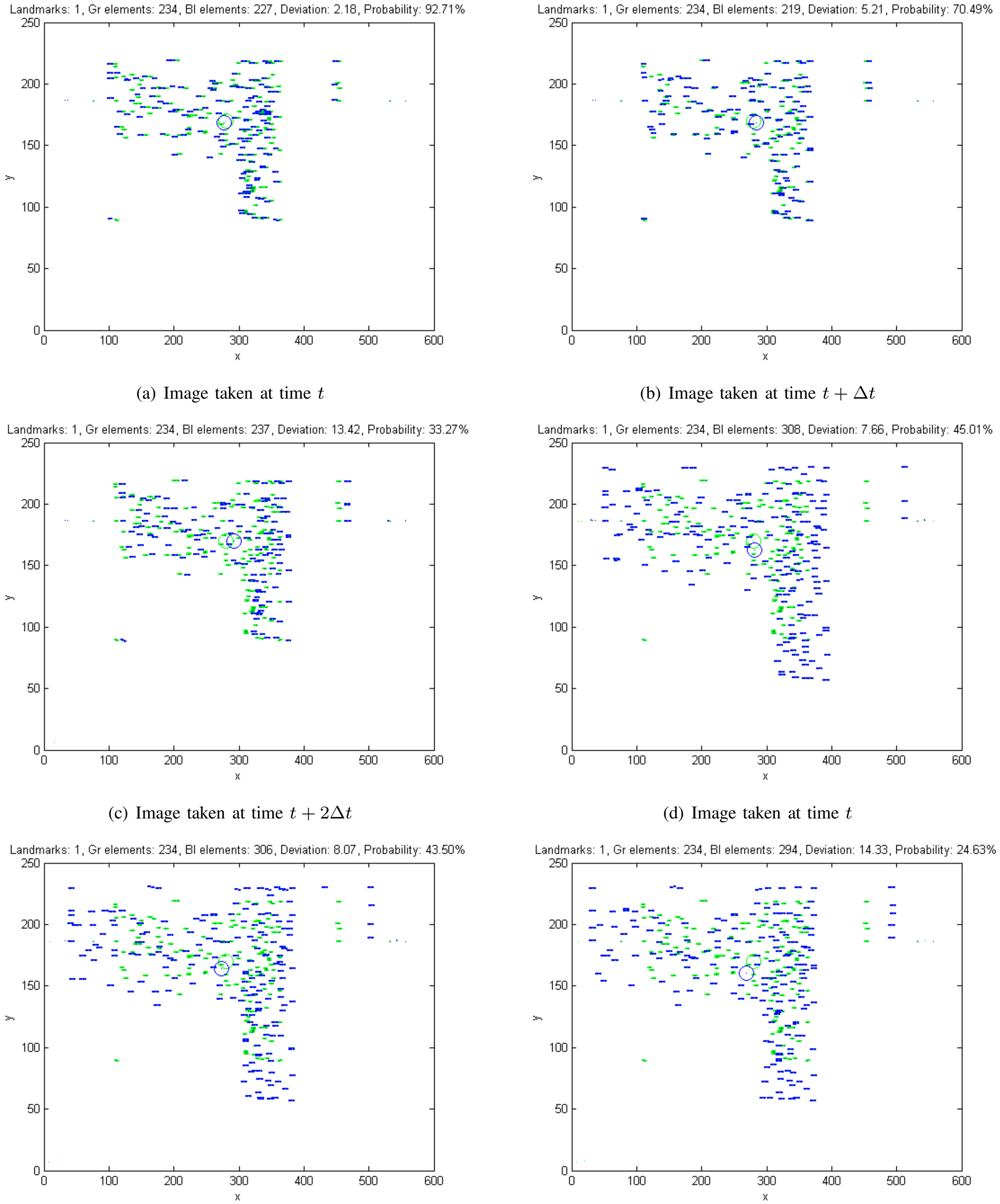
Figure 6 depicts two different landmarks and the similarity score is inferred against the reference landmark of Fig. 3. The distance and velocity at which they were captured remains the same as is in the reference image. In the first graph, Fig. 6(c), the similarity score is quite low, that is 2.04% while in the next graph, Fig. 6(d), the similarity score is high enough, that is 39.13% although the two landmarks are different to each other. In the former case, the probability is low because deviation is large while in the latter, the probability is high because deviation is small. As it can be seen in the last graph, Fig. 6(d), the texture and the shape of the landmark, Fig. 6(b), resemble that of the reference landmark, Fig. 3. In cases such as this, the matching algorithm, and hence the localisation of the robot can be erroneous.

