# Linked Local Visual Navigation and Robustness to Motor Noise and Route Displacement

Lincoln Smith, Andrew Philippides, Paul Graham, and Phil Husbands

Centre for Computational Neuroscience and Robotics
University of Sussex BN1 9QG, United Kingdom
{lincolns,andrewop,paulgr,philh}@sussex.ac.uk

**Abstract.** This paper presents an investigation into the robustness to motor noise of an insect-inspired visual navigation method that links together local view-based navigation in a series of visual locales automatically defined by the method. The method is tested in the real world using specialist robotic equipment that allows a controllable level of motor noise to be used. Extensions to the method, which can improve its robustness to severe motor noise and to major disruptions such as being displaced along its route, are investigated.

## 1 Introduction

We recently introduced an insect-inspired method for navigating across relatively complex environments by linking together local view-based navigation using the across a series of automatically defined visual locales [1]. Within the Linked Local Navigation (LLN) framework, local homing was achieved using the Average Landmark Vector (ALV) model. The LLN method was demonstrated in a variety of simulated and real environments and was shown to be inherently robust to visual noise. In addition, many mobile robot platforms also suffer from high degrees of motor noise, as do insects while being buffeted by gusts of wind for example. Hence, in order to further evaluate the LLN's promise for real world applications and its potential use in biological modelling, this paper presents an investigation of the LLN method's robustness to motor noise. Using specialist robotic equipment we can systematically explore the effects of motor noise in controllable real world conditions, something not possible in previous route-based models [2,3]. The LLN method is shown to be robust to even high levels of motor noise.

We then investigate simple extensions of the LLN method, based on minimal 'place recognition', which can improve robustness to severe motor noise and to major disruptions such as displacement along the learned route. The ability to recognise a location depends on the representation of the visual scene and a range of representations have been used from Fourier components [4] to colour information [5]. Here we present an analysis of place recognition using only a two and three-dimensional Average Landmark Vector and discuss how place recognition can be used to augment our LLN method.

M. Asada et al. (Eds.): SAB 2008, LNAI 5040, pp. 179–188, 2008.

After briefly describing the main navigation algorithm and the robotic equipment used, the experimental results are presented.

## 2 Linked Local Navigation

This section describes the biologically inspired visual navigation algorithm under investigation. At its core is a computationally efficient view-based navigation method that performs well in locations close to the goal. View-based methods compare the current visual scene with a stored representation of the scene at the goal and derive a direction heading from the difference between the two. A series of local view-based navigation steps are chained together in a novel way to allow navigation over complex environments. For full details see [1].

While there are many algorithms capable of implementing local view-based navigation (for review see [6]), the most parsimonious is the ALV method which processes a view into a single vector. The ALV model requires little computation and memory, and has been shown to be effective for visual navigation in both simulation [7] and on autonomous mobile robots [8,1].

To calculate the ALV, features (landmarks) are selected from a 360 degree panoramic view. The ALV is simply the average of the unit vectors from the agent towards each landmark. For navigation, the agent is placed at a goal location and the ALV there (the goal ALV) stored. To return to the goal, the agent calculates the vector difference between the current ALV and the goal ALV and moves in that direction. Since the difference between the ALVs gives the approximate direction of goal, navigation is implemented by iterating this process [7].

Prerequisites for the ALV are therefore a $360^o$ visual system, an ability to align views with an external reference (e.g., a compass direction) and a robust object detection system. Ants and bees have near spherical vision, both gain compass information from celestial cues [9] and it is assumed that they can reliably segregate objects from the background [10]. Thus the ALV method is biologically plausible and has been shown to be computable with simple artificial neural networks [11].

Taking inspiration from observations of ant navigation [12], the full linked local algorithm requires a training phase where a scaffolding behaviour dictates the route to be learnt. In training, the agent travels along a path from start to goal. If the number of landmarks currently seen is different to the number seen at the previous time-step, the ALV calculated at the previous time-step is stored as a waypoint; this simply requires that the agent can perceive the binary event of an appearance or disappearance of a landmark. When the agent is within 5cm of the goal, the goal ALV is calculated and stored as the final waypoint. In this way the environment is broken up into separate visual locales in which local view-based navigation will be effective.

The navigation phase begins with the agent at the start position with an ordered series of stored ALVs as waypoints. The agent then uses the local navigation method (ALV) to navigate towards the first waypoint. When the number of landmarks changes, the agent is assumed to have crossed a boundary into the next visual locale and the navigation system switches to using the ALV associated with the next waypoint in the list. This process continues until the agent reaches the final goal or times out.

