

# Adaptive Behavior

<http://adb.sagepub.com>

---

## Linked Local Navigation for Visual Route Guidance

Lincoln Smith, Andrew Philippides, Paul Graham, Bart Baddeley and Philip Husbands

*Adaptive Behavior* 2007; 15; 257

DOI: 10.1177/1059712307082091

The online version of this article can be found at:  
<http://adb.sagepub.com/cgi/content/abstract/15/3/257>

---

Published by:

 SAGE Publications

<http://www.sagepublications.com>

On behalf of:

**ISAB**

International Society of Adaptive Behavior

**Additional services and information for *Adaptive Behavior* can be found at:**

**Email Alerts:** <http://adb.sagepub.com/cgi/alerts>

**Subscriptions:** <http://adb.sagepub.com/subscriptions>

**Reprints:** <http://www.sagepub.com/journalsReprints.nav>

**Permissions:** <http://www.sagepub.com/journalsPermissions.nav>

**Citations** (this article cites 20 articles hosted on the SAGE Journals Online and HighWire Press platforms):  
<http://adb.sagepub.com/cgi/content/refs/15/3/257>

# Linked Local Navigation for Visual Route Guidance

Lincoln Smith, Andrew Philippides, Paul Graham, Bart Baddeley, Philip Husbands  
*Centre for Computational Neuroscience and Robotics, Department of Informatics, University of Sussex, UK*

Insects are able to navigate reliably between food and nest using only visual information. This behavior has inspired many models of visual landmark guidance, some of which have been tested on autonomous robots. The majority of these models work by comparing the agent's current view with a view of the world stored when the agent was at the goal. The region from which agents can successfully reach home is therefore limited to the goal's visual locale, that is, the area around the goal where the visual scene is not radically different to the goal position. Ants are known to navigate over large distances using visually guided routes consisting of a series of visual memories. Taking inspiration from such route navigation, we propose a framework for linking together local navigation methods. We implement this framework on a robotic platform and test it in a series of environments in which local navigation methods fail. Finally, we show that the framework is robust to environments of varying complexity.

**Keywords** navigation · view-based homing · snapshot · route learning · biomimetic robotics · average landmark vector

## 1 Introduction

Returning to a location using visual information is an important capability for many insect species as well as autonomous robots. View-based homing has therefore received much attention in recent years with a number of algorithms developed to allow an agent to navigate back to a goal using a remembered view from that location (for reviews see: Franz & Mallot, 2000; Vardy & Möller, 2005). In general, these algorithms compare the current view of the world to a view that was stored at the goal location. The direction to goal is calculated from the discrepancy when the stored and current views are compared. This type of navigation strategy can be further categorized into correspondence methods and holistic methods (Möller & Vardy, 2006). Correspondence methods require that common regions are

identified in both the stored goal view and the current view, and the differences in their positions used to derive a movement vector (e.g., Cartwright & Collett, 1983; Vardy & Möller, 2005). Holistic methods use a difference metric to assess the similarity of goal and current views and an agent homes by moving so as to minimize this difference. The difference metric might be root mean square difference of pixel intensities (e.g., Franz, Schölkopf, Mallot, & Bühlhoff, 1998a), or alternatively, the Euclidean distance in some parameter space, where parameters are derived from the whole image [e.g. average landmark vector (ALV): Lambrinos, Möller, Pfeifer, Wehner, & Labhart, 2000; contour model: Möller, 2001]. Despite their differences, both correspondence and holistic methods perform robustly in areas local to the goal.

For view-based homing algorithms, successful navigation requires the current view to be similar to

*Correspondence to:* Andrew Philippides/Philip Husbands, Centre for Computational Neuroscience and Robotics, Department of Informatics, University of Sussex, Brighton, UK.  
*E-mail:* philh@sussex.ac.uk

Copyright © 2007 International Society for Adaptive Behavior (2007), Vol 15(3): 257–271.  
DOI: 10.1177/1059712307082091

the goal view (Cartwright & Collett, 1987). View-based methods may therefore fail when, for instance, the agent must navigate past an opaque barrier where the world looks completely different on either side. Alternatively, if the world looks similar at multiple locations, the correct location cannot be distinguished without additional information and the end point of a navigated route will depend on the start location. Additionally, when the environment prevents a direct route to a goal, a single stored view is unlikely to suffice. These problems are less likely close to the goal and become more prevalent as the size and complexity of the world is increased.

There have been a number of approaches aimed at increasing the scale over which visually guided robots can navigate. Inspired by the organization of spatial memories in the mammalian hippocampus, some research has focused on algorithms for the autonomous construction of place graphs representing the structure of the environment (for review see: Franz, 2000; Trullier, Wiener, Berthoz, & Meyer, 1997). In the engineering sciences, state of the art robot navigation is dominated by probabilistic algorithms characterized by their ability to simultaneously map and localize within an environment (for review see: Thrun, 2002). Both these approaches are concerned with mapping of the environment and rely on extensive exploration. Possibly a more pragmatic solution to the problem of scale – certainly one that is much more parsimonious in terms of memory and learning – is to learn routes consisting of multiple goal locations, or waypoints, linked together. Navigation between waypoints is then achieved using a local homing method. It has long been known that insects demonstrate robust navigation using visually guided routes without recourse to map-like representations (e.g., Collett, Dillmann, Giger, & Wehner, 1992; Rosengren, 1971; Wehner, Boyer, Loertscher, Sommer, & Menzi, 2006) and navigation algorithms in autonomous robotics have used this idea in various forms (Argyros, Bekris, Orphanoudakis, & Kavraki, 2005; Giovannangeli, Gaussier, & Désilles, 2006; Nehmzow & Owen, 2000; Smith & Husbands, 2002; Smith, Philippides, & Husbands, 2006; Vardy 2006).

Successful schemes that link navigation by local homing methods into a route require solutions to a number of problems of general interest, such as place recognition and the detection of change. These problems, in conjunction with those posed by issues such

as environmental noise, make robust route navigation a non-trivial aim. The two major difficulties that have to be overcome are: (1) determining the point at which a new waypoint should be set during route construction, and (2) deciding when a waypoint has been reached during navigation.

An intuitive approach to the first problem is to monitor the current view and store a new waypoint when there is a large change or discontinuity. For an agent operating in discrete time this means a new waypoint is generated if the change in the view is above a predetermined threshold. Similarly, arrival at a waypoint is indicated by the difference between the current and waypoint views being below a threshold. We have previously shown that while this method can be successful, it is inherently environment specific as no single fixed threshold is suitable for all environments, or even for different parts of a single environment (Smith & Husbands, 2002; Smith et al., 2006). Similarly, Vardy (2006) developed a route navigation algorithm within a single simulated environment and found successful navigation to be threshold-dependent and especially sensitive to waypoint arrival detection. Franz, Schölkopf, Mallot, and Bühlhoff (1998b) developed an algorithm for linking local view-based navigation methods, although using a view graph rather than a route. To deal with waypoint selection and detection, Franz et al. (1998b) used thresholds that were empirically derived by exhaustive sampling of the environment.

In this article, we present a non-threshold-based framework for linking local view-based homing methods together into a route: linked local navigation (LLN). Given that our work is insect-inspired it is important that our algorithm is biologically plausible, that is, could be implemented with simple parallel neural architectures and low memory requirements. Moreover, we wanted to create a solution that did not rely on environment-dependent parameters or extensive exploration. We hope, therefore, that our model falls within the set of plausible models of insect route navigation. The LLN framework assumes that the agent can make a single journey from start to goal using an innate behavior such as path integration or chemical trail following. During this trip, the agent constructs route waypoints by storing views of the world when the number of perceived landmarks changes. Similarly, when navigating, if the number of landmarks changes, a waypoint is judged to have been reached. Thus vis-

ual locales are specified by a binary and significant event and one that an insect could plausibly be expected to recognize. The agent subsequently traverses the route by visually homing to each waypoint in turn using a local navigation method.