Finally, in Fig. 7, a comparison between two landmarks is attempted. In the first graph, Fig. 7(c), two landmarks are captured initially at a distance of $5m$ from the sphere-like landmark and $11m$ from the tower-like landmark, Fig. 7(a). In the same graph, a second snapshot has been captured but at a distance of $13m$ from the tower-like landmark and $7m$ from the sphere-like landmark. The velocity at which the initial snapshot was taken is $4km/h$ while at the current (second) snapshot the velocity was increased to $5km/h$. Their similarity score is at an acceptable level, that is, 22.55% for the tower-like landmark and 27.08% for the sphere-like landmark. In the graph of Fig. 7(d) the landmarks of Fig. 7(a) at their initial snapshot are compared with two different landmarks, Fig. 7(b). The hill-like landmark is at a distance of $7.0m$ while the tower-like landmark is at a distance of $13m$. The velocity the image was taken is $5km/h$. The probability in this case is quite low revealing the dissimilarity between the landmarks. In particular, the similarity score of the tower-like landmarks is at 8.34% while for the other two landmarks is at 0.93%.

V. Conclusions and Future Work

The simulation experiments of this work show that a similarity score of 20.0% and above is adequate to identify and recognise a landmark from its optical flow 'fingerprint' . The results are quite encouraging and sensible, especially if we take into consideration the fact that the only information used was optic flow. Our method promises to tackle the homing problem in *a priori* unknown environment using a parsimonious biologically-inspired approach to solve a well-studied problem. Of significant interest is that our model can also help explain the methods employed by insects, and in particular honeybees, to perform localisation and thus homing. To support this, a recent study by Avargues-Weber et al. [39] reveals that honeybees are capable of discriminating faces. It could well be the case of optical flow patterns. In addition, our model does not require the storage or the processing of images every time matching is to be performed. Only the properties of the vectors are stored in every frame, that is, the mean position of all the vectors and the number of vectors.

Future work will focus on using Bayesian statistics to extend the optic flow model to complex scene environments. In this case, a prior distribution needs to be calculated and a likelihood to be determined. The posterior distribution will enable the model to adapt and improve as new observations enter the model. In addition, a robust clustering algorithm for classifying various landmarks would be advantageous. However, the problem in this case is that we do not know which vectors belong to which landmarks. An unsupervised clustering method, therefore, needs to be developed. This problem can also be overcome if there is no counting of



(a) Image taken at time t

(b) Image taken at time $t + \Delta t$

(c) Image taken at time $t + 2\Delta t$

(d) Image taken at time t

(e) Image taken at time $t - \Delta t$

(f) Image taken at time $t - 2\Delta t$

Fig. 5. Optical flow vectors of the reference landmark at its initial setting (distance=$11m$, velocity=$4km/h$) against the optical flow vectors of the same landmark taken at a distance of $11m$ and a velocity of $5km/h$, Figs. 5(a), 5(b), 5(c), and at a distance of $8m$ and a velocity of $4km/h$, Figs. 5(d), 5(e), 5(f).

landmarks and the images are not fragmented into landmarks, but instead are taken as whole piece of information.