While a movement calculation based purely on the difference between the current and goal (or waypoint) ALVs, is adequate for local navigation, it causes problems at the boundary between visual locales. As the difference becomes small so does the the resultant movement and the agent will very likely not reach the waypoint let alone cross the boundary. After investigating various methods to overcome this [1], we settled on a method which applies momentum to the heading of the agent together with an absolute step-size of $2cm$. The heading direction at time $t$, $\theta_t$, is calculated as a weighted average of the angle dictated by the difference of the current and goal ALVs, $\phi_t$, and the previous heading, $\theta_{t-1}$, with the weight, $\omega_t$, of the previous heading increasing with the value of $|\theta_{t-1} - \phi_t|$, as described by the following equations.

$$\theta_t = \omega_t \theta_{t-1} + (1 - \omega_t)\phi_t, \text{ where } \omega_t = \min\left(\frac{|\theta_{t-1} - \phi_t|}{0.5\pi}, 1\right) \qquad (1)$$

This method prevents large jumps in direction, ignoring $\phi_t$ altogether when $|\theta_{t-1} - \phi_t| \geq \frac{\pi}{2}$. Using egocentric polar coordinates, the movement vector is thus $\mathbf{r_t} = (\mathbf{2}, \theta_\mathbf{t})$.

## 3 The Sussex Gantry Robot and Visual Processing

All experiments reported in this paper were performed on a gantry robot: a large volume XYZ Cartesian robot (Figure 1) with an operating volume of $3 \times 2 \times 2m$. Black/dark-grey cardboard tubes of different diameters were placed within the environment to make high contrast landmarks against the white walls of the gantry. The gantry head can be moved with sub-millimetre precision which allows us, through software control, to effectively control the amount of motor noise experienced by the agent.

For the work presented here we used a panoramic camera mounted on the Z-axis. The camera, a VCAM 360, is shown in Figure 1B. The hemispherical mirror projects a $360^o$ image of the environment on to the downward facing CCD video camera. The image was transformed from a circular reflection to a 1-dimensional image representing a 360- degree panorama (Figure 1C). The transformation was accomplished by taking eight 1-pixel-wide radial samples from the panoramic image. The radial positions of these annular samples are shown by the concentric circles in Figure 1C. Three hundred and sixty one-degree, grey-scale levels were calculated for each radial strip through interpolation. These are then averaged across the eight samples to give a 1 x 360 strip of mean grey-scales rounded to integers in the range [0, 255].

This one-dimensional strip is the raw visual input. At each time-step this is processed into landmarks (Figure 1D) from which the ALV is generated, as

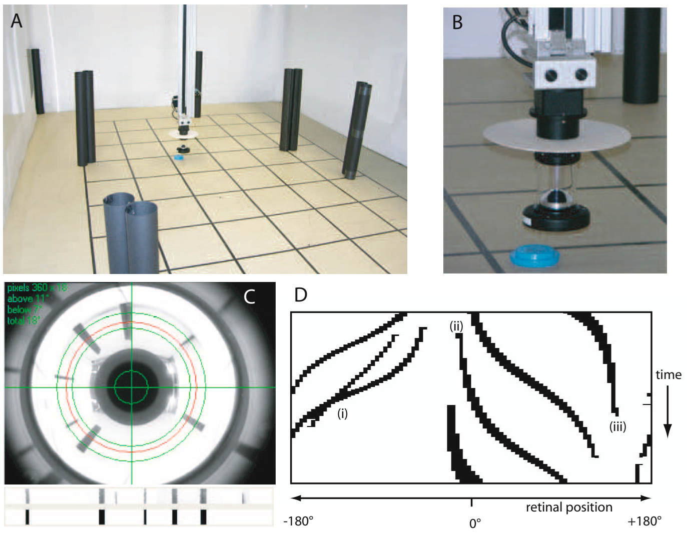

**Fig. 1.** The gantry robot. A: The gantry robot is an XYZ Cartesian robot, which can position a camera at any point in a 3mx2mx2m volume. B: The camera head is a catadioptric system that projects a 360$^o$ panoramic image of the world onto a CCD array. C: A frame capture from the video feed. The three concentric circles (outermost to innermost) indicate the sampled area's upper edge, horizon, and lower edge. The resulting strip, after unwrapping, is shown underneath along with a thresholded strip. D: A trace of the visual input experienced by the agent along a route. This trace demonstrates: (i) occlusion, (ii) appearance of landmarks as they come into perceptual range, and (iii) disappearance of landmarks as they leave perceptual range.

described briefly below. Note that due to occlusions and the limited perceptual range of the agent, the set of perceived landmarks will change during locomotion.