This work uses the ALV model for local navigation (Lambrinos et al., 2000). However, the LLN framework could be applied to any view-based correspondence homing method that parses visual input into distinct features. The ALV provides a sparse representation of a visual scene by processing it into a single two-dimensional vector which essentially points at the mean of all landmark-bearings. We use the ALV model as it is the simplest and most elegant of the view-based homing methods and has been implemented using simple parallel architectures (Hafner & Möller, 2001; Möller, 2000). Moreover, its simplicity provides a transparency of operation which lends itself to analysis and investigation of the linking framework. To test the operation of the LLN framework we have implemented it on a robotic platform. We first demonstrate that our framework succeeds in a number of situations in which local navigation methods fail, for example when there is perceptual aliasing or the need to use an indirect route. We then test the framework in a series of complex environments, showing it to have robust operation for goal and start positions over multiple visual locales.

## 2 Methods

### 2.1 Gantry Robot

All experiments reported in this article were performed on a gantry robot – a large volume XYZ Cartesian robot (Figure 1A). The gantry axis configuration provides an operating volume of  $3\text{ m} \times 2\text{ m} \times 2\text{ m}$ . The sensor end of the Z-axis can be placed anywhere within this volume with sub-millimeter accuracy. Black/dark-gray cardboard tubes of different diameters were placed within the environment to make high contrast landmarks against the white walls of the gantry's outer frame.

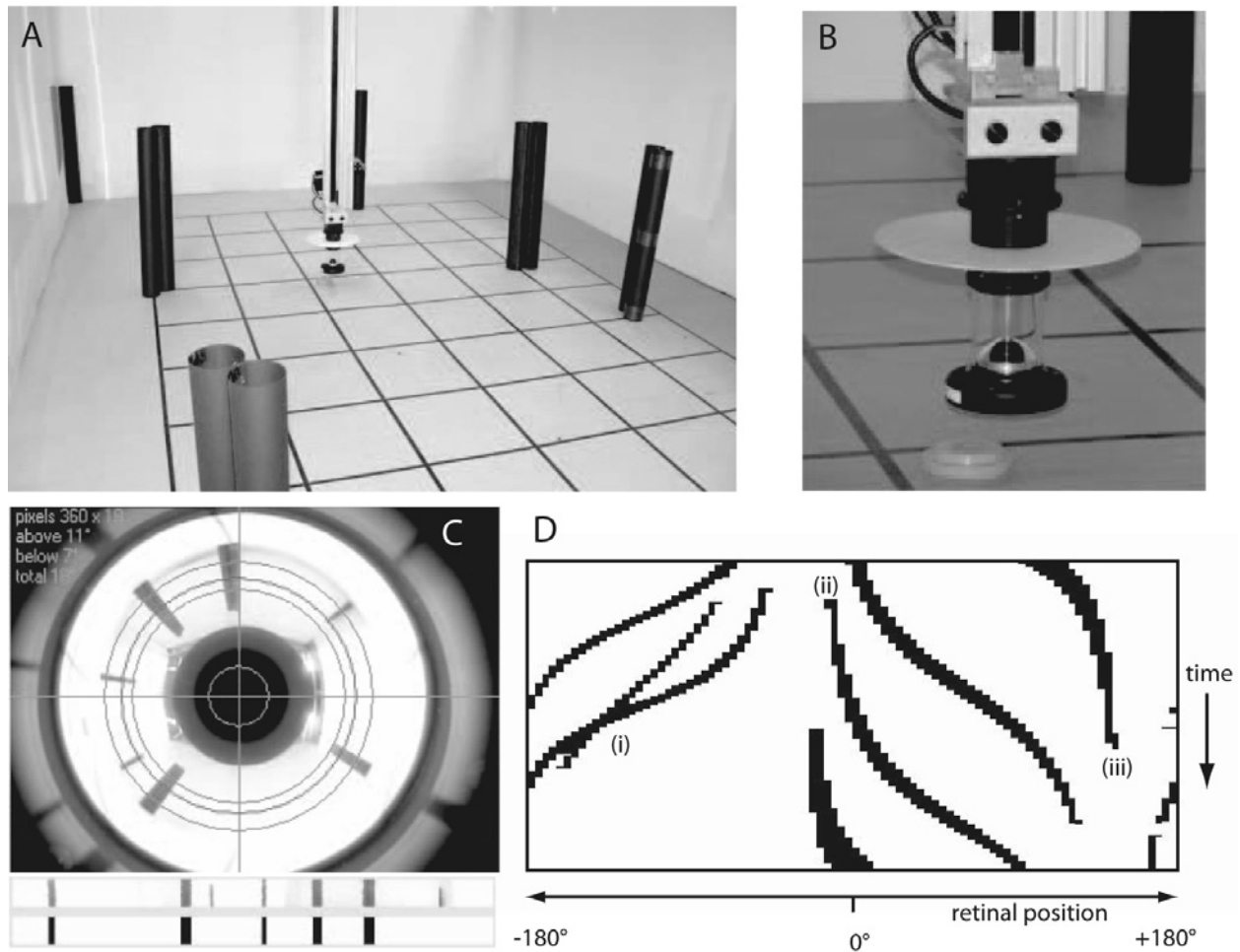
The vertical Z-axis arm accepts interchangeable sensory heads, which are used to collect information from the current XYZ position. For the experiments presented here a catadioptric camera system is mounted on the Z-axis to produce panoramic images. The camera, a VCAM 360, is shown in Figure 1B. The panoramic

mirror projects a  $360^\circ$  image of the environment onto the downward facing CCD video camera which is transformed from a circular reflection to a one-dimensional image representing a  $360^\circ$  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 lines in Figure 1C. Three hundred and sixty one-degree, gray-scale levels were calculated for each radial strip through interpolation and averaged across the eight samples to give a  $1 \times 360$  strip of mean gray-scales rounded to integers in the range  $[0, 255]$ . This one-dimensional strip is the raw visual input. At each time-step this raw input is processed into landmarks (Figure 1D) from which the ALV is generated, as described in Section 2.3. Note that due to the limited perceptual range of the agent and occlusions, the set of perceived landmarks will change during locomotion.

### 2.2 Local Navigation

While there are many algorithms capable of implementing local view-based navigation, the most parsimonious and elegant is the ALV model which processes a view into a single two-dimensional vector. The ALV model thus requires little computation and memory, and has been shown to be effective for visual navigation in both computer simulation (Lambrinos et al., 2000) and on autonomous mobile robots (Möller, 2000). To calculate the ALV, landmarks (recognizable features) are selected from a  $360^\circ$  panoramic view. The ALV is the average of unit vectors directed from the agent to each landmark. For navigation, the agent is placed at a goal location and the ALV (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 this direction. Since the vector difference of the ALVs approximates the direction to goal, navigation is implemented by iteration of this process (Lambrinos et al., 2000).

Prerequisites for the ALV are therefore a  $360^\circ$  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 (Wehner, Michel, & Antonsen, 1996) and we assume they can reliably segregate objects from background (for a suggested method see Möller, 2002).