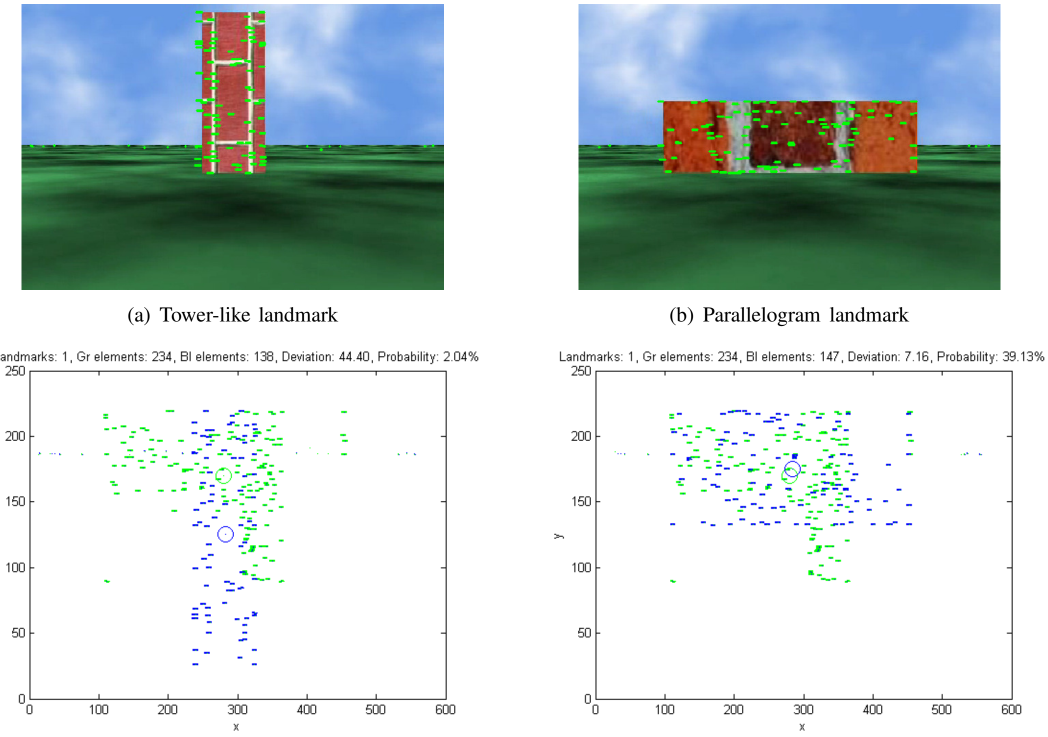
Acknowledgment

The authors would like to thank Dr Klaus-Peter Zauner for his guidance and help in shaping the optic flow idea.

References

[1] F. Matsuno and S. Tadokoro, "Rescue robots and systems in Japan," in *IEEE International Conference on Robotics and Biomimetics*, August 2004, pp. 12–20.

[2] T. Collett, "Insect vision: Controlling actions through optic flow," *Current Biology*, vol. 12, no. 18, pp. 615–617, 2002.



(a) Tower-like landmark

(b) Parallelogram landmark

(c) Optic flow vectors between the tower-like landmark and the reference landmark.

(d) Optic flow vectors between the parallelogram landmark and the reference landmark.

Fig. 6. Comparison of optic flow vectors between two different landmarks and the reference landmark. The distance of $11m$ and the velocity of $4km/h$ remains the same as is in the reference landmark.

[3] S. Schuster, R. Strauss, and K. G. Gotz, “Virtual-reality techniques resolve the visual cues used by fruit flies to evaluate object distances,” *Current Biology*, vol. 12, no. 18, pp. 1591–1594, 2002.

[4] K. Kral and M. Poteser, “Motion parallax as a source of distance information in locusts and mantids,” *Journal of Insect Behavior*, vol. 10, no. 1, pp. 145–163, 1997.

[5] J. Tautz, S. Zhang, J. Spaethe, A. Brockmann, A. Si, and M. Srinivasan, “Honeybee odometry: Performance in varying natural terrain,” *Plos Biology*, vol. 2, no. 7, pp. 915–923, 2004.

[6] H. Esch, S. Zhang, M. Srinivasan, and J. Tautz, “Honeybee dances communicate distances measured by optic flow,” *Nature*, vol. 411, no. 6837, pp. 581–583, 2001.

[7] K. von Frisch, *The Dance Language and Orientation of Bees*. Cambridge, MA, USA: Harvard University Press, 1993.