Landmark recognition is accomplished in several sequential stages, described in detail in [1]. Briefly, The raw visual input is first resolved into 90 panoramic facets resulting in an inter-facet angle of the same order as the inter-ommatidial angle of ants' eyes [13]. Each facet has a receptive field covering 8$^o$, that is, itself and half of each of its neighbours. The activation within each facet is averaged and then thresholded to -1 or 1 depending on whether the output is less than 194. We take advantage of the robot's movement to implement two further processing steps based on lateral and temporal excitation/inhibition which serve to 'clean' the visual signal by ameliorating the problem of perceptual 'flickering' of landmarks; that is, single landmarks which are on the edge of the agents' perceptual range and occluding landmarks which are alternately perceived as one or two objects. Effectively, these steps mean that a new landmark appearing is not perceived until it is at least two facets in width. Similarly, once two landmarks have been perceived as one, they are not perceived as 2 landmarks until the gap between them is at least two facets.

Once this visual processing is complete, landmarks are defined as connected sets of active facets and the bearing of each landmark is calculated as the average of the angular position of the facets containing the landmark edges. Thus landmark bearings are accurate to $\pm 2^o$. These bearings are then used to generate the ALV and this, together with a signal determining whether the number of landmarks has changed, is passed to the main algorithm.

Landmark heights were extracted from the full panoramic image generated by the visual system. For each visible landmark (as detected by the image processing method detailed above) we find the point where the central facet of the landmark changes from 1 to -1, starting from the centre of the 2 concentric circles in Figure 1C and working downwards. As all our landmarks are equal height ($1m$, meaning the tops of visible landmarks are above the field of view of our camera) the elevation of the landmark base gives us a proxy for their perceived height. The visual processing is specialized to the environments used in the experiments, but could be generalized.

## 4 Results 1: Motor Noise

Controllable amounts of motor noise were added as follows. During the training phase (see Section 2), the agent travels along a path from start, **s**, to goal, **g**, in steps of $2cm$. When noise is added to the learning route, the learning step $\mathbf{l_t}$ is defined by:

$$\mathbf{l_t} = \mathbf{2}\frac{\mathbf{h_t}}{||\mathbf{h_t}||} + \left(\mathbf{N}\left(\mathbf{0}, \mathbf{2n_{pc}}\right), \mathbf{N}\left(\mathbf{0}, \mathbf{2n_{pc}}\right)\right) \tag{2}$$

where $\mathbf{h_t} = \mathbf{g} - \mathbf{l_{t-1}}$ is the vector from current position to goal (so at the first step, $\mathbf{h_0} = \mathbf{g} - \mathbf{s}$) and $N\left(0, 2n_{pc}\right)$ is a Normally distributed random number drawn with mean 0 and standard deviation $2\,n_{pc}$, where $n_{pc} \in [0, 0.8]$ sets the standard deviation to a percentage of the step-size $2cm$. Thus noise is added independently to $x$ and $y$ dimensions independently and alters both size and direction of the step taken. At the highest noise levels, this can result in the step being taken in the opposite direction than was intended.

During the navigation phase, after the heading vector, $\mathbf{r_t}$ - as defined in Section 2, is transformed into Cartesian co-ordinates $\mathbf{u_t} = (2cos(\theta_t), 2sin(\theta_t))$ noise is added in the same way as for the learning step, to give a final step $\mathbf{v_t}$ of:

$$\mathbf{v_t} = \mathbf{u_t} + \left(\mathbf{N}\left(\mathbf{0}, \mathbf{2n_{pc}}\right), \mathbf{N}\left(\mathbf{0}, \mathbf{2n_{pc}}\right)\right) \tag{3}$$

with parameters as defined above. Note that the momentum on the heading is calculated according to $\theta_t$, the heading the agent 'thinks' it is taking, rather than the actual direction it takes. We feel this is most realistic in terms of an insect or robot's movements being moderated by environmental noise.