**Figure 1** The gantry robot. A: The gantry robot is an XYZ Cartesian robot, which can position a camera at any point in a  $3\text{ m} \times 2\text{ m} \times 2\text{ m}$  volume. B: The camera head is a catadioptric system that projects a  $360^\circ$  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.

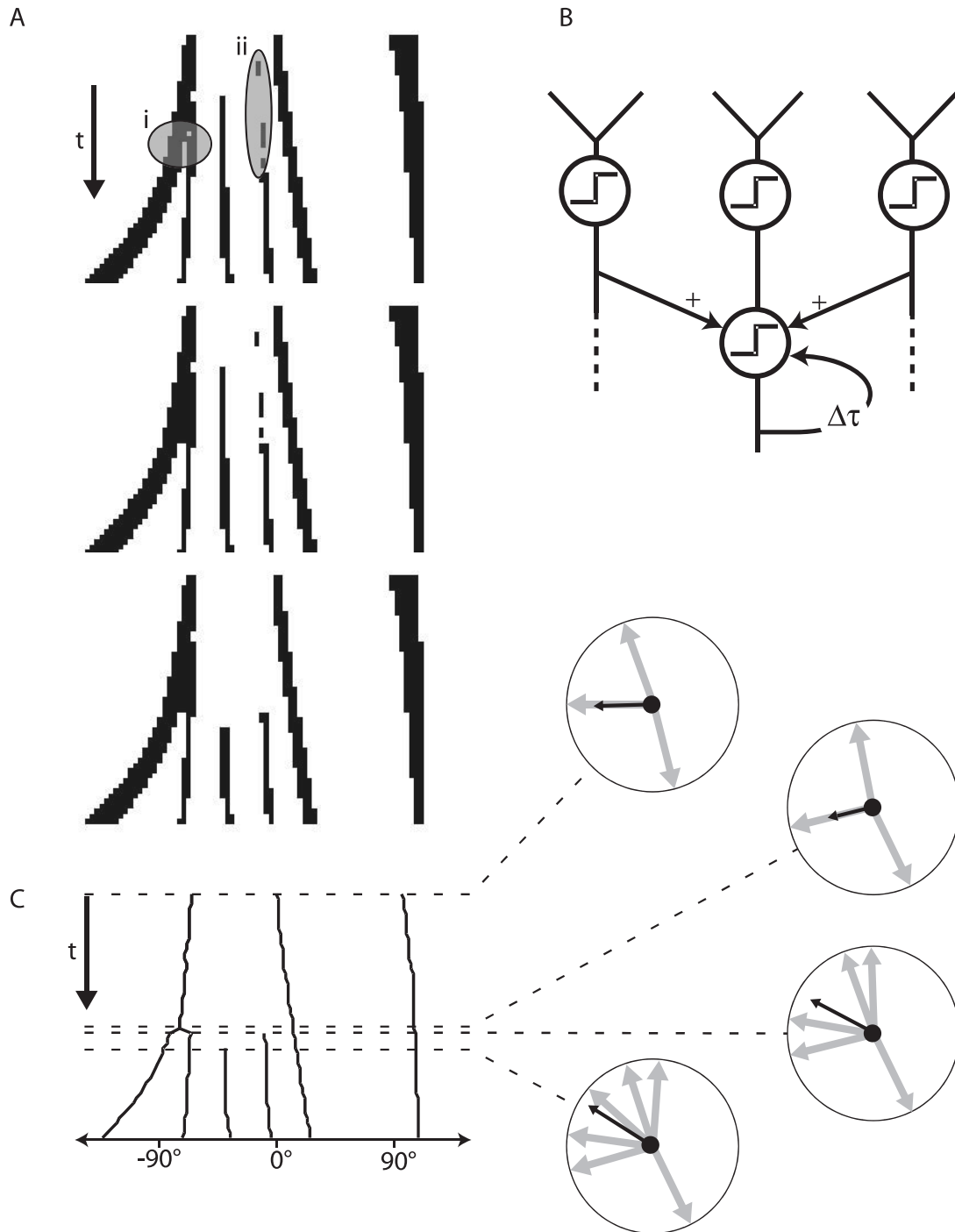
Thus the ALV method is biologically plausible and has been shown to be computable with simple artificial neural networks (Hafner & Möller, 2001; Möller, 2000).

### 2.3 Visual System

As our gantry robot has panoramic vision and fixed alignment, calculation of ALVs only requires a landmark recognition system to reliably distinguish landmarks from background (we use all objects detected by the visual system as landmarks). We note, however,

that the simple landmark recognition system used is not the focus of our article and is clearly optimized for the white-walled, black-object environment used here. Our aim was for a tractable, working system, transparent enough in its action to allow us to focus on the navigation algorithm.

Landmark recognition is accomplished in several sequential stages (Figure 2). 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 (Zollikofer, Wehner, & Fukushi,



**Figure 2** Visual system. A: Landmarks (black objects) perceived over time by the agent after three stages of visual processing. The first stage (upper panel) shows the visual system after raw input is resolved into 90 facets through averaging and thresholding. This stage shows two types of disruptive noise, highlighted in shaded ovals. i: When a landmark appears from behind another they are unreliably perceived as either one or two objects. ii: Landmarks on the edge of the perceptual range are unreliably perceived causing them to flicker in and out of existence. The middle and lower panels show the visual system after excitation and inhibition respectively (see text for details of these operations). B: Schematic of neural circuitry which could achieve the stages of visual processing illustrated in A. C: Retinal positions of landmarks over time. Compass plots show the bearings (thick gray arrows) and the resultant ALV (black arrows) at several points in time (indicated by dashed lines).

1995). Each facet has a receptive field covering  $8^\circ$ , that is, itself and half of each of its neighbors. The activation within each facet is averaged and then thresholded to  $-1$  or  $1$  depending on whether the output is less than 200 (Figure 2A). Two further processing steps based on lateral and temporal excitation/inhibition serve to clean the visual signal and remove noise; these are described later in this section. These steps could be accomplished with simple neural circuitry as illustrated in Figure 2B.

Disruptive noise was generally caused by two factors. Firstly, when landmarks are merging before, or separating after, an occlusion, they may be alternately perceived as one or two objects over a short distance. This is caused by the agent's movement shifting the gap between objects so that it falls between two facets as well as by environmental noise (experiments were conducted over several days at different times of day) coupled with bleed in the camera. After one landmark appears from behind another, the agent briefly perceives it as a single object again, before it fully separates (Figure 2A). The second problem is that landmarks on the threshold of the agent's perceptual range are unreliably perceived. Due to their small apparent size, movement between facets and environmental noise can cause landmarks to flicker in and out of perceptual existence (Figure 2A). For example, as the agent approaches a landmark it increases in size until it is detected by a facet. Further movement, however, means that the landmark passes between two facets, neither reach threshold and the landmark disappears. As the agent does not move directly towards the landmark, it appears and disappears several times as the landmark centre moves across the retina, before appearing as a reliable landmark.

There are many ways to deal with these problems but we decided to take a simple facilitative approach whereby a facet's state depends on its current input, its state at the previous time-step, and the states of neighboring facets. The first stage is excitation and deals with the problem of occlusion. At this stage, any facet which was active in the previous time-step and whose neighbors in the raw output are both currently active is set to 1. This is a form of perceptual "filling in" which means that after two landmarks have been perceived as one, they remain perceived as one until the gap between them is two facets. The visual scene after this stage of processing is indicated in the middle panel of Figure 2A. Note that the occluded landmarks

are perceived together for a longer duration than in the thresholded output (Figure 2A, upper panel). The output of this stage is passed into the second, inhibiting, stage. In this stage any active facets that were not active in the previous time-step and whose neighbors after excitation are both off are set to  $-1$ . Effectively, this means that an object on the edge of the agent's perceptual range is only perceived as a landmark when it is at least two facets in width, at which point it is much more likely to be perceived reliably. The output of this stage is shown in the bottom panel of Figure 2A where perception of the flickering landmark is delayed until it is reliably within perceptual range.