[8] B. A. Cartwright and T. S. Collett, “Landmark learning in bees,” *Journal of Comparative Physiology A*, vol. 151, no. 4, pp. 521–543, 1983.

[9] B. Cartwright and T. S. Collett, “Landmark maps for honeybees,” *Biological Cybernetics*, vol. 57, no. 1–2, pp. 85–93, 1987.

[10] D. Lambrinos, R. Moller, T. Labhart, R. Pfeifer, and R. Wehner, “A mobile robot employing insect strategies for navigation,” *Robotics and Autonomous Systems, special issue: Biomimetic Robots*, vol. 30, no. 1–2, pp. 39–64, 2000.

[11] A. A. Argyros, C. Bekris, S. C. Orphanoudakis, and L. E. Kavraki, “Robot homing by exploiting panoramic vision,” *Journal of Autonomous Robots*, vol. 19, no. 1, pp. 7–25, 2005.

[12] P. Newman, J. Leonard, J. D. Tardos, and J. Neira, “Explore and return: Experimental validation of real time concurrent mapping and localization,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2002, pp. 1802–1809.

[13] R. Smith, M. Self, and P. Cheeseman, “A stochastic map for uncertain spatial relationships,” in *Fourth International Symposium on Robotics Research*, 1988, pp. 467–474.

[14] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Proceedings of the Ninth International Conference on Computer Vision ICCV ’03, Nice, France*, 2003.

[15] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, “An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser measurements,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2003, pp. 206–211.

[16] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[17] J. J. Gibson, *The Perception of the Visual World*. Santa Barbara, CA, USA: Greenwood Publishing Group, 1974.

[18] T. Camus, D. Coombs, M. Herman, and T.-S. Hong, “Real-time single-workstation obstacle avoidance using only wide-field flow divergence,” in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, 1996.

[19] W. Warren and B. R. Fajen, “From optic flow to laws of control,” in *Optic Flow and Beyond*, L. M. Vaina, S. A. Beardsley, and S. K. Rushton, Eds. Kluwer Academic Publishers, 2004, pp. 307–337.

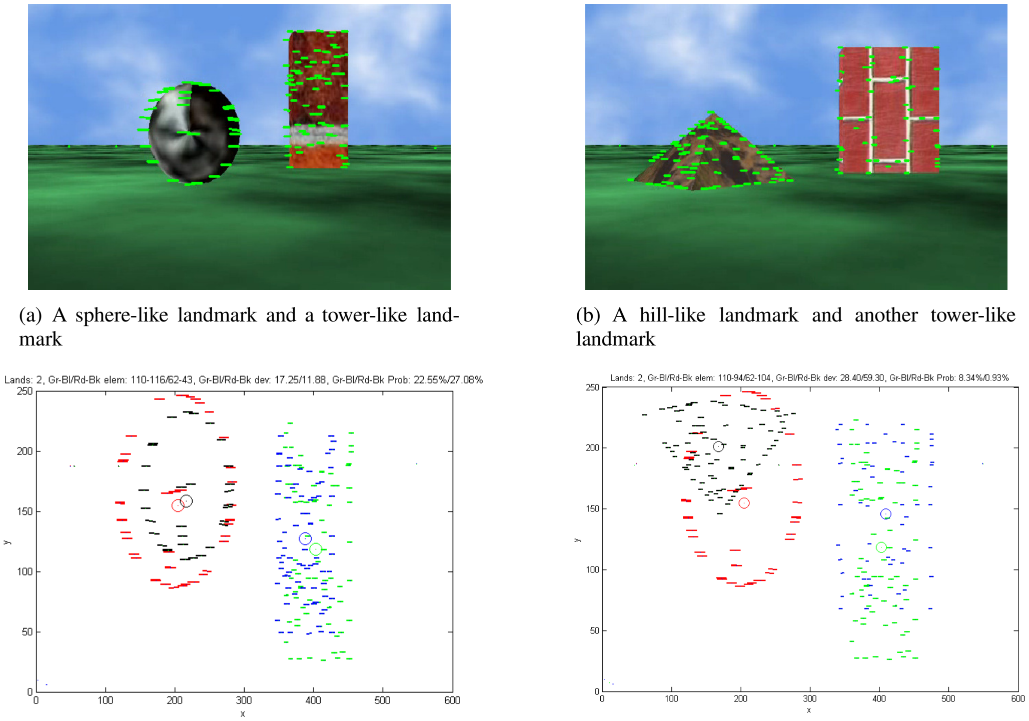
[20] P. C. Merrell, D.-J. Lee, and R. Beard, “Obstacle avoidance for unmanned air vehicles using optical flow probability distributions,” *Sensing and Perception*, vol. 5609, pp. 13–22, 2004.