To assess the algorithm's robustness to motor noise, we initially ran the algorithm with 5 different noise levels applied to the navigation part of the algorithm. The learning run was performed with no noise so we could isolate the effects of noise on the navigation and learning phases of the algorithm independently. 10

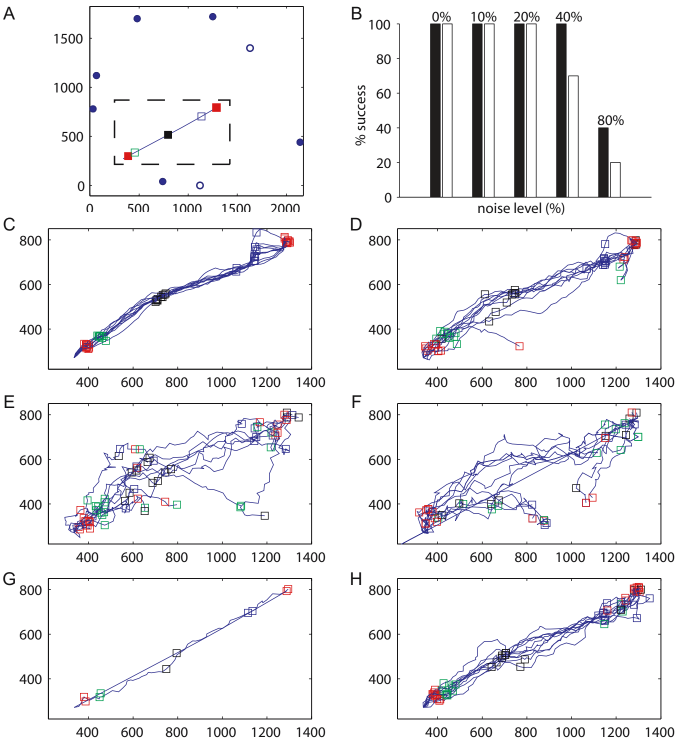

**Fig. 2.** Navigation with motor noise. A: Simple and complex environments with landmarks (circles), waypoints (squares) and learning run (solid line). Dashed line indicates area shown in C-H. Simple environment is as complex environment minus open symbols (2 waypoints and 2 landmarks). B: Navigation success over 10 runs in simple (black) and complex (white) environments for various noise levels ($n_{pc}$). C-F: 10 navigation runs in complex environment ($n_{pc} = 20, 40$ and 80%, C-E respectively) and simple environment ($n_{pc} = 80\%$, F). G: Learning run with 40% noise (straight line shows noiseless learning). H: Navigation after noisy learning run (complex environment, $n_{pc} = 40\%$).

runs were performed for each noise level in first a simple and then a complex environment. The results are shown in Figure 2. Failures were counted as any run in which the agent entered a visual locale which was not the next on its list of waypoints. In both environments, the algorithm was robust to high levels of motor noise.

In the simple environment, the algorithm was successful until the standard deviation of the Gaussian noise reached 1.6 cm, 80% of the step-size. As this noise is applied independently to x and y components of the movement vector, individual steps often took the agent in the opposite direction to which it was trying to home. Successful homing under these conditions demonstrates the benefits of view-based homing as opposed to odometric information. At each step, the direction to home is calculated anew and thus errors do not accumulate. The algorithm will therefore succeed unless by chance the noise on several sequential steps takes the agent in a consistent direction towards a new visual locale which is not the one the agnet 'expects' to encounter next. Thus, one can see that for the high noise level, clusters of failures appear at points where incorrect visual locales occur close to the paths taken by the agent.

In the complex environment, due to the increased number of waypoints and erroneous visual locales, the algorithm's performance dropped when the standard deviation of the Gaussian noise reached 40% of the step-size. Note however that two of the three failures ocurred near the end of the run where there are a succession of waypoints very close to each other (Figure 3D). For many of these 'failures', the algorithm did home successfully to the goal using the goal waypoint and ignoring subsequent boundaries experienced. While the chances of such 'lucky' successes are higher near the goal, success is not guaranteed and so we count these runs as failures.

Finally, we performed a further 10 runs with the standard deviation of the Gaussian noise at 40% on both the learning and navigation parts of the algorithm (Figure 3G-H). The agent homed successfully in all runs. This demonstrates that the algorithm is robust noise on the learning run. Moreover, the greater success with a noisy learning run shows firstly, there is no advantage to the straight-path learning runs we perform; and secondly, the lesser performance of the agent at 40% noise in the complex environment could be due to a 'difficult' set of waypoints to navigate.

## 5 Results 2: Route Recognition

We next wanted to investigate the effects of more severe noise. It is known that ants who are taken from the end of their route near the goal (nest/food) and placed at a new location somewhere near its usual route can successfully navigate back to the route and from there, to the goal. To reproduce this sort of robustness our agent needs, on displacement, to be able to assess which locale it is in, or, more pertinently, which waypoint it should navigate to. Can this be achieved using the minimal visual representation of the world that the agent has available to it?