Once these stages are 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  $\pm 2^\circ$ . We assume this stage could be achieved with edge-detecting neural circuitry. 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. This stage is illustrated in Figure 2C. Note the parsimonious nature of the representation of the scene, from bearings to a single vector, means that there is more chance of aliasing than for more complex visual parameterization. Figure 2C shows the ALV changing quite slowly until the number of landmarks changes from three to four, at which point it jumps in a single time-step. However, changes in the number of landmarks do not always precipitate a large change in ALV. During the navigation run shown in Figure 2C, a landmark appears roughly in the direction of the current ALV and therefore has very little effect on the ALV direction component. A method which distinguishes visual locales by large changes in the ALV might not perceive the agent to be in a new visual locale in this case.

## 2.4 Linked Local Navigation

The algorithm requires an initial phase where a scaffold behavior dictates the route to be learnt. In this phase, the agent travels along a path from start to goal in steps of 2 cm. Step-size is chosen to match that used when navigating (see below). 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. Note

that we do not assume that the agent can “count” landmarks but that any agent that can distinguish features as landmarks will be able to perceive the binary event of a feature appearing or disappearing. When the agent is within 5 cm of the goal, the goal ALV is calculated and stored as the final waypoint.

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 ALV algorithm for local navigation to the first waypoint, noting the number of landmarks seen at each time-step. 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 goal or times out (after 500 time-steps).

## 2.5 Boundary Crossing

There are various ways in which the direction derived from the ALV algorithm can be transformed into a movement vector. If it is to be used as part of a route, however, it must be augmented to enable the agent to cross the boundaries between visual locales. This is because the ALV takes an agent to a goal and not past it. Moreover, near the goal several things occur which are problematic for route navigation. Firstly, the size of the difference between current and goal ALV, and thus the movement signal, is small. This means that if the movement is based on the difference only, the resultant movement may be too small to reach the goal let alone cross the boundary. This results in movements which tend towards the goal but never reach it. If this is the global goal, other cues (odor, etc.) could be assumed to get the agent home. However, in the case of a route, the agent’s path will simply stop.

Obvious solutions to this problem are to impose a minimum or absolute step-size, or to add some form of momentum to the agent’s movement vector so that it incorporates some element of recent movements. The first of these solutions brings to light a second problem. As the agent works in discrete time, any waypoint will be taken on one side of the boundary of a visual locale rather than on it. Thus the movement vectors on either side of the waypoint will be in opposing directions. If a minimum step-size is imposed, the agent may overshoot the waypoint and land between the waypoint and boundary. Its next step then takes it back

over the waypoint in the opposite direction, from where it again reverses direction, resulting in the agent flip-flopping from one side of the goal to the other, never crossing the boundary into the next visual locale. Addition of a momentum term to the movement vector such that the resultant movement is the average of the vector calculated from the current visual scene and the last  $n$  movements will ameliorate the problem of flip-flopping. However, the number of steps,  $n$ , to average over to avoid asymptotic approach to the goal without ignoring current direction information must be determined, with optimal values likely specific to particular environments.

We investigated a number of variants of such types of momentum in combination with minimum or absolute step-sizes in a simulation of the gantry robot. Initial tests highlighted the problems mentioned above and the problem of parameter settings only working in certain environments. To combat this we attempted to minimize the number of pre-defined parameters needed for each method and tested each in 300 random environments containing landmarks with radii between 20 and 30 units. Test sets consisted of 30 environments for each radius size (for full results, see: Smith, 2006). After subsequent testing of the most promising methods on the gantry robot itself, we settled on a method which applies momentum to the heading of the agent together with an absolute step-size of 2 cm, as described in the following equations. Defining the movement vector at time  $t$  as  $\mathbf{r}_t = (r, \theta_t)$  and the difference in the heading of the movement vector derived from the current ALV,  $\phi_t$ , and  $\mathbf{r}_{t-1} = (r, \theta_{t-1})$  as  $\alpha_t$ , we have:

$$\sigma_t = \min\left(\frac{|\alpha_t|}{0.5\pi}, 1\right)$$

$$\theta_t = \sigma_t \theta_{t-1} + (1 - \sigma_t) \phi_t.$$

These equations specify the heading as a weighted average of the current and previous headings, with the weight of the previous heading increasing with its difference from the current heading. Such a scheme assumes that the previous heading is reliable and that the current heading is likely to be wrong if it deviates from the previous heading by too much. Thus if the difference between current and previous headings is  $90^\circ$  or more – a situation one would expect if the agent steps passed a waypoint but not across the boundary –



the current heading is ignored entirely and the agent takes another step in the direction of the previous heading. If in a position between waypoint and boundary this should take the agent into the next visual locale. It will also guard against situations where the ALV is corrupted by environmental noise. Moreover, the maximum change in heading at any given step is  $22.5^\circ$ , meaning that the agent cannot turn through unrealistic angles in a single step. Finally, note that setting the absolute step-size,  $r$ , to 2 cm is not optimal. In general, the smaller this value, the more robust the algorithm is to environments with multiple visual locales as the agent is less likely to step over an entire visual locale. However, a very small step-size is not practical.

### 3 Results

As noted in the Introduction, an agent homing with a single visual memory can fail if the landmarks seen from the goal are different from those seen at the current position. As shown in Figure 1D, in a static world, common ways in which the perceived landmarks can change are:

- A landmark appearing due to agent approach.
- A landmark disappearing as the agent moves away.
- A landmark appearing from behind another: occlusion.
- A landmark being hidden by another (or merging with it).

Environments in which one or more of these events occur, may require multiple waypoints for route navigation. Likewise, environments that contain features that force an agent to take a circuitous route to the goal (e.g., a pool of water) may require multiple waypoints. It should be noted that these situations will not necessarily cause local methods to fail, but that one of these events must happen if they are to fail. For instance, perceptual aliasing, where the view from multiple locations is similar, may cause local methods to fail. For this to happen, the agent must not be able to see all landmarks at all locations. Our framework must therefore be robust to such environments if it is to succeed.