[21] S. Hrabar, G. Sukhatme, P. Corke, K. Usher, and J. Roberts, “Combined optic-flow and stereo-based navigation of urban canyons for a UAV,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 2005, pp. 302–309.

[22] N. Ohnishi and A. Imiya, “Corridor navigation and obstacle avoidance using visual potential for mobile robot,” in *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, May 2007, pp. 131–138.

[23] H. Madjidi and S. Negahdaripour, “On robustness and localization accuracy of optical flow computation for underwater color images,” *Computer Vision and Image Understanding*, vol. 104, no. 1, pp. 61–76, 2006.

[24] A. Vardy, “Biologically plausible methods for robot visual homing,” Ph.D. dissertation, School of Computer Science, Carleton University, 2005.



(a) A sphere-like landmark and a tower-like landmark

(b) A hill-like landmark and another tower-like landmark

(c) Optic flow vectors captured initially at a distance of $5m$ from sphere-like landmark and $11m$ from the tower-like landmark. The initial velocity at which image was taken is $4km/h$. The current (second) snapshot has been taken at $2m$ further away from the initial snapshot and at a velocity of $5km/h$.

(d) Optic flow vectors between the landmarks of Fig. 7(a) at their initial snapshot and two new different landmarks, Fig. 7(b), located at a distance of $13m$ from the tower and $7.0m$ from the hill-like landmark. The velocity the image was taken is $5km/h$.

Fig. 7. Comparison of optic flow vectors between identical and different landmarks.

[25] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, no. 6–7, pp. 591–602, 2009.

[26] J.-C. Zufferey, A. Beyeler, and D. Floreano, "Optic flow to control small UAVs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems: Workshop on visual guidance systems for small autonomous aerial vehicles*, Nice, France, September 2008.

[27] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[28] W. Simpson, "Optic flow and depth perception," *Spatial Vision*, vol. 7, no. 1, pp. 35–75, 1993.

[29] M. Blackburn and H. Nguyen, "Vision based autonomous robot navigation: Motion segmentation," in *Proceedings for the Dedicated Conference on Robotics, Motion, and Machine Vision in the Automotive Industries*, Stuttgart, Germany, September 1995, pp. 353–360.

[30] H. Frenz and M. Lappe, "Absolute travel distance from optic flow," *Vision Research*, vol. 45, no. 13, pp. 1679–1692, 2005.

[31] M. Langer and R. Mann, "Optical snow," *International Journal of Computer Vision*, vol. 55, no. 1, pp. 55–71, 2003.

[32] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), August 24-28*, 1981, pp. 674–679.

[33] OpenCV, "http://opencv.willowgarage.com/wiki/," 2008.

[34] G. Bradski and A. Kaehler, *Learning OpenCV, computer vision with the opencv library*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2008.

[35] B. K. P. Horn and G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[36] M. Dacke and M. Srinivasan, "Evidence for counting in insects," *Animal Cognition*, vol. 11, no. 4, pp. 683–689, 2008.

[37] MATLAB, "http://www.mathworks.com/," 2005.

[38] J. Klein, "Breve: A 3D simulation environment for the simulation of decentralized systems and artificial life," in *Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems*, 2002.

[39] A. Avargues-Weber, G. Portelli, J. Benard, A. Dyer, and M. Giurfa, "Configural processing enables discrimination and categorization of face-like stimuli in honeybees," *Journal of Experimental Biology*, vol. 213, no. 4, pp. 593–601, 2009.