As a first step to answering this question, we examined the runs with 40% noise and assessed which locale the agent 'thought' it was in at each step of each run. This was accomplished by calculating the distance (according to various metrics) of the (possibly parameterised) current view from the view from each of the stored waypoints. At each step, if the closest waypoint is the one associated with the locale the agent is in, it is assumed the agent can home successfully from that point, first to the winning waypoint and subsequently to the goal. We used three distance metrics based on different representations of the visual scene. The first is the sum square difference between a Cartesian representation of current and waypoint ALV. The second uses only the absolute angular difference between the headings of current and waypoint ALV. As a control, we also used the sum-square difference between the current visual scene and that at the waypoint, that is the

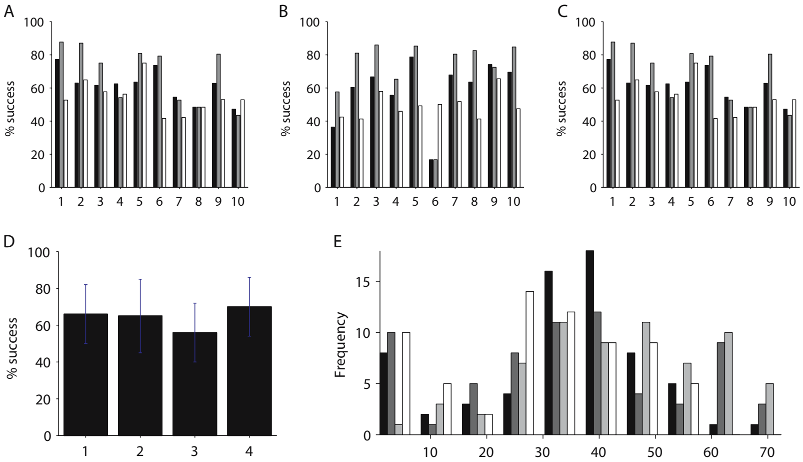

**Fig. 3.** A-C: Percentage route recognition success over 10 runs for Cartesian ALV (black), angular ALV (grey) and unprocessed vision (white). A: Simple environment with 40% noise. B: Complex environment with 40% noise. C: Different learning routes. D-E: Route recognition success over 66 horizontal and vertical transects through the complex environment. D: Mean and standard deviation of percentage success for (left to right) Cartesian ALV, angular ALV, unprocessed vision and 3D sum ALV. E: Distribution of successes of different routes for Cartesian ALV (black), angular ALV (dark grey), unprocessed vision (light grey) and 3D sum ALV (white).

visual input before it is processed into landmarks. Such raw visual input has been shown to be a good indentifier of location in regions close to the goal [14].

Results are shown in Figure 3A-B. In both simple and complex environments results show consistent patterns. The metric based on unprocessed vision is correct over less than half of the route (48 ± 9%), as would be expected from the results of [14]. In both environments the ALV-based metrics perform well in all runs (discounting run 6 of the complex environment, in which navigation failed early) with the angular ALV performing better (69±13% compared to 59±12%). All methods perform best close to the waypoints, on the side where the agent is approaching the waypoints. This is to be expected as the waypoints are points at which the visual scene changes significantly. This effect should be more evident for ALV rather than visual metrics. Indeed, examination of distances to waypoints along each route shows asymmetric profiles for the ALV methods, but characteristic symmetric funnels [14] for the unprocessed vision.

To assess the generality of this result, we examined place recognition over 10 learning runs from a fan of points around the complex environment. Results were fairly consistent with those achieved for the noisy runs (Figure 3C). The angular ALV was again the best (69 ± 17%), while the Cartesian ALV (61 ± 10%) and vision-based metric (54 ± 10%) were slightly more successful than previously. The angular ALV did however have a much greater variance than the other two methods with an almost bi-modal distribution of results. It is therefore worth noting that learning run 1 of these results - in which the angular ALV performs well - is the learning run used for the noisy runs in the complex environment. It is therefore possible that the angular ALV is suited to this particular run, and that its good performance in the noisy runs may well be an overestimate.