Figure 3 illustrates navigation in three environments designed to demonstrate the problems of occlusion, appearance and disappearance of landmarks, and

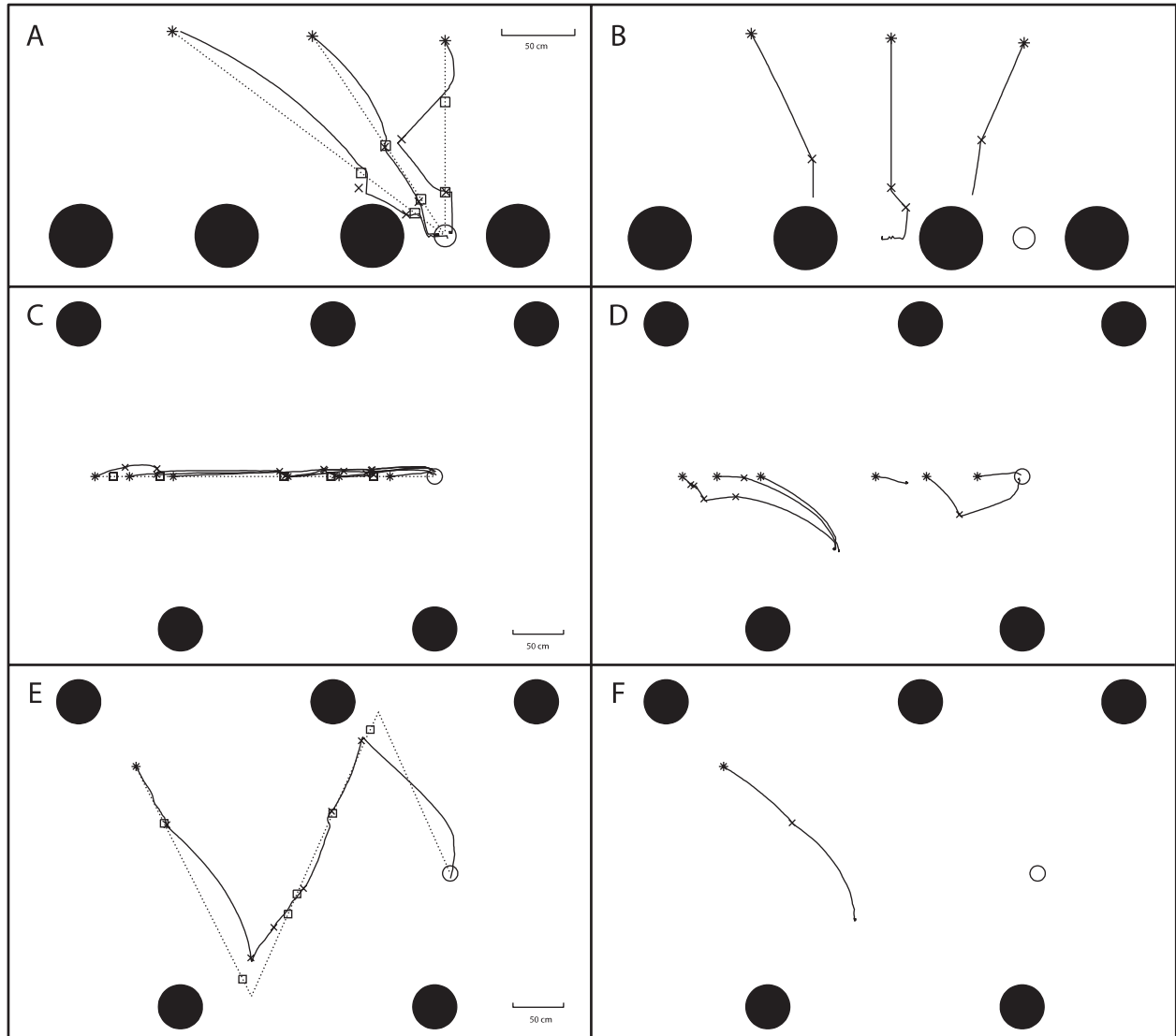
indirect routes. In all instances the LLN framework succeeds while the local method, the average landmark vector algorithm, fails.

The first environment (Figure 3A, 3B and Figure 4) demonstrates how occlusion can lead to perceptual aliasing. The goal is set directly between two landmarks and thus the goal ALV to be matched is (0,0). Examination of the visual input along two trajectories which lead to different positions, highlights the problem (Figure 4). Note the similarity in the perceived visual scene in the later time-steps despite the very different positions of the agents. Moreover, once further processed into the ALV's sparse representation, there is more chance of aliasing since different landmark configurations can produce the same ALV. Crucially in this case, positions where landmarks are on opposite sides of the agent are identical to the goal (Figure 4B and C).

With this information it is easier to interpret the results of Figure 3A. Each run of the ALV vector algorithm initially heads towards the center of the landmark configuration as it tries to achieve the goal ALV of (0,0). The only way to do this is if landmarks are distributed isotropically around the agent and thus the agent moves so as to balance the landmarks on either side of its visual field. These trajectories soon lead it to a point where one landmark is occluded, and the agent sees two landmarks on one side and one on the other. It therefore moves so as to balance these landmarks in its visual field, leading it to collide with a landmark or to a location visually identical to the goal.

Whilst the goal ALV for the LLN method is the same as for the ALV algorithm, because of the intermediate waypoints it avoids this problem. At the first waypoint there is one landmark to the left of the agent and three to the right. The agent thus initially heads for a location with this balance of visual landmarks. Upon reaching the first occlusion point, it heads for a position where two landmarks are to the left and one to the right. This leads it to a point in front of the two landmarks straddling the goal from where perceptual aliasing will not be a problem.

The problems of appearances and disappearances are highlighted in the second environment (Figure 3C and D). Here there are clearly several locations where the balance of landmarks on the retina is the same as is seen from the goal. As it has no higher mechanism to discriminate these locations, which one the ALV algorithm navigates to is dependent upon the starting

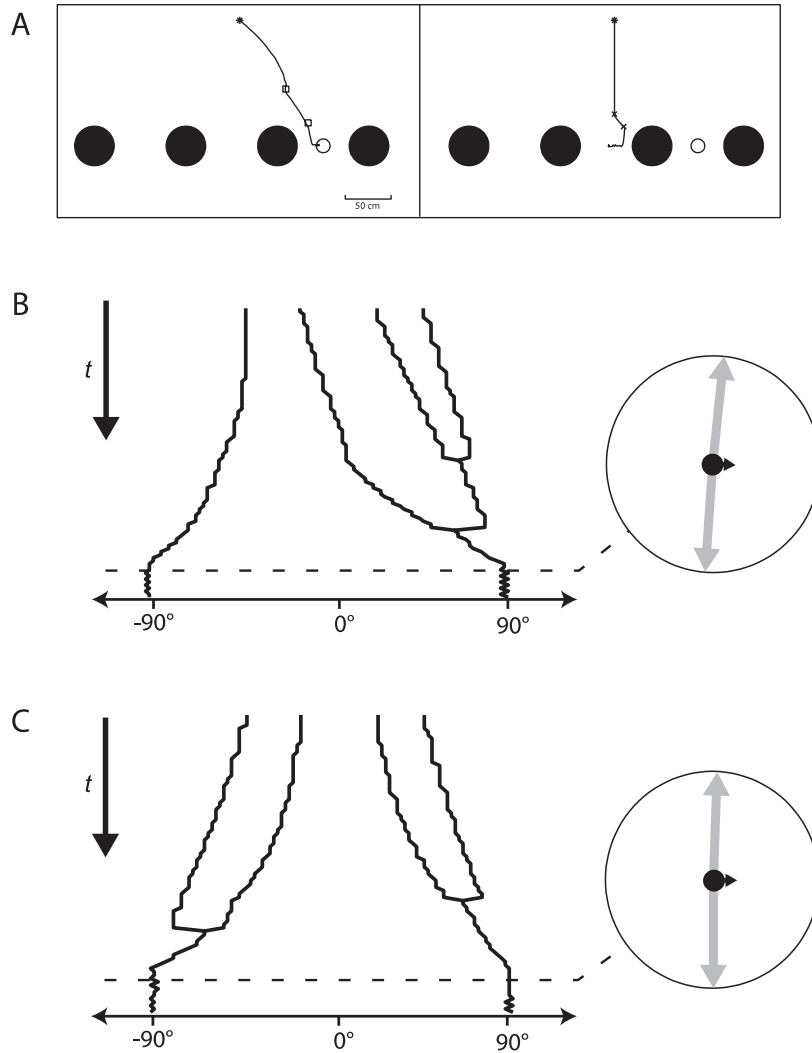


**Figure 3** Environments necessitating linked methods. A, C, E: LLN framework navigating successfully from 3 start points (\*) to a goal (open circle). Dotted lines represent the path taken during the learning phase with open squares being the locations where waypoints were laid. Solid lines represent the paths taken by the agent when navigating to the goal using the stored sequence of views. X-marks signify points where the agent moved into a new visual locale. B, D, F: In the same environments a single ALV stored at the goal fails to guide an agent back there. These environments and routes possess properties causing a single ALV to fail: A, B, perceptual aliasing; C, D, appearance and disappearance of landmarks due to limited perceptual range; and, E, F, an indirect route.

position. This presents little problem for the LLN, however, as the intermediate waypoints lead it to the correct region from which the goal can be found. Similarly, when the agent learns an indirect route to the goal (Figure 3E and F), the LLN follows the route specified by the waypoints along the path.

For a more complete assessment, we further tested the framework from multiple start locations in a ran-

dom environment (Figure 5A). We added extra landmarks to increase the complexity (defined as the number of visual locales). The LLN succeeds in 20 out of the 23 cases compared to 11 out of 23 for the ALV model. As with the environments shown in the previous section, the routes taken by the LLN are generally direct and robust to most deviations from the training path [deviations caused by the inaccuracy of

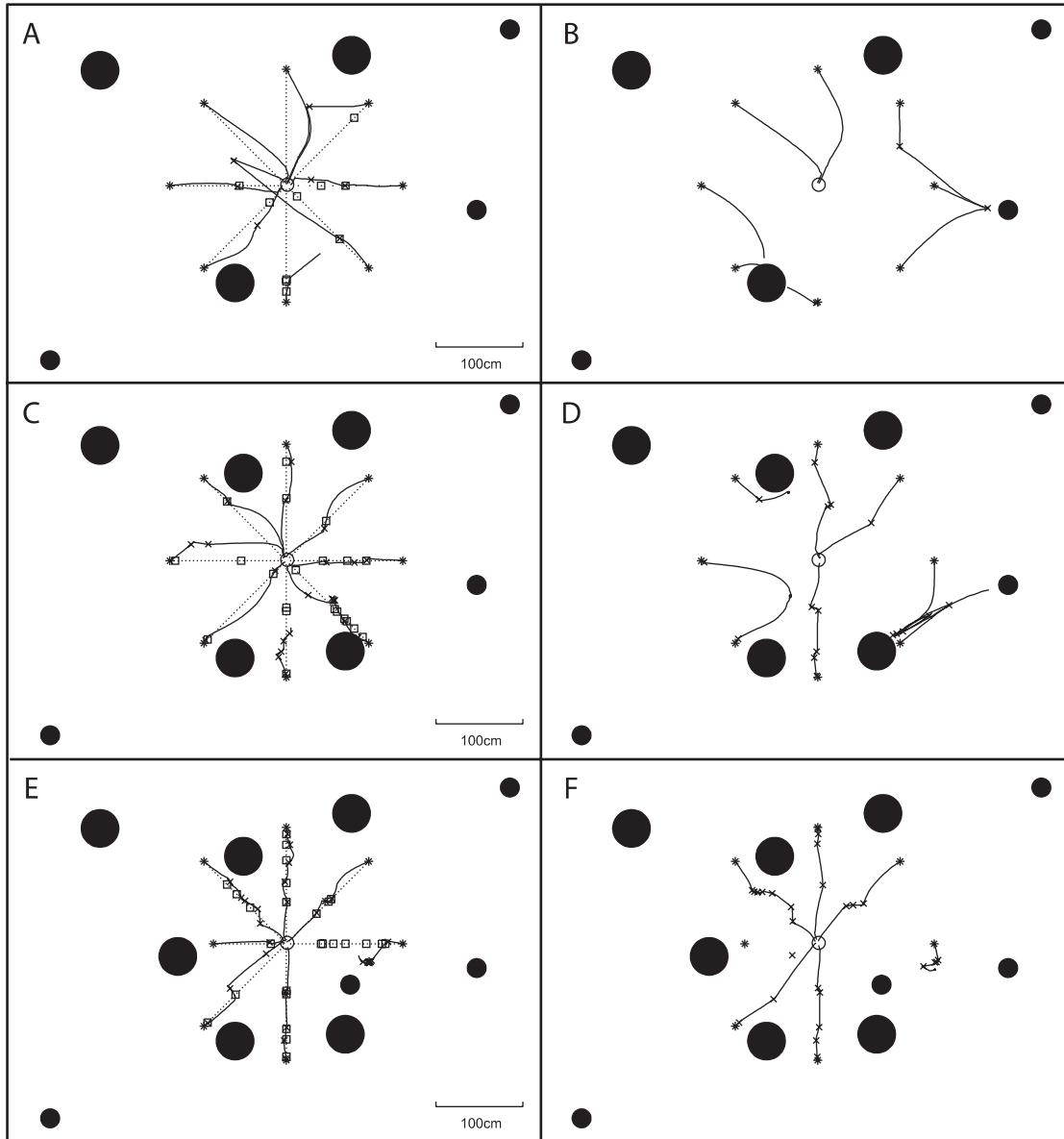


**Figure 4** Perceptual aliasing. A: Trajectories from Figures 3A and B for the LLN framework (left panel) and ALV algorithm (right panel). Symbols and conventions as in Figure 3. B, C: Bearings of landmarks along the trajectories in A show perceptual aliasing due to the simple visual representation of the scene. Compass plots showing bearings of landmarks (thick gray arrows) and the resultant ALV (black arrows) at the end of the LLN framework (B) and ALV algorithm (C) trajectories at the points indicated by the dashed lines. Note that despite slight differences in the bearings of landmarks in the compass plots, after averaging the ALVs are identical.

the isotropic distance assumption of the ALV (see e.g., Franz et al., 1998a)]. These deviations, compounded by the hysteresis of the visual system, mean that boundaries are often detected at different positions to those in training. Nevertheless, as the boundaries occur in the same order as in training, the algorithm succeeds.

The situations in which the LLN fails highlight the limitations of the simple method. Consider the results from the “South” position in Figure 5C. Although the

agent encounters a number of boundaries during the first part of the navigation phase, there are no waypoints set in the training phase. This is because during training, one landmark appears while another disappears simultaneously. Thus, the number of landmarks does not change and no waypoint is set. It is of little surprise that due to the slight variation in the navigation route, the appearance and disappearance occur separately in time, resulting in the agent using inappropriate waypoints.



**Figure 5** LLN framework in environments containing different numbers of landmarks. A, C, E: LLN framework navigating from multiple start points to a single goal. B, D, F: Paths from an agent using a single ALV to home to the goal from the same start points. Symbols and conventions as in Figure 3.

More generally, failures are more likely to occur if waypoints are close to each other. In such instances, differences between training and navigation routes can cause appearances and disappearances to occur in a different order to that in which they were learnt, which may cause failure. Thus, while the robot successfully navigates the tightly spaced waypoints encountered from the “South East” position in Figure 5C, a cluster of waypoints results in failure from the “East” position of Figure 5E.