As a final test, we assessed route recognition for the whole environment using a database of images gathered at a grid of points with a spacing of $5cm$. To generate routes, we took either horizontal or vertical transects through the environment, with waypoints set down whenever the number of landmarks changed across the route. Averaged over all the 66 transects, differences between the ALV-based methods disappeared while the visual method remained at just over 50% (Figure 3D). To attempt to increase the success-rate we incorporated the height of the landmarks into a 3-dimensional ALV, and assessed all methods with a number of different metrics. Intriguingly, the best performing metric (70 ± 16%)was one which used the sum of the 3-dimensional vectors to each landmark (rather than an average which would result in a 3-D ALV). Presumably the success of this method is linked to the fact that it incorporates a sense of the number of visible landmarks. This latter factor is clearly important as it delineates visual locales, though is prone to aliasing. While differences between the best performing metrics were not statistically significant, the difference in distribution of success rates over the runs (Figure 3E) indicates that the methods are correct indifferent parts of the environment. It is therefore possible a combination of metrics could provide more robust route recognition.

## 6 Discussion

In this paper we have demonstrated that the linked local navigation framework is robust to large amounts of motor noise. Moreover, we have shown that using the ALV, or a representation derived from it, can provide a robust measure of place within a route should a more radical displacement occur.

We have also begun preliminary work to increase robustness further by incorporating route recognition within the navigational algorithm, with encouraging results. Briefly, the algorithm proceeds as usual, but should a change in the number of landmarks occur it does not automatically start using the next waypoint. Instead, the agent assesses whether the current view is associated with the visual locale it is expecting. If so, the algorithm proceeds to home to the next waypoint. If not, it continues homing with the current one. Our initial tests use the number of landmarks to assess whether the locale is correct or not. Future work will explore the use of the continuous encodings presented here, as well as more flexible ways of incorporating route recognition. Finally, we are investigating neural network-based implementations of route recognition.

## Acknowledgments

This work was supported by EPSRC grant GR T08753 01. We thank Prof. Collett and members of the CCNR for useful discussion.

## References

1. Smith, L., Philippides, A., Graham, P., Baddeley, B., Husbands, P.: Linked local navigation for visual route guidance. Adaptive Behavior 15(3), 257–271 (2007)
2. Vardy, A.: Long-range visual homing. In: Zha, H. (ed.) Proceedings of the IEEE International Conference on Robotics and Biomimetics, vol. 1, pp. 220–226 (2006)
3. Franz, M.O., Schölkopf, B., Mallot, H.A., Bülthoff, H.H.: Learning view graphs for robot navigation. Autonomous Robots 5, 111–125 (1998)
4. Menegatti, E., Maeda, T., Ishiguro, H.: Image-based memory for robot navigation using properties of the omnidirectional images. Robotics and Autonomous Systems 47(4), 251–267 (2004)
5. Gourichon, S.: Utilisation d'un compas visuel pour la navigation d'un robot mobile. PhD thesis, Université Pierre et Marie Curie, Paris, AnimatLab (2004)
6. Möller, R., Vardy, A.: Local visual homing by matched-filter descent in image distances. Biological Cybernetics 95(5), 413–430 (2006)
7. Lambrinos, D., Möller, R., Pfeifer, R., Wehner, R., Labhart, T.: A mobile robot employing insect strategies for navigation. Robotics and Autonomous Systems 30, 39–64 (2000)
8. Möller, R.: Modelling the landmark navigation behavior of the desert ant Cataglyphis. Technical Report IFI-AI-00.24, Artificial Intelligence Lab, Dept. Computer Science, University of Zurich (November 2000)
9. Wehner, R., Michel, B., Antonsen, P.: Visual navigation in insects: Coupling of egocentric and geocentric information. Journal of Experimental Biology 199, 129–140 (1996)
10. Möller, R.: Insects could exploit uv-green contrast for landmark navigation. Journal of Theoretical Biology 214(4), 619–631 (2002)
11. Hafner, V., Möller, R.: Learning of visual navigation strategies. In: Quoy, M., Gaussier, P., Wyatt, J. (eds.) Proceedings of the European Workshop on Learning Robots, vol. 1, pp. 47–56 (2001)
12. Collett, M., Collett, T.S.: How do insects use path integration for their navigation? Biological Cybernetics 83, 245–259 (2000)
13. Zollikofer, C., Wehner, R., Fukushi, T.: Optical scaling in conspecific cataglyphis ants. Journal of Experimental Biology 198, 1637–1646 (1995)
14. Zeil, J., Hofmann, M., Chahl, J.: Catchment areas of panoramic snapshots in outdoor scence. J. Opt. Soc. Am. A 20(3), 450–469 (2003)