For success of the LLN, it is desirable for waypoints to be spread out in time. Without using a smaller step-size or a more complex visual parameterization of the world, it would be expected that a more cluttered environment would produce more failures. This is intuitively sensible as there is a probability of failure at each boundary. Interestingly, in our choice of test environments, the reverse appears to be true for the ALV algorithm. It performs best in the most clut-

tered environment due to the central position of the goal. There are very few positions in the environment that could have such an isotropic distribution of landmarks. Thus certain configurations of goal and landmarks will result in success, especially if the start or goal positions are not surrounded by landmarks.

When this is not the case, local methods often fail. This is illustrated in Figure 6 which highlights a range of complex environments where the start and goal are not in central positions. For instance, Figure 6A and B show a long route with many waypoints, where the ALV algorithm falls victim to perceptual aliasing. In the second example, we imagine that there is some physical obstacle requiring the agent to take a dog-legged learning path. A local method is likely to fail in such a case. To further complicate this situation, we also consider the case where the obstacle obstructs the agent's view. Despite radically changing visual scenes, the LLN framework succeeds in both cases (Figure 6C–F). Finally, we consider the example of agents trained to complete a circular route (Figure 6G and H). The LLN succeeds although, trivially, as the goal position is also the start position, a local method will not.

## 4 Discussion

We have presented a framework that allows robust route navigation through complex environments. The framework uses linked local methods to navigate to a series of waypoints along a route. We have demonstrated the efficacy of the framework on a robotic platform in multiple environments. In these tests, the interaction of the landmark arrays with the visual system of the agent meant that the environments were visually complex, containing landmark occlusions, appearances and disappearances (cf. Franz et al., 1998b; Vardy, 2006). Route methods are necessary as local view-based homing methods (e.g., the ALV model, Lambrianos et al., 2000) are likely to fail in such visually complex environments.

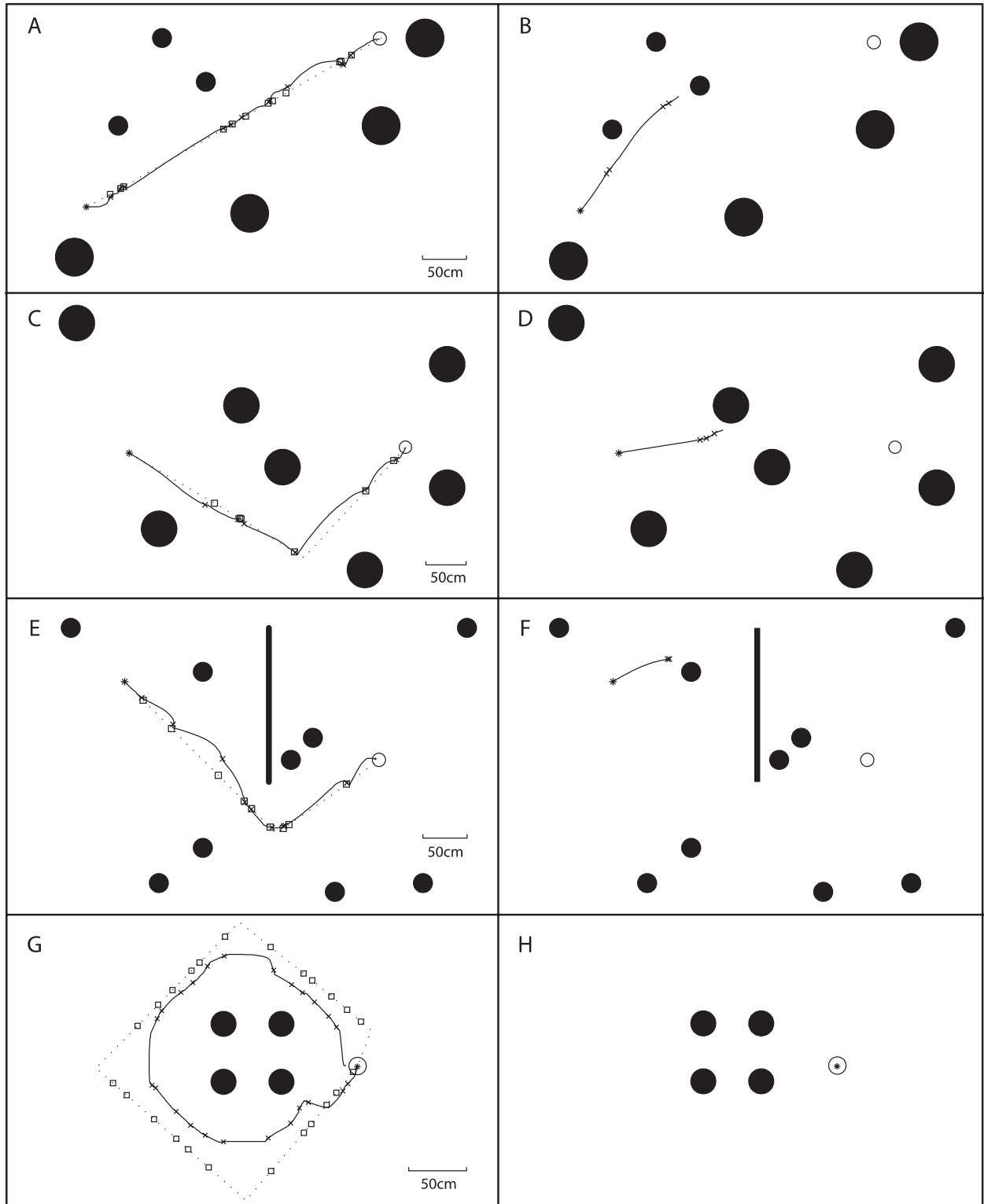
Using a series of waypoints for route navigation requires robust methods for deciding when to store a waypoint during the learning phase and when that waypoint has been reached during the subsequent navigation phase. Methods that use thresholds to signify these events have to be tuned to a specific environment. For this reason we avoided the use of thresholds and assumed the agent had arrived in a new visual

locale when there was a change in the number of perceived landmarks. This event-based indication of boundary crossing could be assumed to be underpinned by simple visual processing.

In this work we purposely chose a simple visual system and a simple visual environment in order to allow us to access the inner workings of our model. This aligned well with the ALV model which works on highly processed images or in simple worlds. Ways in which the ALV method can be adapted to work in visually complex environments have been discussed by Hafner and Möller (2001) and Möller (2001). However, these modified ALV approaches do not have an equivalent to the landmark count that we use as an event marker for crossing the boundary between visual locales. To use our model framework in complex visual environments would require a more sophisticated visual system which selects landmarks from natural scenes (e.g., Argyros et al., 2005; Lehrer & Bianco, 2000). A visual system which reliably extracts landmarks could be used in conjunction with our existing framework and the ALV method of local navigation.

Limitations of the model, and therefore future research directions, are highlighted by considering the impressive performance of insects during route navigation. For example, Kohler and Wehner (2005) allowed Australian desert ants to learn a foraging route then displaced ants to different points along the route. The ants were able to recognize their location and complete their habitual route. This example highlights two key points. Firstly, the ability of ants to recognize their location in a visually complex world with a poor visual system, and, secondly the ability of ants to access route memories out of sequence.

Ants might be able to recognize their location on a route by using low spatial frequency information (i.e., from large distant landmarks) as context for identifying smaller local landmarks. This hierarchical representation of the world increases the robustness and accuracy of place recognition, which allows the ants to be confident about their location within the route sequence. Representing the visual world using multiple spatial scales has been suggested for artificial visual navigation systems (e.g., Cartwright & Collett, 1987; Stürzl & Mallot, 2006). Representing the world on different spatial scales should also ameliorate the major cause of failure in the current model, which was dealing with multiple boundaries in close proximity. Alternatively, a system that only uses landmarks that



**Figure 6** Long routes using LLN. Trajectories are shown where the agent had to deal with a long route (A, B), a dog-leg (C, D), a barrier (E, F) and a circular route (G, H). As previously the ALV algorithm is tested over the same routes and these runs are shown in the right-hand panels. Symbols and conventions as in Figure 3.

are judged to be reliable will be more robust and overcome some of the problems of boundary crossing. Selecting appropriate landmarks could be part of a more extensive learning phase.

Despite these limitations the model presented here is a simple framework which has robust performance in multiple environments whilst maintaining its biological plausibility. As such we believe the LLN model is a good basis for further research into route based autonomous navigation.

## Acknowledgments

L.S., A.P. and B.B. are funded by EPSRC grant GR-T08753-01. P. G. was funded by the BBSRC.

## References

- Argyros, A., Bekris, K., Orphanoudakis, S., & Kavraki, L. (2005). Robot homing by exploiting panoramic vision. *Autonomous Robots*, *19*, 7–25.
- Cartwright, B. A., & Collett, T. S. (1983). Landmark learning in bees. *Journal of Comparative Physiology A*, *151*, 521–543.
- Cartwright, B. A., & Collett, T. S. (1987). Landmark maps for honeybees. *Biological Cybernetics*, *57*, 85–93.
- Collett, T. S., Dillmann, E., Giger, A., & Wehner, R. (1992). Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A*, *170*, 435–442.
- Franz, M. O., & Mallot, H. A. (2000). Biomimetic robot navigation. *Robotics and Autonomous Systems*, *30*, 133–153.
- Franz, M. O., Schölkopf, B., Mallot, H. A., & Bülthoff, H. H. (1998a). Where did I take that snapshot? Scene-based humming by image matching. *Biological Cybernetics*, *79*, 191–202.
- Franz, M. O., Schölkopf, B., Mallot, H. A., & Bülthoff, H. H. (1998b). Learning view graphs for robot navigation. *Autonomous Robots*, *5*, 111–125.
- Giovannangeli, C., Gaussier, P., & Désilles, G. (2006). Robust mapless outdoor vision-based navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3293–3300), Beijing: IEEE.
- Hafner, V. V. & Möller, R. (2001). Learning of visual navigation strategies. In M. Quoy, P. Gaussier, & J. Wyatt, (Eds.), *Proceedings of the European Workshop on Learning Robots* (Vol. 1, pp. 47–56), Prague.
- Kohler, M., & Wehner, R. (2005). Idiosyncratic route-based memories in desert ants, *Melophorus bagoti*: How do they interact with path-integration vectors? *Neurobiology of Learning and Memory*, *83*, 1–12.
- Lambrinos, D., Möller, R., Pfeifer, R., Wehner, R., & Labhart, T. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, *30*, 39–64.
- Lehrer, M., & Bianco, G. (2000). The turn-back-and-look behaviour: bee versus robot. *Biological Cybernetics*, *83*, 211–229.
- Möller, R. (2000). *Modelling the landmark navigation behavior of the desert ant Cataglyphis* (Technical Report: IFI-AI-00.24). Zurich, Switzerland: Artificial Intelligence Lab, Department of Computer Science, University of Zurich.
- Möller, R. (2001). Do insects use templates or parameters for landmark navigation? *Journal of Theoretical Biology*, *210*, 33–45.
- Möller, R. (2002) Insects Could Exploit UV-Green Contrast for Landmark Navigation. *Journal of Theoretical Biology*, *214*, 4, 619–631.
- Möller, R., & Vardy, A. (2006). Local visual homing by matched-filter descent in image distances. *Biological Cybernetics*, *95*, 413–430.
- Nehmzow, U., & Owen, C. (2000). Robot navigation in the real world: experiments with Manchester's *FortyTwo* in unmodified, large environments. *Robotics and Autonomous Systems*, *33*, 223–242.
- Rosengren, R. (1971). Route fidelity, visual memory and recruitment behaviour in foraging wood ants of the genus *Formica* (Hymenopteran Formicidae). *Annales Zoologici Fennici*, *133*, 1–150.
- Smith, L. (2006). *Visual navigation in mobile robots*. Submitted for DPhil., Department of Informatics, University of Sussex.
- Smith, L., & Husbands, P. (2002). Visual landmark navigation through large-scale environments. In R. Dampier (Ed.) *Proceedings of the EEPSC/BBSRC International Workshop on Biologically-Inspired Robotics: The Legacy of W. Grey Walter* (pp. 272–279), EPSRC/BBSRC.
- Smith, L., Philippides, A., & Husbands, P. (2006). Navigation in large-scale environments using an augmented model of visual homing. In J. G. Carbonell & J. Siekmann, (Eds.), *From Animals to Animats 9* (pp. 251–262). Berlin/Heidelberg: Springer.
- Stürzl, W., & Mallot, H. A. (2006). Efficient visual homing based on Fourier transformed panoramic images. *Robotics and Autonomous Systems*, *54*, 300–313.
- Thrun, S. (2002). Robotic mapping: a Survey. In G. Lakemeyer & B. Nebel, (Eds.), *Exploring artificial intelligence in the new millennium*. San Francisco: Morgan Kaufmann.
- Trullier, O., Wiener, S. I., Berthoz, A., & Meyer, J. A. (1997). Biologically based artificial navigation systems: review and prospects. *Progress in Neurobiology*, *51*, 483–544.
- Vardy, A. (2006). Long-range visual homing. In H. Zha (chair) *Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics* (pp. 220–226). IEEE Xplore.
- Vardy, A., & Moller, R. (2005). Biologically plausible visual homing methods based on optical flow techniques. *Connection Science*, *17*, 47–90.

Wehner, R., Boyer, M., Loertscher, F., Sommer, S., & Menzi, U. (2006). Ant navigation: one-way routes rather than maps. *Current Biology*, 16, 75–79.

Wehner, R., Michel, B., & Antonsen, P. (1996). Visual navigation in insects: coupling of egocentric and geocentric

information. *Journal of Experimental Biology*, 199, 129–140.

Zollikofer, C. P. E., Wehner, R., & Fukushi, T. (1995). Optical scaling in conspecific *Cataglyphis* ants. *Journal of Experimental Biology*, 198, 1637–1646.

## About the Authors



**Lincoln Smith** is a research fellow in the Centre for Computational Neuroscience and Robotics (CCNR), University of Sussex. His research area is the visual navigation of autonomous robots; in particular, the use of biological models of insect navigation for direction in the design of algorithms for robotic navigation in large-scale environments.

*E-mail:* lincolns@sussex.ac.uk



**Andrew Philippides** is a research fellow in the CCNR at the University of Sussex. He is interested in the modeling of biological systems at both behavioral and neuronal levels.

*E-mail:* andrewop@sussex.ac.uk



**Paul Graham** is a RCUK research fellow in the CCNR, University of Sussex. His research involves behavioral studies of insect navigation and he is interested in how robotics can be used as a tool for understanding biological systems.

*E-mail:* paulgr@sussex.ac.uk



**Bart Baddeley** is a research fellow in the CCNR. His research background is in neural networks and machine learning. His current research is into autonomous navigation in both biological and artificial systems. *E-mail:* b.t.baddeley@sussex.ac.uk



**Philip Husbands** is Professor of Artificial Intelligence and Co-director of the CCNR, University of Sussex. His research interests include biologically inspired adaptive robotics, evolutionary systems, computational neuroscience and creative systems.